Novel Particle Swarm Optimization for Low Pass FIR Filter Design

[1]SANGEETA MONDAL, [1]S. P. GHOSHAL, [2]RAJIB KAR, [2]DURBADAL MANDAL
[1]Department of Electrical Engg.
[1]Department of Electronics and Communication Engg.
National Institute of Technology, Durgapur
INDIA
rajibkarece@gmail.com

Abstract: - This paper presents an optimal design of linear phase digital low pass finite impulse response (FIR) filter using Novel Particle Swarm Optimization (NPSO). NPSO is an improved particle swarm optimization (PSO) that proposes a new definition for the velocity vector and swarm updating and hence the solution quality is improved. The inertia weight has been modified in the PSO to enhance its search capability that leads to a higher probability of obtaining the global optimal solution. The key feature of the proposed modified inertia weight mechanism is to monitor the weights of particles, which linearly decrease in general applications. In the design process, the filter length, pass band and stop band frequencies, feasible pass band and stop band ripple sizes are specified. FIR filter design is a multi-modal optimization problem. Evolutionary algorithms like real code genetic algorithm (RGA), particle swarm optimization (PSO), and the novel particle swarm optimization (NPSO) have been used in this work for the design of linear phase FIR low pass (LP) filter. A comparison of simulation results reveals the optimization efficacy of the algorithm over the prevailing optimization techniques for the solution of the multimodal, non-differentiable, highly non-linear, and constrained FIR filter design problems.

Key-Words: - FIR Filter, RGA, PSO, NPSO, Parks and McClellan (PM) Algorithm, Evolutionary Optimization, Low Pass Filter

1 Introduction

Digital Filter is an important part of digital signal processing (DSP) system and it usually comes in two categories: finite impulse response (FIR) and infinite impulse response (IIR). FIR filter is an attractive choice because of the ease in design and stability. By designing the filter taps to be symmetrical about the centre tap position, a FIR filter can be guaranteed to have linear phase. Linear phase FIR filters are also required when time domain features are specified [1-2].

Traditionally, there are many well known methods for FIR design, such as the window method, frequency sampling method etc. The windowing method simply consists of truncating or windowing a theoretically infinite filter impulse response by some suitably chosen window function. The window method is fast, convenient, robust but generally suboptimal. A window is a finite array of coefficients selected to satisfy the desirable requirements. There are various kinds of window functions (Butterworth, Chebyshev, Kaiser, and Hamming) available depending on the filter specifications to be met like ripples in pass and stop band, stop band attenuation and transition width. Its major disadvantage is the lack of precise control of the critical frequencies such as pass band and stop band cut-off frequency and the transition width. These values depend on the type of the window and the order of the filter.

Remez Exchange algorithm proposed by Parks and McClellan is used for the design of exact linear phase weighted Chebyshev FIR filter [3]. Further a computer program has been developed for the design of FIR digital filter by McClellan *et al.* [4]. The major limitation of this approach is that the relative values of the amplitude error in the frequency bands are specified by means of the weighting functions and not by the deviations themselves. The program has to be iterated many times in order to meet the filter specifications in terms of stop band attenuation, cut-off frequency and filter length [5].

The objective function for the design of optimal digital filters involves accurate control of various parameters of frequency spectrum and is thus highly non-uniform, non-linear, non-differentiable and multimodal in nature. Classical optimization methods cannot optimize such objective functions and cannot converge to the global minimum solution. Because they have disadvantages such as: i) highly sensitive to starting points when the number of solution variables and hence the size of the solution space increase, ii) frequent convergence to local optimum solution or divergence or revisiting the same suboptimal solution, iii) requirement of continuous and differentiable objective cost function (gradient search methods), iv) requirement of the piecewise linear cost approximation (linear programming), and v) problem of convergence and algorithm complexity (non-linear programming).

So, evolutionary optimization methods have been implemented for the design of optimal digital filters with better control of parameters and the highest stop band attenuation. Different heuristics and stochastic optimization methods have been developed, which have proved themselves quite efficient for the design of FIR filter like GA [7-9], simulated annealing [10], Tabu Search [11], Differential evolution [12] and artificial bee colony optimization [13] etc. GA proves itself to be more efficient in terms of obtaining local optimum while maintaining its moderate computational complexity but they are not very successful in determining the global minima in terms of convergence speed and solution quality [14].

In this paper, the benefits of designing the FIR filter using an evolutionary technique known as Particle Swarm Optimization has been explored. The PSO proves itself to be far more efficient than the previously discussed techniques in many aspects. Particle Swarm Optimization is an evolutionary optimization technique developed by Eberhart *et al.* [15]. The merits of PSO lie in its simplicity to implement as well as its convergence can be controlled via few parameters. Several works have already been done in order to explore the flexibility of FIR filter design provided by PSO [14], [16]. PSO is used for the design of FIR digital filters by using LMS and Minimax strategies for different populations and number of iterations [16].

Several modifications of the conventional PSO technique have been made to increase its efficiency. PSO is used with the differential evolution [17] to design optimal filter. The inertial weights and acceleration coefficients are the parameters of PSO whereas scaling factor and recombination probability are the parameters of DE. With the use of this method, the optimization algorithm becomes insensitive to the parameters of PSO as well as DE. The inertial weight concept and the neighbour topology of PSO are used with the concept of the DE, which avoids the trapping of the solution in local minima as well as it speeds up the convergence process. Quantum-behaved Particle Swarm Optimization (QPSO) which was proposed by Sun *et al.* [18], is a novel algorithm based on the PSO and quantum model. In this concept each particle has quantum behaviour. In quantum mechanics, a particle has a wave function instead of having position and velocity. By using this concept, one can find the positions and velocities of the particles of search space exactly so the algorithm gets modified accordingly. QPSO is used for the design of FIR filters in [19]. Quantum infused PSO is also utilized for the design of digital filters [20]. The global best is selected by comparing the global best obtained from the conventional PSO and the offspring obtained from the QPSO. DEPSO and PSO-QI have been used for FIR filter design problem in [21]. More recently, craziness based PSO (CRPSO) has been applied for FIR filter design problem in [22].

Most of the above algorithms show the problems of fixing algorithm's control parameters, premature convergence, stagnation and revisiting of the same solution over and again [23], [24]. In order to overcome these problems, in this paper, a novel particle swarm optimization technique [25] and a novel fitness function are employed for the FIR low pass (LP) filter design.

The rest of the paper is arranged as follows. In section 2, the FIR LP filter design problem is formulated. Section 3 discusses the algorithms of RGA, conventional PSO and the NPSO algorithm. Section 4 describes the simulation results obtained for FIR LP filter using PM algorithm, RGA, PSO and the proposed NPSO approach. Finally, section 5 concludes the paper.

2 Problem Formulation

The main advantage of the FIR filter structure is that it can achieve exactly linear-phase frequency responses. That is why almost all design methods described in the literature deal with filters with this property. Since the phase response of linear-phase filters is known, the design procedures are reduced to real-valued approximation problems, where the coefficients have to be optimized with respect to the magnitude response only.

A digital FIR filter is characterized by,

$$H(z)=\sum_{n=0}^{N} h(n)z^{-n}, \text{ n=0, 1,…, N} \quad (1)$$

where N is the order of the filter which has (N+1) number of filter's impulse response coefficients, h(n). The values of h(n) will determine the type of the filter, e.g., low pass, high pass, band pass etc. The values of h(n) are to be determined in the design process and N represents the order of the polynomial function. This paper presents the even order FIR LP filter design with h(n) as positive even symmetric. Because the h(n) coefficients are symmetrical, the dimension of the problem is halved. Thus, (N/2+1) number of h(n) coefficients are actually optimized, which are finally concatenated to find the required (N+1) number of filter coefficients. An ideal filter has a magnitude of one in the pass band and a magnitude of zero in the stop band. Error fitness function is formed by the errors between the frequency responses of the ideal filter and the designed approximate filter. In each iteration of the optimization algorithm, error fitness values of particle vectors are calculated and used for updating the particle vectors with new coefficients h(n). The final particle vector obtained after a certain number of iterations or after the error fitness is below a certain limit is considered to be the optimal result, yielding an optimal filter. Various filter parameters which are responsible for the optimal filter design are stop band and pass band normalized frequencies (ω_s, ω_p), pass band and stop band ripples (δ_p and δ_s), stop band attenuation and transition width. These parameters are decided by the filter coefficients. Several scholars have investigated and developed algorithms in which *N*, δ_p, and δ_s are fixed while the remaining parameters are optimized. Other algorithms were originally developed by Parks and McClellan (PM) [3] in which *N*, ω_p, ω_s, and the ratio δ_p / δ_s are fixed.

In this paper, evolutionary optimization algorithms like RGA, conventional PSO and NPSO are individually applied to obtain the actual designed filter response as close as possible to the ideal response.

Now for (1), the particle i.e. the coefficient vector $\{h_0, h_1, \ldots, h_N\}$, which is optimized, is represented in (N/2+1) dimension instead of (N+1) dimension.

The frequency response of the FIR digital filter can be calculated as,

$$H\left(e^{j\omega_k}\right) = \sum_{n=0}^{N} h(n) e^{-j\omega_k n}; \qquad (2)$$

where $\omega_k = \frac{2\pi k}{N}$; $H\left(e^{j\omega_k}\right)$ is the Fourier transform complex vector. This is the FIR filter's frequency response. The frequency is sampled in [0, π] with N points. Different kinds of error fitness functions have been used in different literatures. An error function given by (3) is the approximate error used in PM algorithm for filter design [3].

$$E(\omega) = G(\omega)\left[H_d\left(e^{j\omega}\right) - H_i\left(e^{j\omega}\right)\right] \qquad (3)$$

where $H_d\left(e^{j\omega}\right)$ is the frequency response of the designed approximate filter; $H_i\left(e^{j\omega}\right)$ is the frequency response of the ideal filter; $G(\omega)$ is the weighting function used to provide different weights for the approximate errors in different frequency bands. For ideal LP filter, $H_i\left(e^{j\omega}\right)$ is given as,

$$H_i\left(e^{j\omega}\right) = 1 \quad \text{for} \quad 0 \le \omega \le \omega_c; \qquad (4)$$
$$= 0 \quad \text{otherwise}$$

where ω_c is the cut-off frequency. The major drawback of PM algorithm is that the ratio of δ_p / δ_s is fixed. To improve the flexibility in the error function to be minimized, so that the desired level of δ_p and δ_s may be specified, the error function given in (5) has been considered as fitness function in many literatures [14] [21]. The error fitness to be minimized using the evolutionary algorithms, is defined as:

$$J_1 = \max_{\omega \le \omega_p}\left(|E(\omega)| - \delta_p\right) + \max_{\omega \ge \omega_s}\left(|E(\omega)| - \delta_s\right) \qquad (5)$$

where δ_p and δ_s are the ripples in the pass band and stop band, respectively, and ω_p and ω_s are pass band and stop band normalized cut-off frequencies, respectively. Since the coefficients of the linear phase positive symmetric even order filter are matched, the dimension of the problem is halved. This greatly reduces the computational burdens of the algorithms.

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

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

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 fitness function given in (6) represents the generalized fitness function to be minimized using the evolutionary algorithms RGA, conventional PSO, and the proposed NPSO individually. Each algorithm tries to minimize this error fitness J_2 and thus optimizes the filter performance. Unlike other error fitness functions 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 stop band ripples. Transition width is also kept reduced. Since the coefficients of the linear phase filter are matched, the dimension of the problem is halved. This greatly reduces the computational burdens of the algorithms, applied to the optimal design of linear phase positive even symmetrical FIR filters.

3 Optimization Techniques Employed

3.1 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. At each generation it maintains a population of individuals where each individual is a coded form of a possible solution of the problem at hand called chromosome. Chromosomes are constructed over some particular alphabet, e.g., the binary 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 the objective function of the corresponding optimization problem. The basic steps of RGA as implemented for the optimization of h(n) coefficients are [26]:

- Initialization of real chromosome strings h(n) of n_p population, each consisting of a set of h(n) coefficients. Size of the set depends on the number of coefficients in a particular filter design.
- Decoding of strings and evaluation of error fitness value of each string.
- Selection of elite strings in order of increasing error fitness values from the minimum value.
- Copying of the elite strings over the non-selected strings.
- Crossover and mutation to generate off-springs.
- Genetic cycle updating and repeat from the step of evaluation error fitness value of each string.
- The iteration stops when the maximum number of genetic cycles is reached. The grand minimum error fitness value and its corresponding chromosome string or the desired optimal solution vector is finally obtained.

3.2 Conventional Particle Swarm Optimization (PSO)

PSO is a flexible, robust population-based stochastic search / optimization technique with implicit parallelism, which can easily handle with non-differential objective functions, unlike traditional optimization methods. PSO is less susceptible to getting trapped on local optima unlike GA, Simulated Annealing etc. Eberhart *et al.* [15] developed PSO concept similar to the behaviour of a swarm of birds. PSO is developed through simulation of bird flocking in multi-dimensional space. Bird flocking optimizes a certain objective function. Each particle vector (bird) knows its best value so far (*pbest*). This information corresponds to personal experiences of each particle vector. Moreover, each particle vector h(n) knows the best value so far in the group (*gbest*) among *pbest*s. Namely, each particle tries to modify its position using the following information:

• The distance between the current position and the *pbest*.

• The distance between the current position and the *gbest*.

Similar to GA, in PSO techniques also, real-coded particle vectors of population n_p are assumed. Each particle vector consists of components or sub-strings as required number of normalized filter coefficients, depending on the order of the filter to be designed. Mathematically, velocities of the particle vectors are modified according to the following equation [15]:

$$V_i^{(k+1)} = w \times V_i^k + C_1 \times rand_1 \times \left(pbest_i^k - X_i^k\right) + C_2 \times rand_2 \times \left(gbest^k - X_i^k\right) \quad (7)$$

where V_i^k is the velocity of ith particle vector at kth iteration; w is the weighting function; C_1 and C_2 are the positive weighting factors; $rand_1$ and $rand_2$ are the random numbers between 0 and 1; X_i^k is the current position of i^{th} particle vector h(n) at k^{th} iteration; $pbest_i^k$ is the personal best of the i^{th} particle at the k^{th} iteration; $gbest^k$ is the group best of the group at the k^{th} iteration. The searching point in the solution space may be modified by (8).

$$X_i^{(k+1)} = X_i^k + V_i^{(k+1)} \quad (8)$$

The first term of (7) is the previous velocity of the particle vector. The second and third terms are used to change the velocity of the particle vector. Without the second and third terms, the particle vector will keep on ''flying'' in the same direction until it hits the boundary. Namely, it corresponds to a kind of inertia represented by the inertia constant, w and tries to explore new areas. The steps of conventional PSO as employed for FIR LP filter design is given in Table 1.

Table 1. Steps of PSO

Step 1: Initialization: Population (swarm size) of particle vectors, n_P=25; maximum iteration cycles=350; number of filter coefficients in each particle vector are h(n); filter order, nvar=20; C_1, C_2 =2.05; minimum and maximum values of filter coefficients, hmin=-1, hmax= 1; number of samples = 128; $\delta_p = 0.1$, $\delta_s = 0.01$; initialization of the velocities of all the particle vectors.
Step 2: Generate initial particle vectors of filter coefficients (nvar/2 +1) randomly with limits; Computation of initial error fitness values of the total population, n_P from (6).
Step 3: Compare the fitness of each particle vector with each pbest. If the current solution is better than its pbest, then replace its hpbest by the current solution.
Step 4: Compare the fitness of each particle vector with each gbest. If the fitness of any particle vector is better than its gbest, then replace its hgbest.
Step 5: Update the velocity and position of all particle vectors according to equation (7) and (8), respectively.
Step 6: Repeat steps 2-5 until the maximum iteration cycles or the convergence of minimum error fitness values are met; finally, hgbest is the particle vector of optimal filter coefficients (nvar/2 +1); Form complete (nvar+1) coefficients by copying (because the filter has linear phase) before getting the optimal frequency spectrum.

3.3 Novel Particle Swarm Optimization (NPSO)

The drawback of the conventional PSO used for the generation of optimal coefficients of filter design problem is that it results in sub-optimality problem. In general, the initial solutions are usually far from the global optimum and hence the larger inertia weight *w* may be proved to be beneficial [25]. Large inertia weight enables the PSO to explore globally and small inertia weight enables the PSO to explore locally. This inertia weight *w* plays the important role of balancing the global and local exploration abilities. The value of *w* for all particles will decrease at the same time as the iteration number increases and is calculated using the following expression,

$$w = w_{\max} - (w_{\max} - w_{\min}) \times \frac{iter}{iter_{\max}}; \qquad (9)$$

where, $w_{\max}$ and $w_{\min}$ are the initial and final weight, respectively. The standard PSO has oscillatory problem and is easy to be trapped in local optima if a promising area where the global optimum resides is not identified at the end of the optimization process. The further development of conventional PSO is used to improve the possibility of exploring the search space where the global optimal solution exists. A slightly different approach further provides a well-balanced mechanism between global and local exploration abilities. The proposed weighting function is defined as follows:

$$\begin{aligned} w_{qi}^k &= w_{\max} - \left(w_{\max} - w_{\min}\right) \times \frac{Z_{iter,qi}^k}{Z}, & \text{if } v_{qi}^k \times \left(x_{i,gbest}^k - x_{qi}^k\right) > 0 \quad (10) \\ w_{qi}^k &= w_{qi}^{k-1} & \text{if } v_{qi}^k \times \left(x_{i,gbest}^k - x_{qi}^k\right) < 0 \end{aligned}$$

where *q*=1, 2, … n_p; *i*=1, 2,…*N*. w_{qi}^k is the element inertia weight *i* of particle *q* in iteration *k*. From (10), if v_{qi}^k and $\left(x_{i,gbest}^k - x_{qi}^k\right)$ move in the same direction, the value of w_{qi}^k employed will be linearly decreasing to prevent the particles from flying past the target position during the flight. Otherwise, the value of w_{qi}^k will be kept without decreasing to facilitate a free movement of particles in the search space. Instead of maximum iteration count $iter_{\max}$, another parameter *Z* is designed to further provide a well-balanced mechanism between global and local exploration abilities. It is obvious that the value of *Z* is an important factor to control the linearly decreasing dynamic parameter framework descending from $w_{\max}$ to $w_{\min}$. Suitable selection of *Z* provides a balance between global and local explorations, thus requiring less iterations on average to find a sufficiently optimal solution. The main attractive feature of inertia weight mechanism described above is to monitor the weights of a particle, which were linearly decreased in general applications, to avoid storing too many similar particles at the end of the optimization process. The significance of control of inertia weight *w* in the PSO algorithm is also retained to increase the possibility of occurrence of escaping from local optimal solutions. Update the velocities and positions of the particles. The velocity of the particle vector is updated according to (11).

$$\begin{aligned} V_{qi}^{k+1} = w_{qi} \times V_{qi}^k + C_1 \times rand \times (pbest_{qi}^k - X_{qi}^k) + \\ C_2 \times rand \times (gbest_i^k - X_{qi}^k) \end{aligned} \qquad (11)$$

Eq. (12) is applied to update the position of the particles.

$$X_{qi}^{k+1} = X_{qi}^k + V_{qi}^{k+1}; \quad q = 1,2,...,n_p; i = 1,2,...N. \qquad (12)$$

The steps of NPSO as implemented for linear phase FIR LP filter design are as follows:
Step 1: Initialization: Population (swarm size) of particle vectors, n_P=120; maximum iteration cycles=200; number of filter coefficients (h(n)); filter order, N=20; fixing values of C_1, C_2 as 2.05; minimum and maximum values of filter coefficients, hmin=-2, hmax= 2; number of samples=128; $\delta_p = 0.1$, $\delta_s = 0.01$; initialize the parameter Z; maximum number of iterations and compute the inertia weight as per (10); initialization of the velocities of all particle vectors.
Step 2: Generate initial particle vectors of filter coefficients (N/2+1) randomly with limits; Computation of initial fitness values of the total population, n_P.
Step 3: Computation of population based minimum error fitness value and computation of the personal best solution vectors (pbest), group best solution vector (gbest). The error fitness function is an index to evaluate the error fitness of the particles. Eq. (6) shows the fitness function of the filter design problem.
Step 4: Record and update the best values. The two best values are recorded in the searching process. Each particle keeps track of its coordinate in the solution space that is associated with the best solution it has reached so far. This value is recorded as *pbest*. Another best value to be recorded is *gbest*, which is the overall best value obtained so far by any particle.
Step 5: Update the velocities as per (11); updating the particles as per (12) and checking against the limits of the filter coefficients; finally, computation of the updated error fitness values of the particles and population based minimum error fitness value.
Step 6: Check the end condition. If it is reached, the algorithm stops, otherwise, repeat steps 3-5. In this study, the end condition of NPSO is either the convergence of minimum error fitness values is met or the maximum number of iterations is reached.
Finally, hgbest is the particle vector of optimal filter coefficients (nvar/2 +1); Form complete (nvar+1) coefficients by copying (because the filter has linear phase) before getting the optimal frequency spectrum.

4 Results and Discussions

4.1 Analysis of Magnitude Response of Low pass FIR Filter

This section presents the simulations performed in MATLAB 7.5 for the design of FIR LP filter. The filter order (N) is taken as 20, which results in the number of coefficients as 21. The sampling frequency is taken to be f_s = 1Hz. The number of frequency samples is 128. Each algorithm is run for 50 times to obtain its best results.
Table 2 shows the best chosen parameters for RGA, PSO, and NPSO, respectively.

Table 2. RGA, PSO, NPSO Parameters

Parameters	RGA	PSO	NPSO
Population size	120	25	25
Iteration Cycle	800	350	200
Crossover rate	1	-	-
Crossover	Two Point	-	-
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
w_{max}	-	1.0	1.0
w_{min}	-	0.4	0.4
Z	-	-	100

The parameters of the filter to be designed using the NPSO are: pass band ripple (δ_p) = 0.1, stop band ripple (δ_s) = 0.01. For the LP filter, pass band (normalized) edge frequency (ω_p) = 0.45; stop band (normalized) edge frequency (ω_s) = 0.55; transition width=0.1. The filter has order 20. Table 3 shows the optimized filter coefficients obtained for FIR LP filter, respectively, using the RGA, PSO, and the NPSO individually.

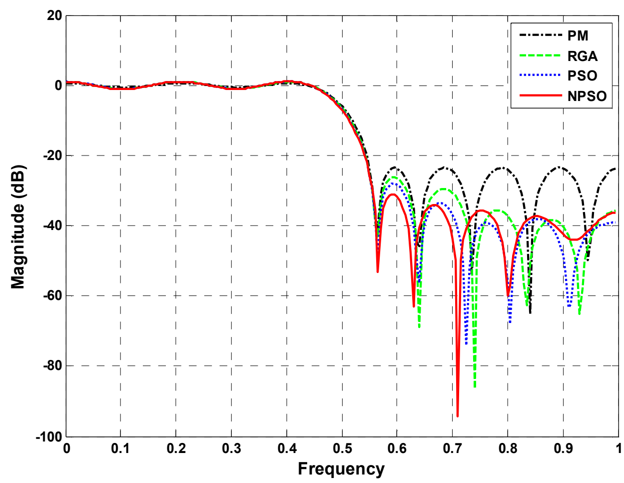

Fig. 1 dB plots for the FIR LP filter of order 20.

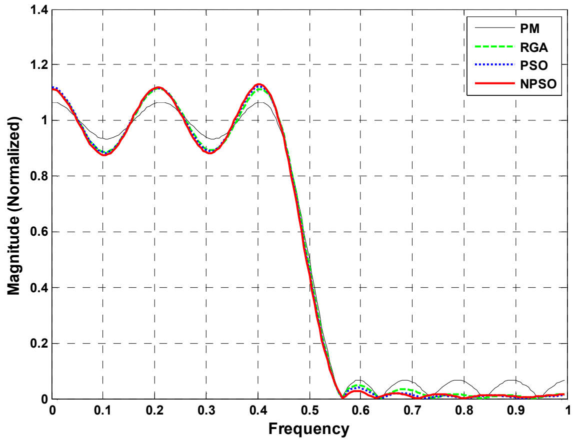

Fig. 2 Normalized plots for the FIR LP filter of order 20.

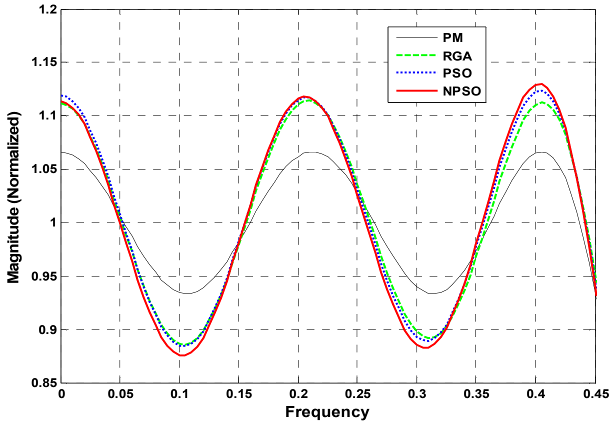

Fig. 3 Normalized Pass band ripple plots for the FIR LP filter of order 20.

Table 3. Optimized coefficients of the FIR LP filter of order 20

h(N)	RGA	PSO	NPSO
h(1)=h(21)	0.020644508012550	0.025116793352393	0.028302685047878
h(2)=h(20)	0.048721413185106	0.047219259300299	0.046719898206481
h(3)=h(19)	0.005868601564964	0.003546242723169	0.001859568002839
h(4)=h(18)	-0.040966865300227	-0.040094047283599	-0.040498019404735
h(5)=h(17)	-0.000863506780022	-0.000520432067214	0.001013961054997
h 6)=h(16)	0.059796031265565	0.060907207778672	0.058649649609037
h(7)=h(15)	-0.001408842862974	-0.001759240756773	-0.000384440011894
h(8)=h(14)	-0.103117834700311	-0.103613994946693	-0.105609977954995
h(9)=h(13)	-0.000440644382089	0.000627623037422	0.001355134966428
h(10)=h(12)	0.317600651261946	0.318119036548684	0.315197573380121
h(11)	0.500018538901557	0.500018548921576	0.500118548901356

Table 4. Other comparative results of performance parameters of all algorithms for the FIR LP filter of order 20

Algorithm	FIR LP filter of order 20		
	Maximum, average stop band ripple (normalized)	Transition width (normalized)	Execution Time for 100 cycles (s)
PM	0.06648, 0.06624	0.1000	-
RGA	0.04943, 0.02559	0.0950	6.2846
PSO	0.03954, 0.01901	0.0980	4.8777
NPSO	0.0279, 0.01862	0.0904	5.3417

Table 5. Statistical parameters of stop band attenuation for different algorithms for the FIR LP filter

Algorithm	FIR LP filter of order 20							
	Pass band ripple (normalized)				Stop band Attenuation (dB)			
	Maximum	Mean	Variance	Standard Deviation	Maximum	Mean	Variance	Standard Deviation
PM	0.067	0.066	0.00000022	0.00047	23.55	23.58	0.00165	0.0406
RGA	0.114	0.113	0.000001	0.001	26.12	33.066	20.578	4.5363
PSO	0.123	0.1197	0.00000622	0.0025	28.06	35.602	18.5511	4.3071
NPSO	0.118	0.12	0.000051	0.0071	31.09	34.884	4.70058	2.168

Table 6. Comparison of NPSO's Results with Other Reported Results

Model	Parameter					
	Filter type	Order	Maximum stop band attenuation (dB)	Maximum pass band ripple (normalized)	Maximum stop band ripple (normalized)	Transition width
Karaboga [11]	Low pass	20	NR*	>0.08	>0.09	>0.16
Liu *et al.* [12]	Low pass	20	NR*	0.04	>0.07	>0.06
Najjarzadeh *et al.* [16]	Low Pass	33	<29dB	NR*	NR*	NR*
Ababneh *et al.* [14]	Low pass	30	<30dB (Approx.)	0.15	0.031	0.05
Sarangi *et al.* [21]	Low Pass	20	< 27dB	>0.1	>0.06	>0.15
Bipul Luitel *et al.* [17]	Low Pass	20	<27 dB	0.291	0.270	>0.13
NPSO	Low Pass	20	31.09	0.118	0.0279	0.0904

NR* means not reported in the referred literature

Table 4 shows the comparative results of performance parameters in terms of maximum and average stop band ripple (normalized), transition width (normalized) for LP filter using PM, RGA, PSO, and the NPSO, respectively. It is noticed that for a narrower transition width, the NPSO results in the best stop band attenuation among all algorithms for all types of filters.

Table 5 shows the comparison of the maximum stop band attenuations achieved for LP filters using PM, RGA, PSO, NPSO, respectively. Table 5 shows that the maximum stop band attenuation achieved for the LP filter using the NPSO is 31.09 dB. It is observed from Table 5 that the NPSO achieves the best stop band attenuation, as compared to those of PM, RGA and PSO FIR LP filter of order 20.

Table 5 also summarizes maximum, mean, variance and standard deviation for pass band ripple (normalized) and stop band attenuation in dB for the designed LP filter using PM, RGA, PSO, and the NPSO, respectively. From Table 5, it is observed that the maximum pass band ripple (normalized) obtained using NPSO is 0.118. Table 6 summarizes the comparison of NPSO based results with other reported results. NPSO results in 31.09 dB stop band attenuation, maximum pass band ripple (normalized) = 0.118, maximum stop band ripple (normalized) = 0.0279, transition width = 0.0904. The simulation results of Sarangi *et al.* [25], show that for the LP filter of order 20, the maximum stop band attenuation (dB) is less than 27dB (approx.), maximum pass band ripple (normalized) is more than 0.1, maximum stop band ripple (normalized) is more than 0.06, transition width is more than 0.15. It is observed from Table 6 that the simulation results obtained for filter order 20 using NPSO are much better than the other reported results.

Figs. 1-4 show the magnitude response of the LP filter using NPSO. The magnitude response in dB is plotted in Fig. 1 for low pass filter. The normalized magnitude response is shown in Fig. 2. Fig. 3 shows the normalized pass band ripple for FIR LP filter of order 20. Fig. 4 shows the plots of normalized stop band ripple.

From the above figures and tables, it is observed that NPSO results in better magnitude response (dB), normalized magnitude response, normalized pass band ripple and normalized stop band ripple for LP filter, as compared to PM, RGA and PSO algorithms.

4.2 convergence profiles of RGA, PSO and NPSO

In order to compare the algorithms in terms of the error fitness value, Figs. 5-7 show the convergences of error fitness values obtained when RGA, PSO, and the NPSO, respectively, are employed.

NPSO converges to much lower error fitness value as compared to RGA and PSO which yield suboptimal higher values of error fitness values. RGA converges to the minimum error fitness value of 3.109 in 38.1096s; PSO converges to the minimum error fitness value of 2.479 in 19.5108s; whereas, the NPSO converges to the minimum error fitness value of 1.01 in 9.7218s. The above-mentioned execution times may be verified from Figs. 5-7 and Table 4. For the designed FIR LP filter, NPSO converges to the least minimum error fitness value in finding the optimum filter coefficients with lesser number of iteration cycles.

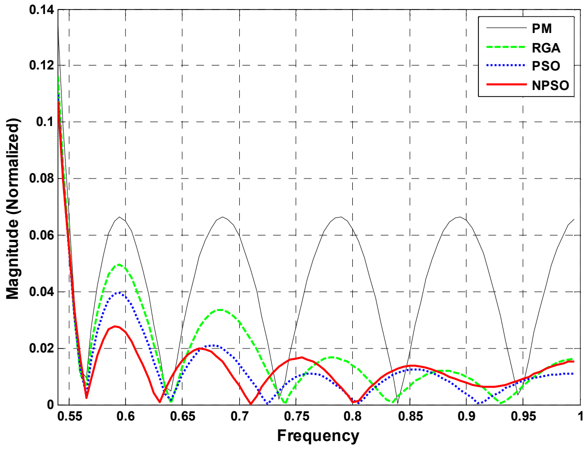

Fig. 4 Normalized Stop band ripple plots for the FIR LP filter of order 20.

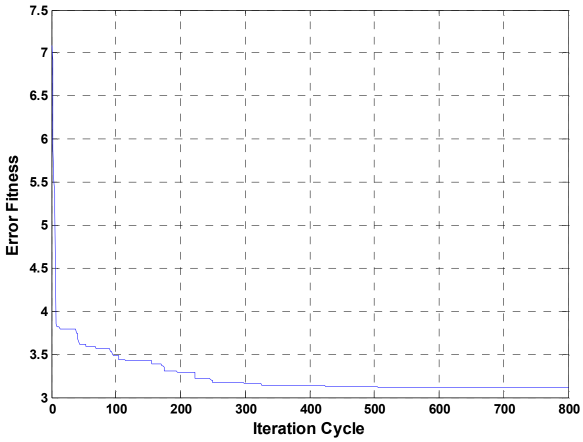

Fig. 5 Convergence Profile for RGA in case of FIR LP Filter of Order 20.

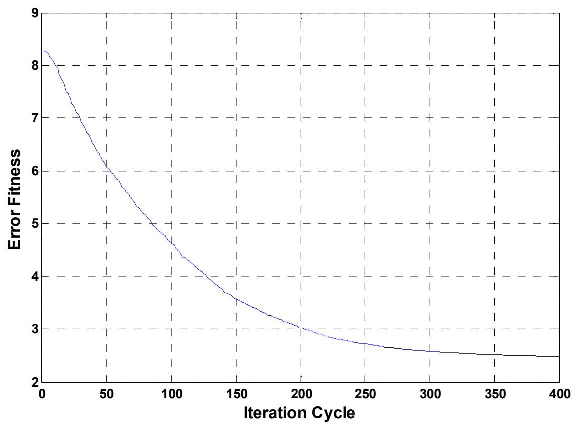

Fig.6 Convergence Profile for conventional PSO in case of FIR LP Filter of Order 20.

With a view to the above fact, it may finally be inferred that the performance of NPSO is the best among all algorithms. All optimization programs are run in MATLAB 7.5 version on core (TM) 2 duo processor, 3.00 GHz with 2 GB RAM.

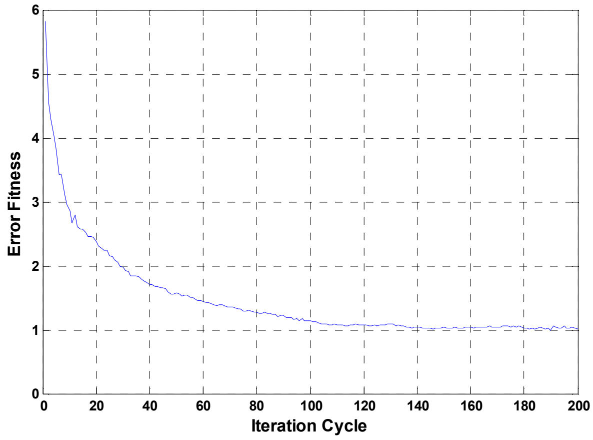

Fig. 7 Convergence Profile for NPSO in case of FIR LP Filter of Order 20.

5 Conclusion

In this paper, a novel particle swarm optimization algorithm (NPSO) is applied to the solution of the constrained, multi-modal FIR low pass filter design problem with optimal filter coefficients. Comparison of the results of PM, RGA, PSO, and NPSO algorithm has been made. It is revealed that NPSO has the ability to converge to the best quality near optimal solution and possesses the best convergence characteristics in much less execution times among the algorithms. The simulation results clearly indicate that NPSO demonstrates the best performance in terms of magnitude response, minimum stop band ripple and maximum stop band attenuation with the narrowest transition width. Thus, the NPSO may be used as a good optimizer for obtaining the optimal filter coefficients in any practical digital filter design problem of digital signal processing systems.

References:

[1] L. Litwin, FIR and IIR digital filters, *IEEE Potentials*, pp. 28–31, 2000.

[2] T.W. Parks, C.S. Burrus, *Digital Filter Design*, Wiley, New York, 1987.

[3] T.W. Parks, J.H. McClellan, Chebyshev approximation for non recursive digital filters with linear phase, *IEEE Trans. Circuits Theory*, CT-19, 1972, pp. 189–194.

[4] J.H. McClellan, T.W. Parks, L.R. Rabiner, A computer program for designing optimum FIR linear phase digital filters, *IEEE Trans. Audio Electro acoust.*, AU-21, pp. 506–526, 1973.

[5] L.R. Rabiner, Approximate design relationships for High-pass FIR digital filters,

IEEE Trans. Audio Electro acoust., AU-21, 1973, pp. 456–460.

[6] O. Herrmann, W. Schussler, Design of non-recursive digital filters with linear phase, *Electron. Lett.*, 6, 1970, pp.329–330.

[7] N.E. Mastorakis, I.F. Gonos, M.N.S Swamy, Design of Two Dimensional Recursive Filters Using Genetic Algorithms, *IEEE Transaction on Circuits and Systems* I - *Fundamental Theory and Applications*, 50, 2003, pp. 634–639.

[8] S.U. Ahmad, A. Antoniou, A genetic algorithm approach for fractional delay FIR filters, *IEEE International Symposium on Circuits and Systems*, pp.2517-2520, 2006.

[9] Hung-Ching Lu, Shian-Tang Tzeng, Design of arbitrary FIR log filters by genetic algorithm approach, *Signal Processing*, 2000, 80, pp. 497-505.

[10] S. Chen, IIR Model Identification Using Batch-Recursive Adaptive Simulated Annealing Algorithm, *6th Annual Chinese Automation and Computer Science Conference*, 2000, pp.151–155.

[11] D. Karaboga, D.H. Horrocks, N. Karaboga, A. Kalinli, Designing digital FIR filters using Tabu search algorithm, *IEEE International Symposium on Circuits and Systems*, 1997, vol.4, pp.2236-2239.

[12] G. Liu, Y.X. Li, and G.He, Design of Digital FIR Filters Using Differential Evolution Algorithm Based on Reserved Gene, *IEEE Congress on Evolutionary Computation,* July 2010, pp. 1-7.

[13] N. Karaboga, A new design method based on artificial bee colony algorithm for digital IIR filters, *Journal of the Franklin Institute*, 2009, 346, (4), pp.328–348

[14] J.I. Ababneh, M. H. Bataineh, Linear phase FIR filter design using particle swarm optimization and genetic algorithms, *Digital Signal Processing*, 18, 657–668, 2008.

[15] J. Kennedy and R. Eberhart, Particle swarm optimization, *Proceedings of the IEEE International Conference on Neural Networks*, vol. 4, 1995, pp. 1942-1948.

[16] M. Najjarzadeh, A.Ayatollahi, FIR Digital Filters Design: Particle Swarm Optimization Utilizing LMS and Minimax Strategies, *International symp. on Signal Processing and Information Technology*, ISSPIT 2008, pp. 129-132.

[17] B. Luitel, G. K. Venayagamoorthy, Differential Evolution Particle Swarm Optimization for Digital Filter Design, *IEEE Congress on Evolutionary Computation* (CEC 2008), PP. 3954-3961, 2008.

[18] J. Sun., B.Feng and W.B.Xu, Particle Swarm Optimization with Particles Having Quantum Behaviour, in *Proc. Congress on Evolutionary Computation*, 2004, pp. 325-331.

[19] Wei Fang, Jun Sun, Wenbo Xu, and Jing Liu, FIR Digital Filters Design Based on Quantum-behaved Particle Swarm Optimization, *First International Conference on Innovative Computing, Information and Control*, 2006, vol 1, pp. 615-619.

[20] B. Luitel, G.K. Venayagamoorthy, Particle Swarm Optimization with Quantum Infusion for the Design of Digital Filters, *Swarm Intelligence Symposium*, 2008, pp. 1-8.

[21] A. Sarangi, R.K. Mahapatra, S.P. Panigrahi, DEPSO and PSO-QI in digital filter design, Expert Systems with Applications, 2011, vol. 38, No.9, pp.10966-10973.

[22] S. Mandal, S. P. Ghoshal, R. Kar, D. Mandal, Optimal Linear Phase FIR Band Pass Filter Design using Craziness Based Particle Swarm Optimization Algorithm, *Journal of Shanghai Jiaotong University* (*Science*), Springer. Vol. 16, No. 6, pp. 696-703, 2011

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

[24] 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, 2009.

[25] Chun-Lung Chen, Rong-Mow Jan, Tsung-Ying Lee, and Cheng-Hsiung Chen, A Novel Particle Swarm Optimization Algorithm Solution of Economic Dispatch With Valve Point Loading, *Journal of Marine Science and Technology,* Vol. 19, No. 1, pp. 43-51, 2011.

[26] D. Mandal, S. P. Ghoshal, and A. K. Bhattacharjee, Application of Evolutionary Optimization Techniques for Finding the Optimal set of Concentric Circular Antenna Array, *Expert Systems with Applications*, vol. 38, pp. 2942-2950, 2010.