

Swarm Intelligence based Optimal Linear Phase FIR High Pass Filter Design using Particle Swarm Optimization with Constriction Factor and Inertia Weight Approach

Sangeeta Mandal, Rajib Kar and Durbadal Mandal, Sakti Prasad Ghoshal

Abstract—In this paper, an optimal design of linear phase digital high pass finite impulse response (FIR) filter using Particle Swarm Optimization with Constriction Factor and Inertia Weight Approach (PSO-CFIWA) has been presented. 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. The conventional gradient based optimization techniques are not efficient for digital filter design. Given the filter specifications to be realized, the PSO-CFIWA algorithm generates a set of optimal filter coefficients and tries to meet the ideal frequency response characteristic. In this paper, for the given problem, the designs of the optimal FIR high pass filters of different orders have been performed. The simulation results have been compared to those obtained by the well accepted algorithms such as Parks and McClellan algorithm (PM), genetic algorithm (GA). The results justify that the proposed optimal filter design approach using PSO-CFIWA outperforms PM and GA, not only in the accuracy of the designed filter but also in the convergence speed and solution quality.

Keywords—FIR Filter; PSO-CFIWA; PSO; Parks and McClellan Algorithm, Evolutionary Optimization Technique; Magnitude Response; Convergence; High Pass Filter

I. INTRODUCTION

DIGITAL filters are used in numerous applications from control systems, systems for audio and video processing, and communication systems to systems for medical applications to name just a few. They can be implemented in hardware or software and can process both real-time and off-line (recorded) signals. Digital filters in hardware form can now routinely perform tasks that were almost exclusively performed by analog systems in the past, whereas, digital filters can be implemented in software using low-level or user-friendly high-level programming languages. Nowadays digital filters can be used to perform many filtering tasks which in the not so distant past were performed almost exclusively by analog filters and are replacing the traditional role of analog filters in many applications.

Sangeeta Mondal and Shakti Prasad Ghoshal are with the Department of Electrical Engg., National Institute of Technology, Durgapur, West Bengal, 713029 (e-mail: spghoshalnitdgp@gmail.com).

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

Beside the inherent advantages, such as, high accuracy and

reliability, small physical size, and reduced sensitivity to component tolerances or drift, digital implementations allow one to achieve certain characteristics not possible with analog implementations such as exact linear phase and multi-rate operation. Digital filtering can be applied to very low frequency signals, such as those occurring in biomedical and seismic applications very efficiently. In addition, the characteristics of digital filters can be changed or adapted by simply changing the content of a finite number of registers, thus multiple filters are usually used to discriminate a frequency or a band of frequencies from a given signal which is normally a mixture of both desired and undesired signals. The undesired portion of the signal commonly comes from noise sources such as power line hum etc. or other signals which are not required for the current application. There are mainly two types of filter algorithms. They are Finite Impulse Response filter (FIR) and Infinite Impulse Response filter (IIR). In case of a FIR filter, the response due to an impulse input will decay within a finite time. But, for IIR filter, the impulse response never dies out. It theoretically extends to infinity. FIR filters are commonly known as non-recursive filters and IIR filters are known as recursive filters. These names came from the nature of algorithms used for these filters. Implementation of FIR filters is easy, but it is slower when compared to IIR filters. Though IIR filters are fast, practical implementation is a bit tough compared to FIR filters [1]. 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, the FIR filter can be guaranteed to have linear phase. Finite impulse response (FIR) digital filters are known to have many desirable features such as guaranteed stability, the possibility of exact linear phase characteristic at all frequencies and digital implementation as non-recursive structures. Linear phase FIR filters are also required when time domain specifications are given [2]. Traditionally, different techniques exist for the design of digital filters. Out 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 pass band and stop band, stop band 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. The most frequently used method for the design of exact linear phase weighted Chebyshev FIR digital filter is the one based on the Remez-exchange algorithm proposed by Parks and McClellan [3]. Further improvements in their results have been reported in [4]. The main limitation of this procedure is that the relative values of the amplitude error in the frequency bands are specified by means of the weighting function, and not by the deviations themselves. Therefore, in case of designing high-pass filters with the given stop band deviation, filter length and cut-off frequency, the program has to be iterated many times [5]. A number of models have been developed for the finite impulse response (FIR) filter techniques and design methods. This is a thrust research area, aiming at obtaining more general and innovative techniques that are able to solve and/or optimize new and complex engineering problems [6]. The trade-off has to be made by the designer on one or the other of the design specifications. So, evolutionary methods have been employed in the design of digital filters to design with better parameter control and to better approximate the ideal filter [7]. Different heuristic optimization algorithms such as genetic algorithm (GA) [7] simulated annealing algorithms [8] etc. have been widely used to the synthesis of design methods capable of satisfying constraints which would be unattainable. When considering global optimization methods for digital filter design, the GA seems to be the promising one. Filters designed by GA have the potential of obtaining near global optimum solution. Although standard GA (mostly referred to as Real Coded GA (RGA)) have a good performance for finding the promising regions of the search space, they are inefficient in determining the local minimum in terms of convergence speed and solution quality [9-10].

The approach detailed in this paper takes advantage of the power of the stochastic global optimization technique called particle swarm optimization. Although the algorithm is adequate to applications in any kind of parameterized filters, the authors have chosen to focus on real-coefficient FIR filters, in view of their importance in engineering practice. Particle Swarm Optimization (PSO) is an evolutionary algorithm developed by Eberhart et al. [11-12]. Several attempts have been made towards the optimization of the FIR Filter [10] using PSO algorithm. The PSO is simple to implement and its convergence may be controlled via few parameters. The limitations of the conventional PSO are that it may be influenced by premature convergence and stagnation problem [13-14]. In order to overcome these problems, the PSO algorithm has been modified in this paper and is employed for FIR filter design.

This paper describes an alternative technique for the FIR high pass digital filter design using Particle Swarm Optimization with Constriction Factor and Inertia Weight Approach (PSO-CFIWA). PSO-CFIWA algorithm tries to find the best coefficients that closely match the ideal

frequency response. Based upon the improved PSO approach, this paper presents a good and comprehensive set of results, and states arguments for the superiority of the algorithm. Simulation result demonstrates the effectiveness and better performance of the proposed designed method.

The rest of the paper is arranged as follows. In section II, the FIR filter design problem is formulated. Section III briefly discusses on the algorithm of GA, classical PSO and the PSO-CFIWA algorithm. Section IV describes the simulation results obtained for high pass FIR digital filter using PM algorithm, GA and the proposed approach. Finally, section V concludes the paper.

II. HIGH PASS FIR FILTER DESIGN

A digital FIR filter is characterized by,

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

where N is the order of the filter which has $(N+1)$ number of coefficients. $h(n)$ is the filter's impulse response. It is calculated by applying an impulse signal at the input. 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 most widely used FIR with $h(n)$ as even symmetric and the order is even. The length of $h(n)$ is $N+1$ and the number of coefficients is also $N+1$. In the algorithm, the individual represents $h(n)$. In each iteration, these individuals are updated. Fitness of particles is calculated using the new coefficients. In each iteration, this fitness is used to improve the search and result obtained after a certain number of iterations or after the error is below a certain limit is considered to be the optimal result. Because its coefficients are symmetrical, the dimension of the problem reduces by a factor of 2. The $(N+1)/2$ coefficients are then flipped and concatenated to find the required $N+1$ coefficients. The least error is used to evaluate the fitness of the individual. It takes the error between the frequency response of the ideal and the actual filter. An ideal filter has a magnitude of one on the pass band and a magnitude of zero on the stop band. So, the error for this fitness function is the difference between the magnitudes of this filter and the filter designed using the evolutionary algorithms GA and PSO-CFIWA. The individuals that have lower error values represent the better filter i.e., the filter with better frequency response. Various filter parameters which are responsible for the optimal filter design are the stop band and pass band normalized frequencies (ω_s, ω_p) , the pass band and stop band ripples δ_p and δ_s , the stop band attenuation and the transition width. These parameters are mainly decided by the filter coefficients which are evident from transfer function in (1).

Several scholars have investigated and developed algorithms in which N , δ_p , and δ_s are fixed while the remaining

parameters are optimized [6]. Other algorithms were originally developed by Parks and McClellan (PM) in which N , w_p , w_s , and the ratio δ_p/δ_s are fixed [3]. In this paper, swarm and evolutionary optimization algorithms are applied in order to obtain the actual filter response as close as possible to the ideal response. Now for (1), coefficient vector $\{h_0, h_1, \dots, h_n\}$ is represented in $N+1$ dimensions. The particles are distributed in a D dimensional search space, where $D = N+1$ for the case of FIR filter.

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

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

where $w_k = \frac{2\pi k}{N}$; $H(e^{j\omega_k})$ is the Fourier transform complex vector. This is the FIR filter frequency response. The frequency is sampled in $[0, \pi]$ with N points; the positions of the particles in this D dimensional search space represent the coefficients of the transfer function. In each iteration, these particles find a new position, which is the new set of coefficients. Fitnesses of particles are calculated using the new coefficients. These fitnesses are 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. Different kinds of fitness functions have been used in different literatures. An error function given by (3) is the approximate error used in Parks–McClellan algorithm for filter design [3].

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

where $G(\omega)$ is the weighting function used to provide different weights for the approximate errors in different frequency bands, $H_d(e^{j\omega})$ is the frequency response of the desired filter and is given as,

$$H_d(e^{j\omega_k}) = 1 \quad \text{for } 0 \leq \omega \leq \omega_c; \\ = 0 \quad \text{otherwise} \quad (4)$$

and $H_i(e^{j\omega})$ is the frequency response of the approximate filter.

$$H_d(\omega) = [H_d(\omega_1), H_d(\omega_2), H_d(\omega_3), \dots, H_d(\omega_K)]^T \quad \text{and} \\ H_i(\omega) = [H_i(\omega_1), H_i(\omega_2), H_i(\omega_3), \dots, H_i(\omega_K)]^T$$

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 [18].

The error to be minimized is defined as:

$$J_1 = \max_{\omega \leq \omega_p} (|E(\omega)| - \delta_p) + \max_{\omega \geq \omega_s} (|E(\omega)| - \delta_s) \quad (5)$$

where δ_p and δ_s are the ripples in the pass band and stop band; and ω_p and ω_s are pass band and stop band normalized cut off frequencies, respectively. (5) represents the fitness function to be minimized using the evolutionary algorithms.

The algorithms try to minimize this error and thus increase the fitness. Since the coefficients of the linear phase filter are matched, meaning the first and the last coefficients are the same; the dimension of the problem is reduced by one-half. By only determining one half of the coefficients, the filter could be designed. This greatly reduced the computational complexity of the algorithms.

III. EVOLUTIONARY TECHNIQUES USED

A. Genetic Algorithm (GA)

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.

Steps of RGA as implemented for optimization of $h(n)$ coefficients are [15, 16]:

- Initialization of real chromosome strings 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* of each string.
- Selection of elite strings in order of increasing *Error* 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.
- The iteration stops when the maximum number of cycles is reached. The grand minimum *Error* and its corresponding chromosome string or the desired solution vector is finally obtained.

B. 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 [17-18]. PSO is less susceptible to getting trapped on local optima unlike GA, Simulated Annealing, etc. Eberhart and Shi [10] developed PSO concept similar to the behaviour of a swarm of birds. PSO is developed through simulation of bird flocking in multidimensional space. Bird flocking optimizes a certain objective function. Each particle (bird) knows its best value so far (pbest). This information corresponds to personal experiences of each particle. Moreover, each particle knows the best value so far in the group (gbest) among pbests. 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 (6).

$$V_i^{(k+1)} = w * V_i^{(k)} + C_1 * rand_1 * (pbest_i^{(k)} - S_i^{(k)}) + C_2 * rand_2 * (gbest^{(k)} - S_i^{(k)}) \quad (6)$$

where $V_i^{(k)}$ is the velocity of i^{th} particle at k^{th} 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; $S_i^{(k)}$ is the current position of i^{th} particle vector at k^{th} iteration; $pbest_i^{(k)}$ is the personal best of i^{th} particle vector at k^{th} iteration; $gbest^{(k)}$ is the group best of the group at k^{th} iteration. The searching point in the solution space may be modified by the following equation:

$$S_i^{(k+1)} = S_i^{(k)} + V_i^{(k+1)} \quad (7)$$

The first term of (6) is the previous velocity of the particle vector. The second and third terms are used to change the velocity of the particle. Without the second and third terms, the particle 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.

C. Particle Swarm Optimization with Constriction Factor and Inertia Weight Approach (PSO-CFIWA)

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 (PSO-CFIWA). For Particle Swarm Optimization with Constriction Factor and Inertia Weight Approach (PSO-CFIWA) [12, 13], the velocity of (6) is manipulated in accordance with (8).

$$V_i^{k+1} = CFa \times (w^{k+1} * V_i^k + C_1 * rand_1 * (pbest_i - S_i^k) + C_2 * rand_2 * (gbest - S_i^k)) \quad (8)$$

Normally, $C_1=C_2=1.5-2.05$ and Constriction Factor (CFa) is given in (9).

$$CFa = \frac{2}{|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}|} \quad (9)$$

where

$$\varphi = C_1 + C_2, \text{ and } \varphi > 4.$$

For $C_1=C_2=2.05$, the computed value of $CFa = 0.73$.

The best values of C_1 , C_2 , and CFa are found to vary with the designs of filters.

Inertia weight (w^{k+1}) at $(k+1)^{th}$ cycle is as given in (10).

$$w^{k+1} = w_{\max} - \frac{w_{\max} - w_{\min}}{k_{\max}} \times (k + 1) \quad (10)$$

where

$w_{\max}=1.0$; $w_{\min}=0.4$; k_{\max} = Maximum number of iteration cycles. The solution updating is the same as (7).

The design aim in this paper is to obtain the optimal combination of the filter coefficients, so as to acquire the maximum stop band attenuation with least transition width increment. The values of the parameters used for the PSO-CFIWA technique are given in Table I.

IV. RESULTS AND DISCUSSIONS

A. Analysis of Magnitude Response of High Pass FIR filters

The MATLAB simulation has been performed extensively to realize the high pass FIR filter of the orders of 30 and 40. Hence, the lengths of the filter coefficients are 31 and 41, respectively. The sampling frequency has been chosen as $f_s = 1\text{Hz}$. Also, for all the simulations the sampling number is taken as 128. The parameters of the high pass filter to be designed are as follows:

- Pass band ripple (δ_p) = 0.01
- Stop band ripple (δ_s) = 0.001
- Pass band normalized cutoff frequency (ω_p) = 0.45
- Stop band normalized cutoff frequency (ω_s) = 0.40

The control parameters' values of RGA and PSO-CFIWA used in this work are given in Table I and Table II, respectively. Each algorithm is run for 30 times to get the best solutions. Figs. 1 and 2 show the gain plot and the magnitude plot, respectively, for the high pass FIR filter of the order of 30. The best optimized coefficients for the designed filters with the order of 30 and 40 have been calculated by both RGA and PSO-CFIWA and given in Tables III and IV, respectively. Figs. 3 and 4 show the gain plot and the magnitude plot, respectively, for the high pass FIR filter of the order of 40. From the figures, it is evident the proposed filter design approach PSO-CFIWA produces higher stop band attenuation and smaller ripple compared to that of PM and RGA.

TABLE I
GA PARAMETERS

Parameter	Value
Population Size	120
Crossover rate	1
Crossover	Two Point Crossover
Maximum Iteration Cycle	500
Mutation rate	0.01
Mutation	Gaussian Mutation
Selection	Roulette Wheel
Selection Probability	1/3

TABLE II
 PSO CFIWA PARAMETERS

Parameter	Value
Population Size	120
Maximum Iteration Cycle	500
CI & C2	2.05
<i>C_{Fa}</i>	0.73
W_{min}	0.4
W_{max}	1.0

The filters designed by the PSO-CFIWA algorithm have sharper transition band responses than that produced by GA algorithms. For the stop band region, the filters designed by the PSO-CFIWA method results in the improved responses than the other. In some situations, it is found that the performance of GA is inferior to PM. But the performance of PSO CFIWA is consistently better than the performances of PM and GA algorithms.

TABLE III
 OPTIMIZED COEFFICIENTS OF FIR FILTER OF ORDER 30

h(N)	RGA	PSOCFIWA
h(1)=h(31)	-0.021936936290133	-0.018052101196575
h(2)=h(30)	0.009894574634720	0.004231156408695
h(3)=h(29)	0.024447574147984	0.021081909684010
h(4)=h(28)	0.003141365867601	0.009402420404598
h(5)=h(27)	-0.026453238596167	-0.021483927291159
h(6)=h(26)	-0.016376104143824	-0.022083346009252
h(7)=h(25)	0.014726120434996	0.014161063279814
h(8)=h(24)	0.038071092404889	0.036013223228672
h(9)=h(23)	-0.009608574065867	-0.006596745264573
h(10)=h(22)	-0.047629842434778	-0.048948345038178
h(11)=h(21)	-0.016700813278929	-0.019709723088824
h(12)=h(20)	0.062358850767844	0.067088249738827
h(13)=h(19)	0.076547973471596	0.079238882383898
h(14)=h(18)	-0.070167621281884	-0.075046158009142
h(15)=h(17)	-0.310940982209537	-0.308270199702038
h(16)	0.575267503881518	0.574597842780004

TABLE IV
 OPTIMIZED COEFFICIENTS OF FIR FILTER OF ORDER 40

h(N)	RGA	PSOCFIWA
h(1)=h(41)	-0.018990542022313	-0.016920267413033
h(2)=h(40)	-0.003486818029446	-0.004311666214144
h(3)=h(39)	0.012653303611510	0.007641129696056
h(4)=h(38)	0.013040421009306	0.011644320663902
h(5)=h(37)	-0.015710646633533	-0.012432739220420
h(6)=h(36)	-0.014178393682391	-0.015075135419202
h(7)=h(35)	0.009159494494393	0.003636518887090
h(8)=h(34)	0.026096641613082	0.017249790395627
h(9)=h(33)	0.005847288800364	0.008750699471996
h(10)=h(32)	-0.022277408243303	-0.017680440699840
h(11)=h(31)	-0.015854797525643	-0.015248444335953
h(12)=h(30)	0.021594299149472	0.016693464410608
h(13)=h(29)	0.037298943972098	0.031204651302548
h(14)=h(28)	-0.004846279517654	-0.001295138938641
h(15)=h(27)	-0.051959379423136	-0.047807169035042
h(16)=h(26)	-0.022761276938976	-0.021827715608804
h(17)=h(25)	0.063353478297367	0.062728719345397
h(18)=h(24)	0.073443186781609	0.082248589139126
h(19)=h(23)	-0.076058321957523	-0.074389198770264
h(20)=h(22)	-0.309541958811529	-0.307060847190878
h(21)	0.575741314093542	0.575719784658169

A. Comparative effectiveness and convergence profiles of RGA and PSO-CFIWA

In order to compare the algorithms in terms of the convergence speed, Fig. 5 shows the plot of minimum error values against the number of iteration cycles when RGA is employed. Fig. 6 shows the plot of minimum error values against the number of iteration cycles when the proposed new PSO is employed. The convergence profiles have been shown for the filter order of 30. A similar plot may be obtained for the FIR filter of order 40. From the figures drawn for this filter, it is seen that the PSOCFIWA algorithm is significantly faster than the RGA algorithm for finding the optimum filter. The new PSO converges to a much lower fitness in lesser number of iterations. Further, RGA yields suboptimal higher values of Error but PSOCFIWA yields near optimal (least) Error values. With a view to the above fact, it may finally be inferred that the performance of PSOCFIWA technique is better as compared to RGA in designing the optimal FIR filter. All optimization programs are run in MATLAB 7.5 version on core (TM) 2 duo processor, 3.00 GHz with 2 GB RAM.

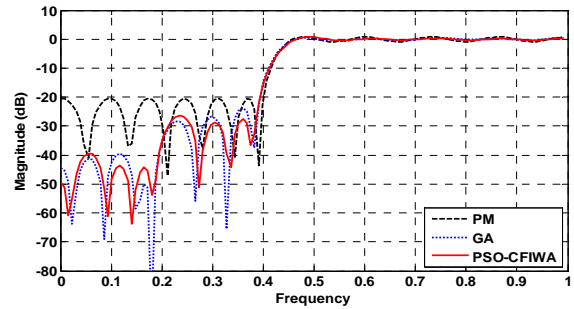


Fig. 1 Gain Plot of 30th Order High Pass FIR Filter

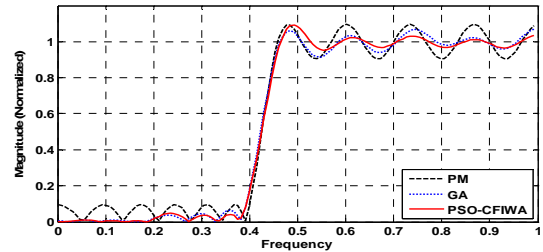


Fig. 2 Magnitude Response of 30th Order High-Pass FIR Filters

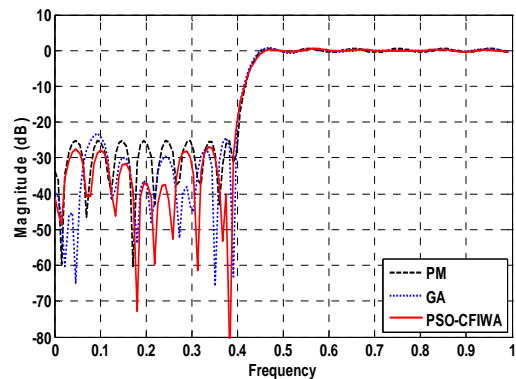


Fig. 3 Gain Plot of 40th Order High Pass FIR Filter

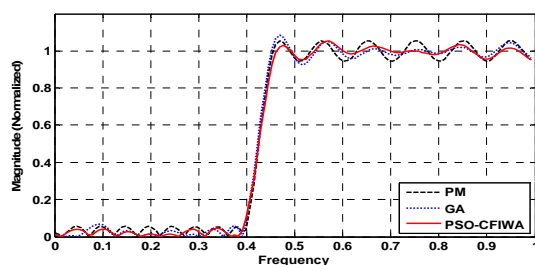


Fig. 4 Magnitude Response of 40th Order High-Pass FIR Filters

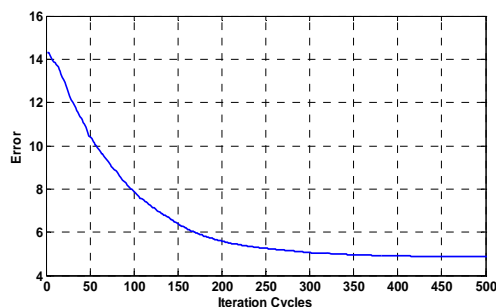


Fig. 5 Convergence Profile for RGA in case of 30th order High Pass FIR Filters

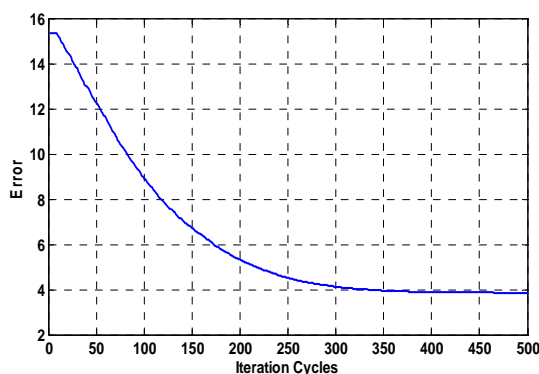


Fig.6 Convergence Profile for PSOCFIWA in case of 30th Order High Pass FIR Filters

VI. CONCLUSIONS

This paper presents a new method for designing linear phase digital high pass FIR filters by using nonlinear stochastic global optimization based on PSO-CFIWA. Filters of orders 30 and 40 have been realized using RGA as well as the proposed PSO algorithm called PSO-CFIWA. Extensive simulation results justify that the proposed algorithm outperforms RGA in the accuracy of the magnitude response of the filter as well as in the convergence speed and is adequate for use in other related design problems.

REFERENCES

- [1] Litwin L. "FIR and IIR digital filters". IEEE Potentials. 0278-6648, 2000, 28–31.
- [2] Parks T W, Burrus C S. "Digital Filter Design". Wiley, New York, 1987.
- [3] Parks T W, McClellan J H. "Chebyshev approximation for non recursive digital filters with linear phase". IEEE Trans. Circuits Theory, CT-19 (1972) 189–194.

- [4] McClellan J H, Parks T W, Rabiner L R. "A computer program for designing optimum FIR linear phase digital filters". IEEE Trans. Audio Electro acoust., AU-21 (1973) 506–526.
- [5] Rabiner L R. "Approximate design relationships for low-pass FIR digital filters". IEEE Trans. Audio Electro acoust., AU-21 (1973) 456–460.
- [6] Herrmann O, Schussler W. "Design of non-recursive digital filters with linear phase". Electron. Lett., 6 (1970), 329–330.
- [7] Mastorakis N E, Gonos I F, Swamy M N S. "Design of Two Dimensional Recursive Filters Using Genetic Algorithms". IEEE Transaction on Circuits and Systems I - Fundamental Theory and Applications, 50 (2003) 634–639.
- [8] Chen S. "IIR Model Identification Using Batch-Recursive Adaptive Simulated Annealing Algorithm". In Proceedings of 6th Annual Chinese Automation and Computer Science Conference, 2000, pp.151–155.
- [9] Luitel B, Venayagamoorthy G K. "Differential Evolution Particle Swarm Optimization for Digital Filter Design". 2008 IEEE Congress on Evolutionary Computation (CEC 2008), PP. 3954-3961, 2008.
- [10] Ababneh J I, Bataineh M H. "Linear phase FIR filter design using particle swarm optimization and genetic algorithms". Digital Signal Processing, 18, 657–668, 2008.
- [11] Kennedy J, Eberhart R. "Particle Swarm Optimization". in Proc. IEEE int. Conf. On Neural Network, 1995.
- [12] Eberhart R, Shi Y. "Comparison between Genetic Algorithms and Particle Swarm Optimization". Proc. 7th Ann. Conf. on Evolutionary Computation, San Diego, 2000.
- [13] Ling S H, Iu H H C, Leung F H F, and Chan K Y. "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.
- [14] Biswal B, Dash P K, and Panigrahi B K. "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.
- [15] Mandal D, Ghoshal S P, and Bhattacharjee A K. "Radiation Pattern Optimization for Concentric Circular Antenna Array With Central Element Feeding Using Crazyness Based Particle Swarm Optimization". International Journal of RF and Microwave Computer-Aided Engineering, 20(5): 577-586, John Wiley & Sons, Inc., Sept. 2010.
- [16] Mandal D, Ghoshal S P, and Bhattacharjee A K. "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.
- [17] Mandal D, Ghoshal S P, and Bhattacharjee A K. "Comparative Optimal Designs of Non-uniformly Excited Concentric Circular Antenna Array Using Evolutionary Optimization Techniques". IEEE Second International Conference on Emerging Trends in Engineering and Technology, ICETET'09 (2009), 619-624.
- [18] Sarangi A, Mahapatra R M, Panigrahi S P. "DEPSO and PSO-QI in digital filter design". Expert Systems with Applications,. Volume 38, Issue 9, September 2011, Pages 10966-10973.