Comparative Analysis of Noise Signal on Application of FIR Filter

Ela Mehra
Electrical Engineering Department
M.I.T.S.
Gwalior, India
mehraela@gmai.com

Dr. S. Wadhwani
Electrical Engineering Department
M.I.T.S.
Gwalior, India

Abstract— Optimization is the process of making something better. Optimization consists in trying variations on an initial concept and using the information gained to improve on the idea. Mathematical operations cannot solve the large scale difficulties efficiently. These causes have contributed to the development of alternative solutions. The techniques used here to design filter are PSO, SFLA and ACO.

Keywords— PSO, ACO, SFLA,Filter

I. INTRODUCTION

The function of a filter is to remove unwanted parts of the signal, such as random noise, or to extract useful parts of the signal, such as the components lying within a certain frequency range. The two purposes served by filters are signal separation and signal restoration. The process of removing interference, noise or other signal from the desired signal is called signal separation whereas signal restoration is the process in which the restoration of the distorted signal is done. Most widely used method is window design method. Some of the window functions used are Hamming Window, Kaiser Window, Bartlet Window etc. The window function truncates the infinite length response into the finite length response. The limitation of this procedure is that the relative values of the amplitude error in the frequency bands are specified by means of weighting function and not by the deviations themselves.

A different evolutionary algorithm such as Genetic Algorithm (GA), Differential Evolution (DE) and Artificial Bee Colony Optimization (ABC) etc. has been used for the design of digital filters. The optimal design of a filter consists in choosing a set of coefficients of the filter to have a frequency response that optimally approximates the desired response. The FIR filter design is a nonlinear, non-differentiable and multimodal optimization problem that requires a suitable objective function to provide an accurate control of the various parameters of frequency spectrum. Therefore, the traditional optimization methods based on gradient do not represent a proper approach to solve this problem. Thus the need for the optimized algorithm arose which design the filter as close as the specified filter.

The following section is divided as follows: section II contains problem formulation. This is followed by section III which consist of three further sub-section containing the brief about algorithms namely PSO, SFLA and ACO. The section IV contains results and conclusion.

II. PROBLEM FORMULATION

The advantage of the FIR filter structure is that it can achieve linear phase frequency response. Hence all design methods are described in the literature deal with filters with this property. A digital FIR filter is characterized by,

$$\mathrm{H(z)} = \sum_{\mathrm{n=0}}^{\mathrm{N}} \mathrm{h(n)\,z^{-n}} \qquad n = 0,1,2,3, \dots\dots N \qquad \dots(1)$$

Where N is the filter order which has N+1 number of filter coefficients, h(n). The coefficients h(n) will determines the low pass filter, high pass filter, etc. The coefficients h(n) are to be determined and N represents the order of the polynomial function. This dissertation presents the even order FIR low pass filter design with coefficients h(n). Hence, (N/2+1) number of h(n) coefficients are optimized, that are finally concatenated to find the required (N+1) number of filter coefficients [2].

Magnitudes of Ideal filter in the pass band and stop band are one and zero. Error function is formed by the errors from the magnitude responses of the ideal filter and the designed filter. In each iteration of the evolutionary algorithm, fitness values of corresponding particle vectors are calculated and are used for updating the particle vectors with new coefficients h(n). The particle vectors obtained after some number of iterations is considered to be the optimal result or best result, obtaining an optimal filter.

Filter parameters which are responsible for the filter design are stop band normalized cut-off frequency ω_s , pass band normalized cut-off frequency ω_p, pass band and stop band ripples δ_p and δ_s respectively.

In this dissertation, three optimization algorithms are used to obtain the magnitude filter response as close as possible to the ideal response and the particle vectors i.e. the coefficients ($h_0,h_1,..,h_N$), are optimized.

The frequency response of the FIR digital filter is calculated as,

$$H(e^{j\omega k}) = \sum_{n=0}^{N} h(n)e^{-j\omega kn} \qquad \ldots\ldots(2)$$

Where

$\omega_k = \frac{2\pi k}{N}$. This is the FIR filter frequency response. The frequency is sampled in [0, π] with N points.

Here error fitness function given by "(4)" has been adopted in order to achieve minimum ripples in pass band and stop band and optimum transition width.

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

Where

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

$$J_1 = \frac{\max}{\omega \le \omega p}(|E(\omega)| - \delta p) + \frac{\max}{\omega \ge \omega s}(|E(\omega)| - \delta s) \quad \ldots (4)$$

Where

δpandδs 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.

Algorithms try to minimize this error fitness J and hence optimize the filter performance. J involves summation of all the absolute errors for the whole frequency band and thus, minimization of J gives much higher stop band attenuation and lesser stop band ripples and transition width.

Passing the noise signal through the optimized filters shows the filter's working. The noise signal which is passed through the low pass filter is given by the equation:

$$x = \sin(2\times\pi\times50t) + 2\times\sin(2\times\pi\times400t) \quad \ldots.(5)$$

Here the component first and second represents the mixture of the low frequency and high frequency component. For further analysis the FFT of the signal is also taken. With the help of the FFT we can determine the frequency domain analysis.

III. OPTIMIZATION TECHNIQUES

Optimization refers to the selection of the best element from some set of available alternatives. In the simplest case, an optimization problem consists of maximizing or minimizing a real function by systematically choosing input values from within an allowed set and computing the value of the function. Evolutionary computing techniques mostly involve metaheuristic optimization algorithms.

A. Particle Swarm Optimization

Kennedy and Eberhart developed PSO. It is an optimization algorithm which is inspired by the social behavior of a flock of birds trying to reach an unknown destination. In PSO, each solution is a 'bird' in the flock and is referred to as a 'particle'. As opposed to GAs, the evolutionary process in the PSO does not create new birds from parent ones [10].

Based on the fitness of the particles they are categorized in the increasing order of their durability. The particle having the best fitness becomes the leader of the group. All other particles are accelerated in the direction of this particle, but also in the direction of their own best solutions that they have discovered previously [19].

All particles also have the opportunity to discover better particles en route, in which case the other particles will change direction and head towards the new 'best' particle. By approaching the current best solution from different directions in search space, the chances are good that these neighboring solutions will be discovered by some of the particles.

Every particle in the algorithm acts as a point in the N-dimensional space. Each particle keeps the information in the solution space for each iteration and the best solution is calculated, that has obtained by that particle is called personal best (pbest). This solution is obtained according to the personal experiences of each particle vector. Another best value that is tracked by the PSO is in the neighborhood of that particle and this value is called best among all pbests.

Throughout the process, each particle i monitors three values: its current position (x_i^k); the bestposition it reached in previous cycles ($pbest_i$); its flying velocity (w^{k+1}).

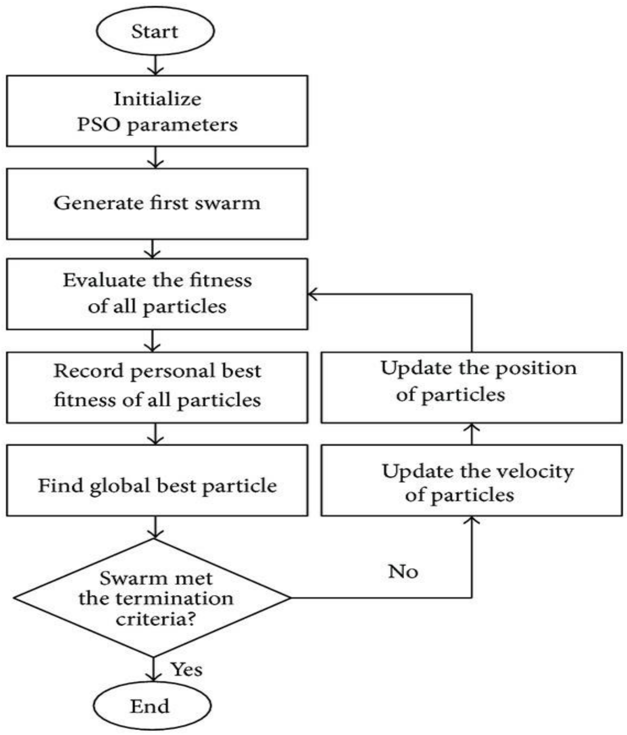

Fig. 1. Flow chart of PSO

These three values are represented as follows [2]:

Current Position $x_i^k = (x_{i1}^k, x_{i2}^k, ...,x_{iN}^k)$ …….(6)

Best previous Position $pbest_i = (pbest_{i1},...pbest_{iN})$ …. (7)

Flying Velocity $w^{k+1} = (w^{k+1}{}_1, w^{k+1}{}_2, ...,w^{k+1}{}_N)$ ….(8)

Mathematically velocity of the particle vectors is given according to the following equation:

$$v_i^{k+1} = w^{k+1} + c_1 \times rand_1 (pbest_i - x_i^k) + c_2 \times rand_2 \times (gbest^k - x_i^k) \quad .... (9)$$

where v_{ik} is the velocity of the i^{th} particle at k_{th} iteration; c_1 and c_2 are the weights of local information and global information commonly named learning factor; $rand_1$ and $rand_2$ are the random numbers between 0 and 1; x_i^k is the current position of the i^{th} particle at k^{th} iteration; $pbest_i$ is the personal best of the i^{th} particle at k^{th} iteration; $gbest^k$ is the group best at k^{th} iteration.

The particle position in the solution space is given by the following equation:

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad .... (10)$$

Where,

x_i^{k+1} = Old Position, x_i^k = Current Position ,

v_i^{k+1} = New Velocity

$$w_{min} \geq w^{k+1} \geq w_{max}$$

The parameter w^{k+1} is the inertia weight and it is used to balance global exploration and local exploitation of the solution space. Itis employed to control the impact of the previous history of velocities on the current velocity. It directs the trade-off between global and local exploration abilities of the flying points. A larger inertia weight w_{max} facilitates global exploration while a smaller inertia weight w_{min} tends to facilitate local exploration to fine-tune the current search area [2].

$$w^{k+1} = w_{max} - (w_{max} - w_{min}) * \frac{(k+1)}{k_{max}} \quad .... (11)$$

Where

k_{max}= Maximum number of the iteration cycles.

B. *Shuffled Frog Leaping Algorithm*

Shuffled Frog Leaping Algorithm has found its ideas from the concepts of Shuffled Complex Evolution (SCE) algorithm and Particle Swarm Optimization (PSO). The ideas of these two algorithms are combined to design an improved meta-heuristic to solve discrete and/or combinatorial problems.

Since the elements of both the algorithms are combined thus this algorithm contain elements of local search and global information exchange. In SFLA the virtual population of frog is portioned into different meme lexes. These frogs act as hosts or carriers of memes. The algorithm performs simultaneously an independent local search in each memeplex. The local search is completed using a particle swarm optimization-like method adapted for discrete problems but emphasizing a local search. To ensure global exploration, the virtual frogs are periodically shuffled and reorganized into new perplexes [15].

An initial population of P frogs is created randomly. For S-dimensional problems (S variables), a frog i is represented as X_i= (x_{i1}, x_{i2},.,x_{iS}). Afterwards, the frogs are sorted in a descending order according to their fitness. Then, the entire population is divided into m meme lexes, each containing n frogs (i.e. P=m×n). In this process, the first frog goes to the first memeplex, the second frog goes to the second memeplex, frog m goes to the mthmemeplex, and frog m+1 goes back to the first memeplex, etc.

Within each memeplex, the frogs with the best and the worst fatnesses are identified as X_b and X_w, respectively. Also, the frog with the global best fitness is identified as X_g[4].

Accordingly, the position of the frog with the worst fitness is adjusted as follows:

Change in frog position (D_i) = rand ()×(X_b – X_w) ……(12)

New position X_w = current position X_w + D_i ….. (13)

Here $D_{max} \geq D_i \geq -D_{max}$

where rand( ) is a random number between 0 and 1; and D_{max} is the maximum allowed change in a frog's position.

If this process produces a better solution, it replaces the worst frog. Otherwise, the calculations in "(12)" and "(13)" are repeated but with respect to the global best frog (i.e. X_g replaces X_b) [16].

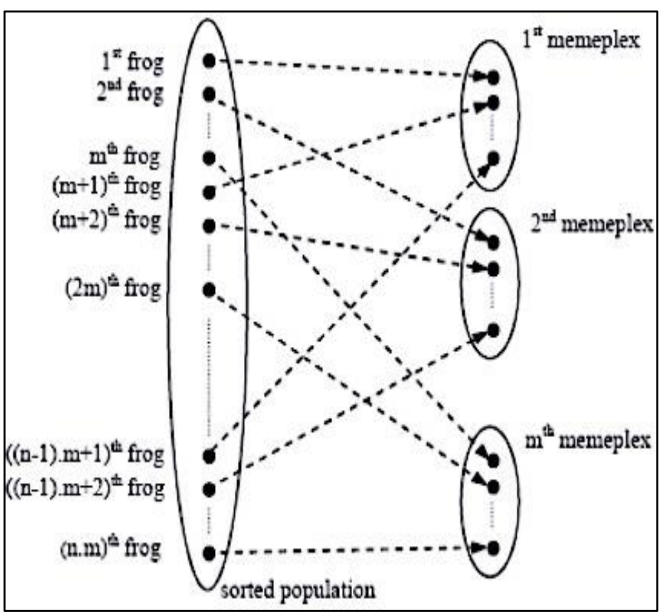

Fig. 2.Memplex Prationing Process

C. Ant Colony Optimization

The main foundation behind the working of ACO is similar to PSO. In this algorithm there is no evolution of genes but there is change in the social behavior. This algorithm was developed by Dorigo. The ability of the ant to find the shortest path between the nest and food source is the principle, which is used to determine the most feasible solution.

The algorithm in the short can be explained as follows. When the ant travel from the nest to food source they leave behind chemical called pheromone. This is used to form the indirect communication. Initially the path is not defined. The ants wander throughout the area. But when a certain ant find the shortest path the pheromone trail is followed by others. As a result of this process the concentration of pheromone increases. The new ants that later starts out from the nest to find food will also choose the shortest path. Over time, this positive feedback (autocatalytic) process prompts all ants to choose the shorter path.

Implementing the ACO for a certain problem requires a representation of S variables for each ant, with each variable i has a set of n_i options with their values l_{ij}, and their associated pheromone concentrations {τ_{ij}}; where i=1, 2,., S, and j=1, 2,., n_i. As such, an ant is consisted of S values that describe the path chosen by the ant.

In the ACO, the process starts by generating m random ants (solutions). An ant k (k=1, 2,., m) represents a solution string, with a selected value for each variable. Each ant is then evaluated according to an objective function. Accordingly, pheromone concentration associated with each possible route (variable value) is changed in a way to reinforce good solutions, as follows:

$$\tau_{ij} = \rho\tau_{ij}(t-1) + \Delta\tau_{ij} \quad t = 1,2, \dots \dots, T \qquad \dots \dots . . (14)$$

where T is the number of iterations (generation cycles), τ_{ij} is the revised concentration of pheromone associated with option l_{ij} at iteration t, τ_{ij} (t-1) is the concentration of pheromone at the previous iteration (t-1), $\Delta\tau_{ij}$ is change in pheromone concentration and r is pheromone evaporation rate (0–1) [4]. T is the number of iterations (generation cycles), τ_{ij} is the revised concentration of pheromone associated with option l_{ij} at iteration t, τ_{ij} (t-1) is the concentration of pheromone at the previous iteration (t-1), $\Delta\tau_{ij}$ is change in pheromone concentration and r is pheromone evaporation rate (0–1).

The reason for allowing pheromone evaporation is to avoid too strong influence of the old pheromone to avoid premature solution stagnation. In "(14)", the change in pheromone concentration $\Delta\tau_{ij}$ is calculated as:

$$\Delta\tau_{ij} = \sum_{k=1}^{m} \begin{cases} R/fitness_k, & if\ option\ l_{ij} is\ chosen\ by\ ant\ k \\ 0, & otherwise \end{cases} \quad \ldots (15)$$

Where

R= constant called pheromone reward factor

$fitness_k$= objective function which is calculated for an ant k. It is to be noted that as the amount of pheromone gets higher the solution improves.

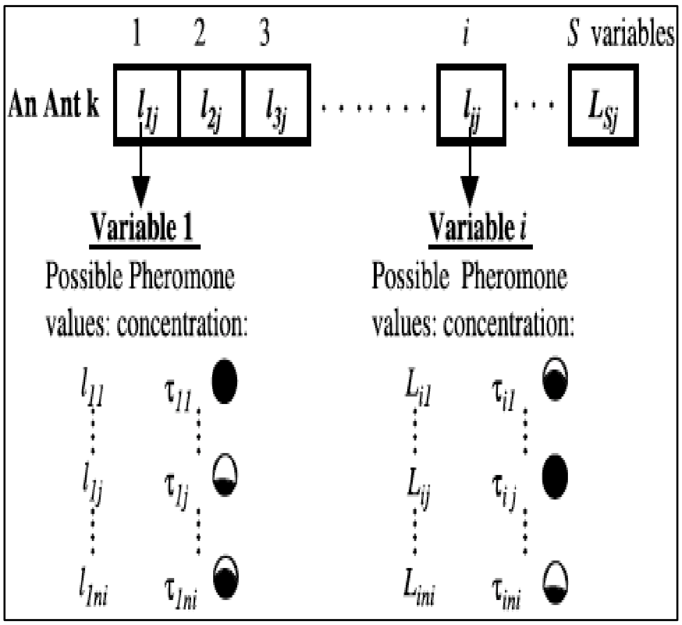

Fig. 3. Ant Representation

Therefore, for minimization problems, "(15)" shows the pheromone change as proportional to the inverse of the fitness. In maximization problems, on the other hand, the fitness value itself can be directly used [13].

After the updating of pheromone is done the ant path is changed. The change in the path of the ant is done according to the pheromone concentration. The value of each variable for ant k at iteration t is done according to following probability:

$$P_{ij}(k,t) = \frac{[\tau_{ij}(t)]^{\alpha} \times [\eta_{ij}]^{\beta}}{\sum_{l_{ij}}[\tau_{ij}(t)]^{\alpha} \times [\eta_{ij}]^{\beta}} \quad \ldots\ldots (16)$$

The values of each constant are defined as follows:-

$P_{ij}(k,t)$ = probability that option l_{ij} is chosen by ant k for variable i at iteration t.

$\tau_{ij}(t)$ = pheromone concentration associated with option l_{ij} at iteration t.

η_{ij}= heuristic factor for preferring among available options. It also indicates about the goodness of the available ant k to selected option l_{ij}

α and β are exponent parameters that control the relative importance of pheromone concentration versus the heuristic factor. The values of α and β should be greater than zero [4].

IV. RESULTS AND CONCLUSIONS

The value of the sampling frequency adopted is *fs*= 1 Hz and the number of sampling points is taken as 205. The Tables 1 list the parameters of the algorithms used in designing of filter.

Table I and Table II shows the parameters selection and the optimized parameter obtained when the algorithms are run. These optimized parameters obtained from algorithms namely SFLA, PSO and ACO are used to design the filter.

From Table III to Table V the statistical values of magnitude (dB), normalized magnitude, normalized stop-band ripple, normalized pass-band ripple and stop-band ripple (dB) are shown.

Parameters	**PSO**	**ACO**	**SFLA**
No. Of Decision Variables	20	20	20
Maximum Iteration	1500	1500	1500
Population	50	50	50
Sample Size	-	40	-
C_1	2.05	-	-
C_2	2.05	-	-
Intensification Factor	-	0.5	-
Deviation Distance Ratio	-	1	-
Var Max.	+1	+1	+1
Var Min.	-1	-1	-1

Table 1: Parameter Selection.

h(N)	SFLA	PSO	ACO
h(1)=h(21)	0.03488326	0.036834771	0.034883483
h(2)=h(20)	0.03214452	0.155003058	0.0321408168
h(3)=h(19)	-0.0388624	0.305793775	-0.0388472666
h(4)=h(18)	-0.1176467	0.353130100	-0.1176499349
h(5)=h(17)	-0.0807840	0.220117593	-0.0808226437
h(6)=h(16)	0.10593390	-0.003311502	0.105906919
h(7)=h(15)	0.32014733	-0.140718066	0.3201399756
h(8)=h(14)	0.39679004	-0.111690133	0.3968076672
h(9)=h(13)	0.29657640	-0.000273690	0.2965737458
h(10)=h(12)	0.13133144	0.061987534	0.1313425105
h(11)	0.02660291	0.043831736	0.0266184512

Table 2: List of the Coefficients of the Filter of Order of 20 Designed by SFLA, PSO and SFLA.

The magnitude in decibels is represented by *fig.4*. The normalized magnitude response of the stop band is shown in *fig. 5*. *Fig. 6* and *fig. 7* presents the normalized magnitude response of the stop band and pass band respectively. *Fig.* 8 represents stop band attenuation in dB of the various algorithms.

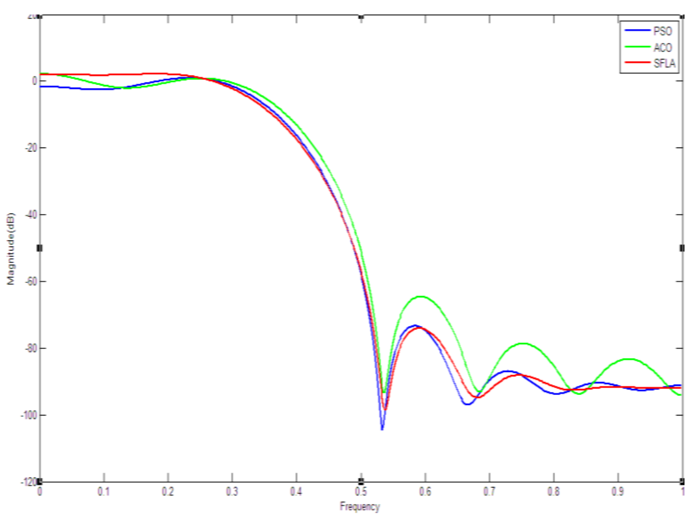

Fig. 4. Magnitude (Db) Plot gor the Low Pass FIR Filter of Order 20

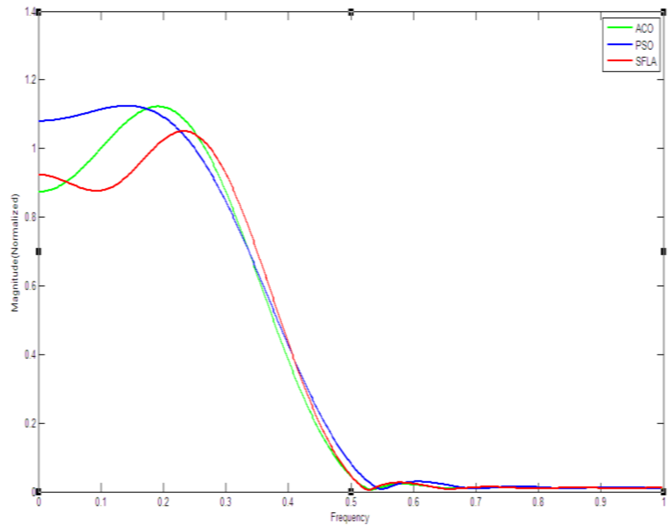

Fig. 5. Magnitude (normalized) Plot for the Low Pass FIR Filter of Order 20

In the normalized curves the curve shown by *fig. 5* the ripples are less in both pass band and stop band. However the minimum ripples are in PSO.

The Table III presents a summary of the simulation results obtained by PSO, ACO and SFLA. For the order 20 filters, PSO obtained a maximum attenuation of 1.1128548 (normalized) and mean attenuation of 0.40778197 (normalized).

The ACO filters obtained a maximum attenuation of 1.10829860 (Normalized) and mean attenuation of 0.43526938 (normalized) whereas the values for SFLA are 1.04965605 (normalized) and 0.38361822 (normalized) respectively.

Algorithm	Min	Max	Mean	Median	Mode	Std.
PSO	0.009	1.112	0.407	0.059	0.009	0.447
ACO	0.005	1.108	0.432	0.047	0.005	0.483
SFLA	0.004	1.049	0.383	0.048	0.004	0.428

Table 3: Statistical Parameters of Stop Band Attenuation For Different Algorithms For the FIR LP Filter (Normalized)

In *fig.* 7 we can notice that the pass band ripple of the LP FIR filter of order 20 designed by PSO is smaller than ACO, and the SFLA obtained the best result regarding this performance parameter.

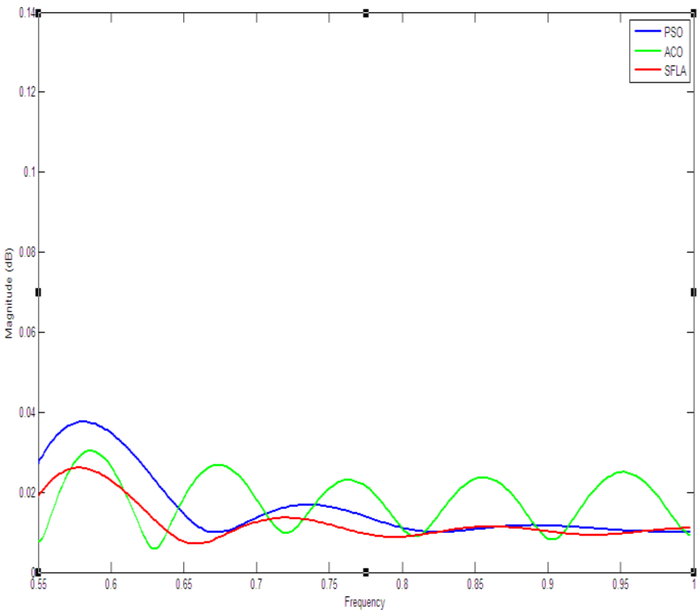

Fig. 6. Magnitude (Normalized) Plot for the Stop Band Low Pass FIR Filter of Order 20

For a good filter the ripple must be minimum. The ripples in the filter designed by the PSO algorithm are minimum. Thus we can say that for minimum magnitude of stop band ripple will be obtained if designed by PSO algorithm.

Algorithm	Min.	Max.	Mean	Median	Mode	Std.
PSO	0.003	1.102	0.434	0.087	0.003	0.473
ACO	0.004	1.026	0.409	0.103	0.004	0.425
SFLA	0.004	1.047	0.383	0.048	0.004	0.427

Table 4: Statistical Parameters of Pass Band Attenuation for Different Algorithms for the FIR LP Filter (Normalized)

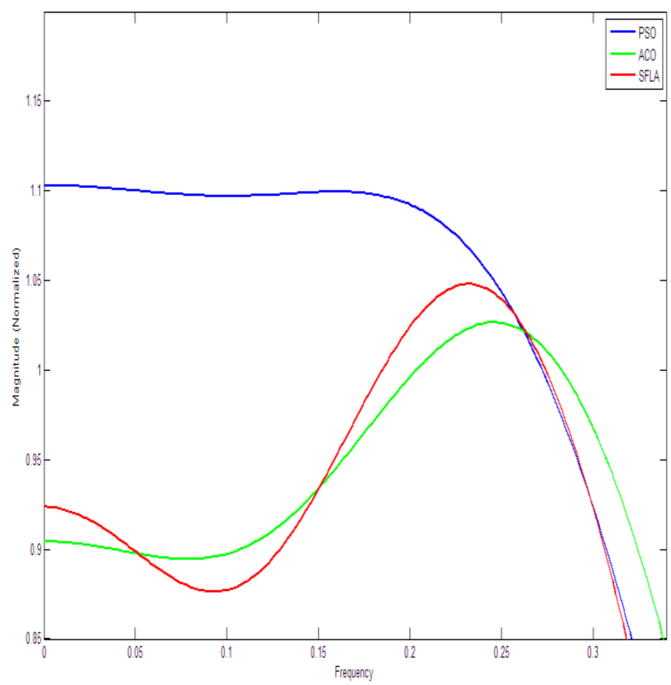

Fig. 7. Magnitude (normalized) Plot for the Pass Band Low Pass FIR Filter Of Order 20

The Table V presents the maximum, mean, variance and standard deviation for dB attenuation in the filters stop band. The magnitude response in dB of the stop band is illustrated in *Fig. 8.*

The pass band of a filter is the area where the frequency of the given specifications must not have any disturbances. In other words the ripples are not desirable. Thus the minimum ripples are obtained in the design of filter through PSO algorithm. The ripples obtained in the algorithms ACO and SFLA are close to each other but ther is a wide gap between them and of PSO.

Algorithm	Min.	Max.	Mean	Median	Mode	Std.
PSO	-97.2	0.651	-40.4	-44.4	-97.2	35.99
ACO	-126.1	2.203	-48.3	-65.5	-126.1	41.88
SFLA	-126.7	2.239	-40.3	-42.0	-126.7	37.91

Table 5: Statistical Parameters of Stop Band Attenuation for Different Algorithms for the FIR LP Filter (decibles)

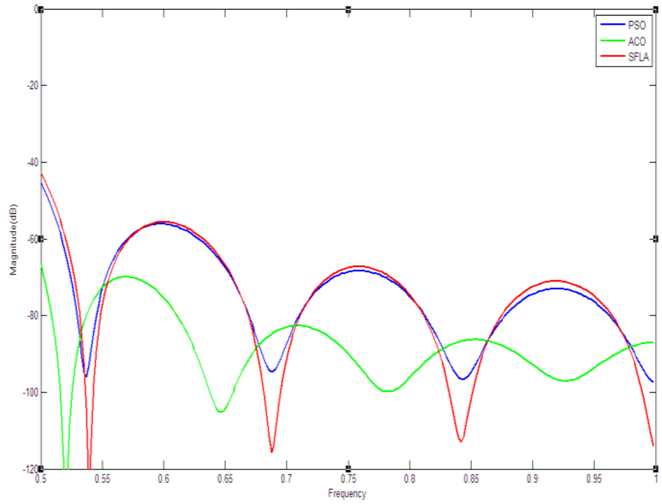

Fig. 8. Magnitude (Decibles) Plot For the Pass Band Low Pass FIR Filter of Order 20

The following figure shows the filtering of the noise signal when passed through the filters designed by the three algorithms. The signal when passed through the filter designed by PSO shows that the magnitude of the filtered data is in the range of ± 1 whereas that designed by SFLA crosses the range.

All noises lying outside the desired range are removed and the signals the resultant signal is smooth without any noise. To further visualize the filtering action the FFT of the signal was taken. The component lying in the high frequency range is removed whereas that in low frequency range is shown by the low pass filters. *Fig. 9* shows the FFT.

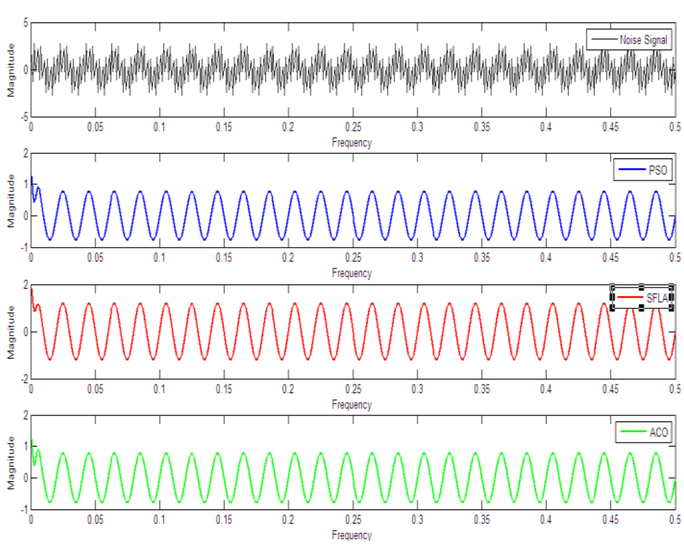

Fig. 9. Filtering Action Shown By Different Low Pass Filter

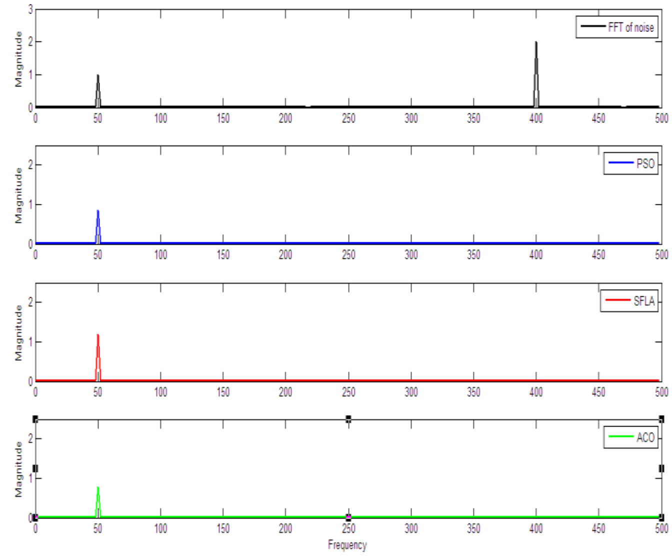

Fig. 10. FFT Action Shown By Different Low Pass Filter

In this paper the traditional Particle Swarm Optimization, Ant Colony Optimization and Shuffled Frog Leaping Algorithm were adopted to adjust the coefficients of a low pass FIR filter. The simulation results demonstrated that PSO, ACO and SFLA are also efficient approaches to solve the problem of filter design. Regarding the best LP FIR filters designed by the met heuristics evaluated in this work, the PSO algorithm presented better results for pass band ripple. The best FIR filter designed by PSO obtained the smallest pass band ripple, but SFLA and ACO presented a close value for this same performance parameter. Therefore, the simulation results presented in this work demonstrated that the best LP FIR filters of order 20 were designed by PSO.

The further evaluation of the optimized filter designed is done by the FFT analysis. The noise signal is passed through the optimized filters and the frequency of desired region is obtained. The magnitude of the frequency wave is best obtained by the PSO filter closely followed by the ACO whereas the filter through the SFLA algorithm shows a wide range of magnitude. In the FFT analysis the component in the high frequency region is removed whereas that in low frequency region is retained.

Once the PSO also have presented satisfactory results, all the three met heuristic algorithms analyzed can be considered efficient optimizers to solve the problem of digital filters design.

In future works, another met heuristics and deterministic methods will be considered in order to realize a more extensive comparison between methods that can be used to project FIR and IIR digital filters. We also expect to evaluate the use of different objective functions for this problem.

REFERENCES

[1]. J. Kennedy, "The Particle Swarm: Social Adaptation of Knowledge", Proceeding of the IEEE International Conference on Evolutionary Computation. IEEE Service Center, 1997. pp. 303-308.

[2]. S. Mondal, S.P. Ghoshal, R. Kar, D. Mandal, "Novel Particle Swarm Optimization for Low Pass Filter Design", IEEE Symposium on Humanities, Science and Engineering Research (SHUSER), vol. 8, 2012, pp. 413-418.

[3]. R.C. Eberhart, Y. Shi, "Comparison Between Genetic Algorithms and Particle Swarm Optimization", Proceeding of the 7th Annual Conference on Evolutionary Programming, vol. 1447, 1998. pp. 611-618.

[4]. E. Eldeltagi, T. Hegzay, D. Grierson, "Comparison Among Five Evoutionary-Based Optimization Algorithms", Advance Engineering Informatics, vol 19, 2005, pp. 43-53.

[5]. Y. Shi, R. Eberhart, "A Modified Particle Swarm Optimizer", Proceeding of the IEEE International Conference on Evolutionary Computation, IEEE Press,1998. pp. 69-73.

[6]. M. Eusuff, K. Lansey, F. Pasha, "Shuffled Frog Leaping Algorithm : A Memetic Meta-Huristic for Discrete Optimization", Engineering Optimization, vol. 38, 2006,pp. 129-154.

[7]. A.G.V. Severino, L.L.S. Linhares, F.M.U. de Araujo, "Optimal Design of Digital Low Pass Finite Impulse Response Filter using Particle Swarm Optimization and Bat Algorithm", Proceeding of IEEE 12th International Conference on Informatics in Control, Automation and Robotics (ICINCO), vol. 1, 2015, pp. 207-214.

[8]. F.van den Bergh, "An Analysis of Particle Swarm Optimizers", PhD(Computer thesis), University of Pretoria, Pretoria, 2002.

[9]. 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.

[10]. J. Kennedy, R.Eberhart, "Particle Swarm Optimization", Proceeding of the IEEE International Conference on Neural Networks, vol. 4, 1995, pp. 1942-1948.

[11]. V. Maniezzo, L.M. Gambardella, F. de Luigi, "Ant Colony Optimization", In: New Optimization Techniques in Engineering. Studies in Fuzziness and soft Computing, vol. 141, 2004, pp. 101 – 121.

[12]. T. Stutzle, M. Lopez, P. Pellegrini, M. Maur, M. Oca, M. Birattari, M. Dorigo, "Parameter Adaptation in Ant Colony Optimization", Autonomous Search, Springer,2011, pp. 191-215.

[13]. M. Birattari, G.D. Caro, M. Dorigo, "Towards the Formal Foundation of Ant Programming", Ant Algorithms – Proceedings of ANTS 2002 – Third International Workshop, vol. 2463, 2002, pp. 188-201.

[14]. K.Y. Wong, "Parameter Tuning for Ant Colony Optimization: A Review", International Conference on Computer and Communication Engineering, IEEE, 2008, pp. 542-545.

[15]. B. Amiri, M. Fathian, A. Maroosi, "Application of Shuffled Frog-Leaping Algorithm on Clustering", The International Journal of Advanced Manufacturing Technology, vol. 45, 2009, pp. 199-209.

[16]. G. G. Samuel, C. C. A. Rajan, "A Modified Shuffled Frog Leaping Algorithm for Long-Term Geneartion Maintenance Scheduling" , Proceeding of the Third International Conference on Soft Computing for Problem Solving, vol. 1, 2013, pp. 929-1043.

[17]. M. Dorigo, C. Blum, "Ant Colony Optimization Theory: A Survey", Proceeding of the Third International Conference on Soft Computing for Problem Solving, vol. 258, 2014, pp. 11-24.

[18]. M. Dorigo, G. D. Caro, "Ant Colony Optimization: A New Meta-Heuristic", Proceeding of the IEEE on Evolutionary Computation, vol. 2, 1999, pp. 1470-1477.

[19]. I. Sharma, B. Kuldeep, A. Kumar, V. K. Singh, "Performance of Swarm Based Optimization Techniques for Designing Digital FIR Filter: A Comparative Study", International Journal of Engineering Science and Technology, vol. 19, 2016, pp. 1564-1572.

[20]. D. G. Cabrero, D. N. Ranasinge, "Fine Tuning the Ant Colony System Algorithm Through Particle Swarm Optimization", International Journal of Hybrid Intelligence System, vol. 9, 2012, pp. 171-183.

[21]. L. R. Rabiner, "Techniques for Designing Finite Duration Impulse Response Digital Filters", IEEE Transactions on Communication Technology, vol. 19, 1971, pp. 188-195.

[22]. N. Kumar, "Optimal Design of FIR and IIR Filters Using Some Evolutionary Algorithms", Masters Thesis, Indian Institute of Technology, Delhi, India, 2013.