Linear Phase Low Pass FIR Filter Design using Genetic Particle Swarm Optimization with dynamically varying neighborhood Technique

Avishek Ghosh, Arnab Ghosh, Arkabandhu Chowdhury, Amit Konar
Department of Electronics and Tele-communication Engineering, Jadavpur University
Kolkata-700032, India
Email: avishek.ghosh38@gmail.com, arnabju90@gmail.com, arjia_2005@yahoo.com, konaramit@yahoo.co.in

Eunjin Kim[1], Atulya K. Nagar[2]
1. Department of Computer Science, John D. Odegard School of Aerospace Sciences, University of North Dakota, Grand Forks, ND 58202-9015, USA
2. Department of Math and Computer Science, Liverpool Hope University, Liverpool, UK
Email: ejkim@cs.und.edu, nagara@hope.ac.uk

***Abstract*—The paper presents an elegant approach for designing linear phase low pass digital FIR filter using swarm and evolutionary algorithms. Classical gradient based approaches are not efficient enough for accurate design and thus evolutionary approach is considered to be a better choice. In this paper a hybrid of Genetic Algorithm and Particle Swarm Optimization algorithm with varying neighbourhood topology, namely Genetic Lbest Particle Swarm Optimization with Dynamically Varying Neighbourhood (GLPSO DVN) is used to find the filter coefficients. In this work two objective functions (error metrics) are minimized. The first one is based on stop and pass band ripple and the second one studies the mean square error between the ideal and actual designed filter. The hybrid algorithm is found to produce fitter candidate solution than the classical Lbest PSO. The results are compared with the results obtained by solving the same problem using Lbest PSO (LPSO). It is also observed that GLPSO DVN gives better results than LPSO and as well LPSO DVN.**

Keywords- Llbest PSO; Genetic Algorithm; crossover; mutation; Digital filters; Low pass filters; Finite impulse response filter

I. INTRODUCTION

A filter is a frequency selective network which allows a certain band (or bands) of frequency to pass while attenuate the others. Filter may be of different kinds, namely low pass filter, high pass filter, band pass filter, band reject filter, notch filter etc. Filter circuits may be analog or digital. The design of analog filters makes use of electronic components like resistor, capacitor, transistor etc. and the input to such signals is continuous time signal. These type of filters are helpful for video/audio signal amplification, noise reduction etc. In contrast, digital filters use digital processors which perform certain mathematical calculations on the sampled values of the input signal. Though analog filters are cheaper, digital filters are vastly used for practical applications as digital filters have a greater degree of precision and stability. Also digital filters are more flexible because their characteristics can be altered simply by changing the program. The availability of memory to store digital signals aids to more sophisticated signal processing algorithms to be used to design the filter. A great variety of filters, such as very low frequency filters can be easily designed by digital processors where it requires bulky components to realize it in analog domain. Digital filters are appropriate in time sharing environments and the processors can be easily fabricated in small packages with low power consumption and high packing density.

There are two major classes of digital filters namely, finite impulse response (FIR) filters and infinite impulse response (IIR) filters depending on the length of the impulse response [1]. For FIR filter the length of impulse response is finite and for IIR filters the length is infinite. Designing FIR filter is attractive because of the ease of design and the higher stability of the filter. Since there is no required feedback in the FIR filter, all the poles are located at the origin and within the unit circle which infer the fact that these filters are inherently stable. FIR filters have only zeros (no poles) apart from the origin, hence also known as all-zero filters. FIR filters also known as feed forward or non-recursive or transversal filters [2]. A linear phase filter has its coefficients symmetrical about its center. In the design of frequency-selective filters, the desired filter characteristics are specified in the frequency domain in terms of the desired magnitude and phase response of the filter. In the filter design process, the job is to determine the coefficients of the linear phase FIR filter.

There exist certain methodologies to design a FIR filter. Windowing method is the most popular one. There exists various kinds of window functions (for example, Butterworth, Chebyshev), depending on the requirements of ripples on the passband and stopband, stopband attenuation, transition width etc. The most frequently used method to design exact linear phase weighted Chebyshev FIR digital filter is the one based on the Remez-exchange algorithm proposed by Parks and McClellan, also known as the PM algorithm [3]. These various windows convert the infinite length impulse response of ideal filter into a finite window to design an actual response. But this method does not allow sufficient control of the frequency response in the various frequency bands [4].

The modern approach to FIR filter design uses evolutionary algorithm. The coefficients of the filter to be designed are found by minimizing the error function in such a way that the designed filter closely resembles the ideal or desired one. Since population based stochastic and meta-heuristic search methods have proven to be effective in multidimensional nonlinear environment, all of the constraints of filter design can be taken care of by the use of these algorithms [4]. Previously, computational optimization technique such as Genetic Algorithm (GA) [5], Particle Swarm Optimization (PSO) [2], [6], Differential Evolution (DE) [4] is applied to design linear phase FIR filter. In this work, a new neighbourhood toplogy called dynamically varying neighbourhood (DVN) for Lbest PSO is described. Along with a new algorithm hybridizing GA with Lbest PSO, namely GLPSO is proposed. It is worth observing that GLPSO DVN leads to more accurate solutions, thus minimizing error to higher extent and also the rate of convergence is higher than that of Lbest PSO.

II. FIR FILTER DESIGN

The input signal *x(t)* is sampled and the samples are denoted as *x(0), x(1),* ....etc. Also the output is found at sample intervals. The output of a filter is generally written as,

$$y(n) = -\sum_{k=1}^{Q} a_k y(n-k) + \sum_{k=o}^{P} b_k x(n-k) \tag{1}$$

For a FIR filter, $a_k = 0$ for all *k*. Depending upon the type of operation, the order of the filter is carefully chosen. The order of the filter equals to the number of the past input samples. Therefore the order of the filter will be one less than the number of coefficients. For example,

$$ZeroOrder, y(n) = a_0 x(n)$$
$$FirstOrder, y(n) = a_0 x(n) + a_1 x(n-1)$$
$$SecondOrder, y(n) = a_0 x(n) + a_1 x(n-1) + a_2 x(n-2)$$

Using z transforms, the response of the N-1 th order filter will be

$$Y(z) = a_0 X(z) + a_1 z^{-1} X(z) + a_2 z^{-2} X(z) + ... + a_{N-1} z^{-(N-1)} X(z) \tag{2}$$

The corresponding transfer function will be,

$$T(z) = \frac{Y(z)}{X(z)} = a_0 + a_1 z^{-1} + a_2 z^{-2} + ... + a_{N-1} z^{-(N-1)} \tag{3}$$

So all the poles will be at origin eliminating any chance of instability.

The transfer function for an IIR filter would be,

$$T(z) = \frac{Y(z)}{X(z)} = \frac{a_0 + a_1 z^{-1} + ... + a_{N-1} z^{-(N-1)}}{1 + b_1 z^{-1} + ... + b_{M-1} z^{-(M-1)}} \tag{4}$$

To design a filter, various filter parameters come into picture. They are passband and stopband cut-off frequencies *(w_p and w_s)*, passband and stopband ripple *(δ_p and δ_s)* attenuation factor in stopband and transition width. The parameters are totally determined by the filter coefficients [7], [8]. For a design problem, a certain set of desired parameters will be given. The job is to find the filter coefficients in such a way so that the deviation of these parameters from the desired value will be minimum. In present case the job is to design a Low pass filter with a given set of parameters. Figure 1 shows a typical example of how a designed filter deviates in characteristics from an ideal filter. For the filter shown in figure 1, the passband and the stopband frequencies are 0.3 and 0.4 (normalized with respect to *π*) and the transition width *(w_s-w_p)* is 0.1. The passband and stopband ripples are shown [4].

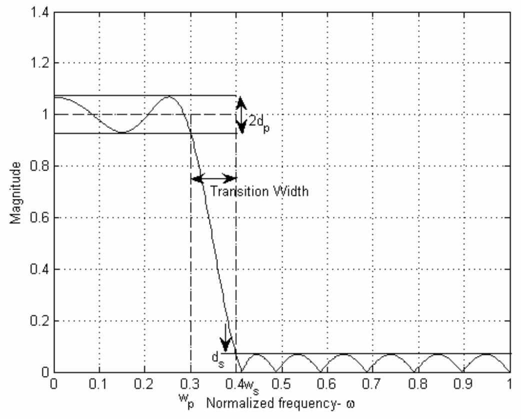

Figure 1. Ideal and Actual filter Magnitude response showing passband and stopband ripples and transition band in actual filter.

III. ALGORITHM DESCRIPTION

A. Lbest version of Classical PSO

Particle swarm optimizer (PSO), introduced by Kennedy and Eberhart in 1995 [12], [13], [14] is one of the most useful and important optimizing technique. Its very idea has been derived from the flocking nature of birds or fish. Usually optimization problems can be expressed as *min f(x)*, $x=[x^1,x^2,...,x^D]$, where *D* is the number of the parameters to be optimized. In PSO, each particle with its corresponding position and velocity in the *D* dimensional problem space represents a potential solution. Every particle flies through the space by exchanging useful information of fitness values of others and thereby finding the global best position (with the least fitness value). It also takes into account its personal best position so far in the search process and combining these two results the particle updates its velocity according to a pre-defined inertia weight on its personal best as well as global best position. Thus every particle has a tendency to fly towards better search area over the course of search process. The velocity V_i^d and position X_i^d updates of d^{th} dimension of the i^{th} particle are presented below:

$$\begin{aligned} V_i^d &= w \times V_i^d + c_1 \times rand1_i^d \times (pbest_i^d - X_i^d) \\ &+ c_2 \times rand2_i^d \times (gbest^d - X_i^d) \end{aligned} \tag{5}$$

$$X_i^d = X_i^d + V_i^d \tag{6}$$

Where c_1 and c_2 are the acceleration constants, $rand1_i^d$ and $rand2_i^d$ are two uniformly distributed random numbers in the range [0,1]. $x_i=[x_i^1, x_i^2, \ldots, x_i^D]$ is the position of the ith particle; $pbest_i=[pbest_i^1, pbest_i^2, \ldots, pbest_i^D]$ is the best previous position yielding the best fitness value $pbest_t^i$ for the i^th^ particle; $gbest=[gbest^1, gbest^2, \ldots, gbest^D]$ is the best position discovered by the whole population; $V_i=[V_i^1, V_i^2, \ldots, V_i^D]$ represents the rate of the position change (velocity) for particle *i*. *w* is the inertia weight used to balance between the global and local search abilities.

Classical PSO contains two main variants: global PSO and local PSO. In the local version of PSO, each particle's velocity is adjusted according to its personal best and the best performance achieved so far within its neighborhood instead of learning from the personal best and the best position achieved so far by the whole population in the global version. The velocity updating equation becomes:

$$V_i^d = w \times V_i^d + c_1 \times rand1_i^d \times (pbest_i^d - X_i^d) + c_2 \times rand2_i^d \times (lbest_i^d - X_i^d) \quad (7)$$

Where $lbest_i = [lbest_i^1, lbest_i^2, \ldots, lbest_i^D]$ is the best position achieved within its neighbourhood.

Focusing on improving the local version of PSO, different neighbourhood structures are proposed and discussed [10][11][14][15][16]. Except these local PSO variants, some variants that use multi-swarm [17], sub-population [18] can also be included in the local version PSOs if we treat the sub-groups as special neighbourhood structures. In the existing local versions of PSO with different neighbourhood structures and the multi-swarm PSOs, the swarms are predefined or dynamically adjusted according to the distance. Hence, the freedom of the swarms is limited.

B. Dynamically varying Neighbourhood Topology

In the proposed PSO technique we construct a ring of unity radius. Local best position of every particle is found and velocity is updated according to the lbest as well as personal best position of each particle. Equal amount of acceleration coefficient *(c1 = c2)* is given to either of lbest and pbest. Relatively smaller inertia coefficient (w) is given to the current position of the particle. This procedure is carried for a certain number of iterations called *'max_iter'*. After that, a generation is completed and we reconstruct the ring of unity radius by a random fashion so that a new neighbourhood topology is developed. Then again we repeat the same procedure.

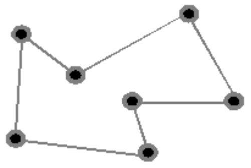
Figure 2. Ring topology of unity radius

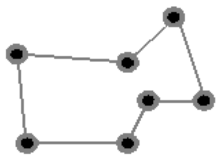
Figure 3. State of particles after few iterations

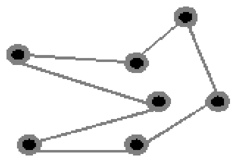
Figure 4. Reconstruct ring topology *(dynamically varying neighbour)*

In a classical PSO since the small sized swarms are searching using their own best historical information, they are easy to converge to a local optimum because of PSO's convergence property. In order to avoid, this special topology will be very useful.

C. Genetic Algorithm(GA)

Genetic Algorithms (GAs) are search algorithms based on the mechanics of the natural selection process (biological evolution) [9]. The most basic concept is that optimization is based on evolution, and the "Survival of the fittest" concept. GA have the ability to create an initial population of feasible solutions, and then recombine them in a way to guide their search to only the most promising areas of the state space. Each feasible solution is encoded as a chromosome (string) also called a genotype, and each chromosome is given a measure of fitness via a fitness (evaluation or objective) function. The generations of the new solutions are developed by genetic recombination operators:

- *Biased Reproduction*: selecting the fittest to reproduce
- *Crossover*: combining parent chromosomes to produce children chromosomes
- *Mutation*: altering some genes in a chromosome.

a) Crossover:

Actually it is a segment exchanging method. Two particles are chosen randomly. Then two cut points from each of them are selected and segments between the cut-points are exchanged. Thus, we get two new particles, i.e. we have two sons corresponding to two parents. Let us take two particles with corresponding positions within brackets.

x(parent1)=[2.2 3.4 / 6.8 5.4 2.0 5.8 6.4 7.5 / 8.9 6.7],

x(parent2)=[4.3 5.4 / 7.3 9.3 3.2 2.1 4.8 2.5 / 8.4 8.4].

Now let us assume the two cut points are 3rd and 8th. So sons are:

x(son1)=[2.2 3.4 7.3 9.3 3.2 2.1 4.8 2.5 8.9 6.7],

x(son2)=[4.3 5.4 6.8 5.4 2.0 5.8 6.4 7.5 8.4 8.4].

b) Mutation:

After completing crossover we have a new population pool and every particle in the pool will now go through the mutation by a probability. Actually two types of mutations are introduced, namely Exchange Mutation and Simple Inversion. Let us take a particle for example,

x(particle)=[1.2 4.4 1.4 5.3 6.4 7.1 8.3 4.0 1.5 6.0].

Now for Exchange Mutation, we have to choose randomly two exchange points, say 4th and 9th. Then the two data corresponding to those points are exchanged. So, the mutated particle will be

x(mutated)=[1.2 4.4 1.4 1.5 6.4 7.1 8.3 4.0 5.3 6.0].

For Inversion Mutation, we also have to choose two points and simply invert the segment between the points. If the points are 4th and 9th, the mutated particle will be

x(son)=[1.2 4.4 1.4 1.5 4.0 8.3 7.1 6.4 5.3 6.0].

c) Selection (Biased reproduction):

During each successive generation, a proportion of the existing population is selected to breed a new generation. Selection is based on the 'survival of the fittest' technique. After crossover population size becomes double. We evaluate the fitness of each chromosome and select fit chromosomes to maintain the population size.

D. GLPSO with Dynamically varying Neighbourhood:

Here we use two types of flag namely flag1 and flag2. Flag1 denotes the beginning of new generation i.e. the reconstruction of ring topology is done. Flag1 increases with generation and when it reaches a certain value it reset to zero and Genetic operations are carried over that generation along with PSO. Flag2 denotes in which iterations of that generation genetic operations are done. So here we can identify two types of generation, normal generation and special generation. In normal generation particles go through lbest PSO. In the end of every generation reconstruction of ring topology is done. In special generation particles go through genetic operations along with lbest PSO.

Pseudo Code:

```
Initialize particles
generation=0;
flag1 = 0;
Do
    if flag1<max_gen
        Call lbest PSO with iteration equal to max_iter
        generation = generation+1;
        flag1 = flag1+1;
    else
        Call GLPSO with iteration equal to max_iter
        generation = generation+1;
        flag1 = 0;
    endif
Reconstruct ring topology;
While maximum generation criteria (maximum Fes) not reached
```

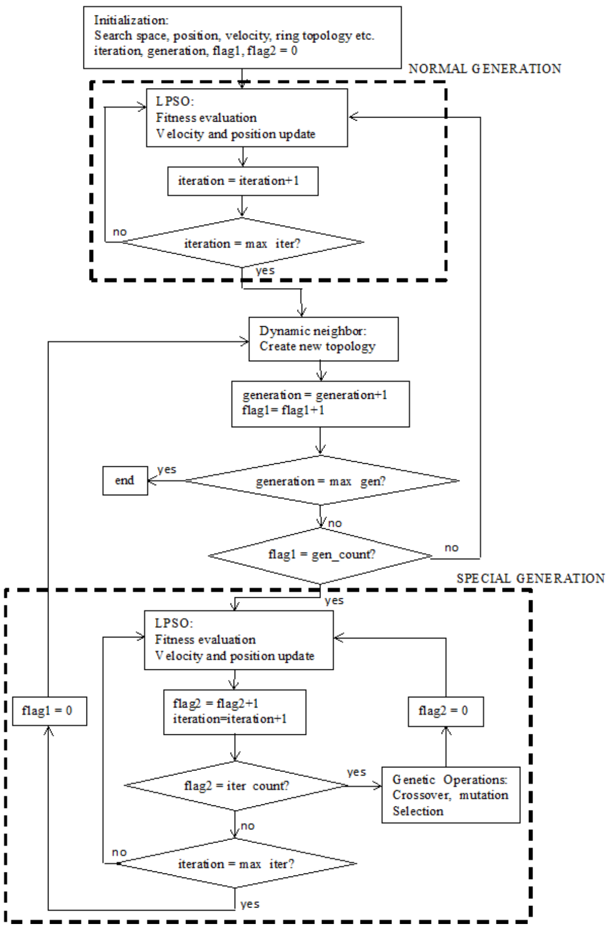

Figure 5.   Flowchart showing algorithm flow

IV. GLPSO BASED FIR FILTER DESIGN

The transfer function of the FIR filter is given as (equation 3)

$$T(z) = a_0 + a_1 z^{-1} + a_2 z^{-2} + ... + a_{N-1} z^{-(N-1)} \tag{8}$$

For design purpose, it is assumed that the filter is a 19th order filter and so N = 20. The coefficients $\{a_0, a_1 ... a_{n-1}\}$ are found by iterating Genetic Algorithm Particle Swarm Optimization algorithm. N denotes the dimension and at every iteration the position of updated particles denotes a set of coefficients. If the filter has linear phase characteristics, $a_0 = a_{N-}$

$_1$, $a_1=a_{N-2}$ etc. and the dimension is reduced to *N/2* effectively. The particles are distributed in *N/2* dimensional hyperspace and the fitness of the particles may be found in computing either of the two objective functions described below. The fitness is used to improve the search at every iteration, and the results obtained after a certain number of iterations are considered to be the final result. Depending upon fitness, two objective functions are formulated in the present work.

Objective function I

Different kinds of objective functions are found in different literature to give a proper metric of error. The frequency response of the FIR filter is given by,

$$H(e^{jw}) = \sum_{n=0}^{N-1} h(n)e^{-jwn} \tag{9}$$

An error function can be approximated using Parks–McClellan algorithm [2], [4],

$$E(w) = G(w)[\ H(e^{jw}) - H_i(e^{jw})] \tag{10}$$

where *G(w)* denotes the weighting function used to provide different weights for the approximate errors in different frequency bands and H denotes the frequency response of the ideal filter given by,

$$\begin{aligned} H_i(e^{jw}) &= 1, 0 \le w \le w_c \\ &= 0, otherwise \end{aligned} \tag{11}$$

where w_c is the cut-off frequency of the ideal filter.

The major drawback of Parks-McClellan algorithm is that the ratio of δ_p/δ_s is fixed. To improve the flexibility in the objective function to be minimized, and that the desired level of δ_p and δ_s may be specified, the error function given in (name) has been considered as fitness function in many literatures [2], [4], [19], [20]. The error fitness to be minimized using the GPSO algorithm, is defined as:

$$J_1 = \max_{w<w_p}(|E(w)| - \delta_p) + \max_{w>w_s}(|E(w)| - \delta_s) \tag{12}$$

δ_p and δ_s denote pass band and stop band ripple, w_p and w_s denote pass band and stop band normalized cut-off frequencies. GAPSO tries to minimize J_1.

Objective function II

The second fitness function considers the mean square error between the frequency response of the ideal and the designed filter. The error is taken on sample by sample basis in the frequency domain. An ideal filter has unity magnitude in pass band and null magnitude in stop band. So the error function is the squared value of the magnitude difference of the two filters averaged upon the total number of samples taken into consideration. Mathematically,

$$J_2 = \frac{1}{L}\sum_{k=1}^{L}[Desired\ \ (k) - Ideal\ \ (k)]^2 \tag{13}$$

Where *Ideal(k)* is 1 up to cut-off frequency w_c and 0 otherwise and *Desired(k)* denotes samples of the magnitude response of the designed filter. L is the total number of samples taken into consideration. The objective is to minimize J_2 so that the designed filter closely resembles the ideal filter.

V. Results

The filters are designed to optimize the coefficients which give the best frequency response. This is determined by the ripples on the passband and the stopband. The desired ripple in this problem on the passband δ_p is 0.1 and that on the stopband δ_s is 0.01. In each case, passband and stopband cut off frequencies are 0.25 and 0.3 respectively and the cut-off frequency of the ideal filter is 0.275 (all normalized with respect to π. Filters with 20 coefficients (19th order filters) are designed. The design parameters are written on Table I.

TABLE I. Parameters Used in Algorithms

Parameters	**Values**
Acceleration Coefficient *(c_1, c_2)*	1.49445
Inertia weight *(w)*	0.729
Maximum FEs	5000
population size	25
max_iter	20
max_gen	3
mutation probability	.05
crossover rate	80%

The coefficients obtained from the filter design have been enlisted in Table II.

TABLE II. Coefficient Obtained Using LPSO and GLPSO DVN

Co-efficients	**LPSO**		**GLPSO DVN**	
	Case1	***Case2***	***Case1***	***Case2***
1 (20)	0.0740	0.0325	0.0560	0.0312
2 (19)	0.0337	0.0329	0.0523	0.0328
3 (18)	0.0017	0.0090	0.0174	0.0085
4 (17)	-0.0445	-0.0291	-0.0427	-0.0302
5 (16)	-0.0115	-0.0588	-0.0544	-0.0579
6 (15)	-0.0366	-0.0470	-0.0656	-0.0483
7 (14)	0.0002	0.0107	0.0283	0.0103
8 (13)	0.0210	0.1059	0.1033	0.1056
9 (12)	0.2727	0.2044	0.1801	0.2042
10 (11)	0.2608	0.2672	0.2789	0.2667

Ripple of passband and stopband obtained in both Case1 and Case2, along with the average value and standard deviation are enlisted in Table III. It is worth noting that although the value of ripple is slightly higher in case of LPSO, the deviation is zero in case of GLPSO DVN.

TABLE III. PASSBAND AND STOPBAND RIPPLES

Ripple		**LPSO**		**GLPSO DVN**	
		Case1	***Case2***	***Case1***	***Case2***
Passband δ_p	*avg*	0.1524	0.1572	0.1624	0.1579
	min	0.1441	0.1572	0.1574	0.1579
	max	0.1604	0.1572	0.1723	0.1579
	std	0.0032	0.0000	0.0012	0.0000
Stopband δ_s	*avg*	0.3120	0.2226	0.1613	0.2246
	min	0.2915	0.2226	0.1597	0.2246
	max	0.3321	0.2226	0.1703	0.2246
	std	0.0015	0.0000	0.0009	0.0000

Error, magnitude and gain obtained using LPSO, LPSO DVN and GLPSO DVN are plotted in fig 6-11. In case of error we can see the gradual improvement of proposed algorithm. Imposing new neighbourhood topology (LPSO_DVN) we have better convergence and accuracy than the classical LPSO. But hybridization (GLPSO_DVN) leads to best result in both cases by means of convergence as well as accuracy. From magnitude and gain plots, it can be observed that GLPSO_DVN infers superior results than classical LPSO and LPSO_DVN.

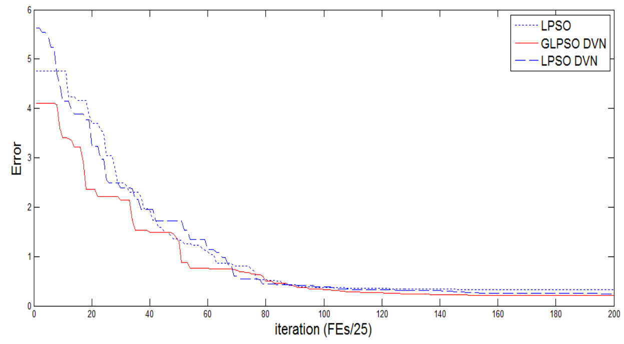

Figure 6. Error graph for Case 1

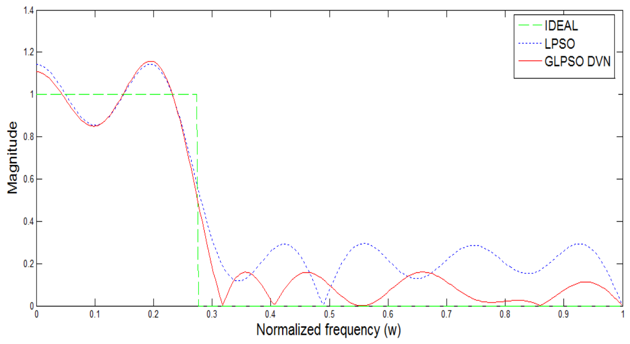

Figure 7. Magnitude response of filter for Case 1

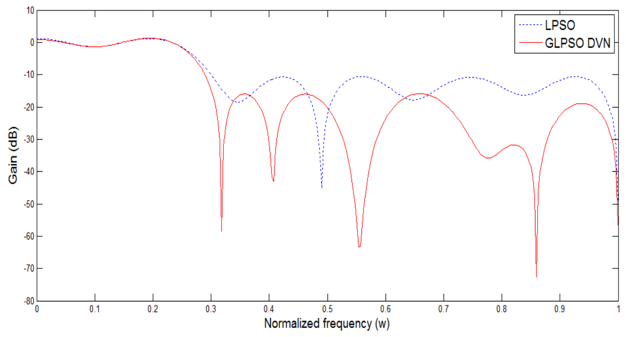

Figure 8. Gain plot of filter for Case 1

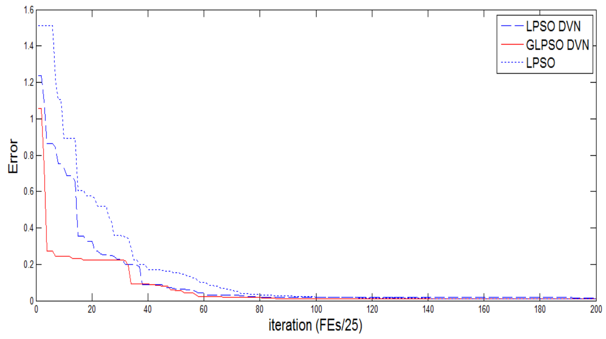

Figure 9. Error graph for Case 2

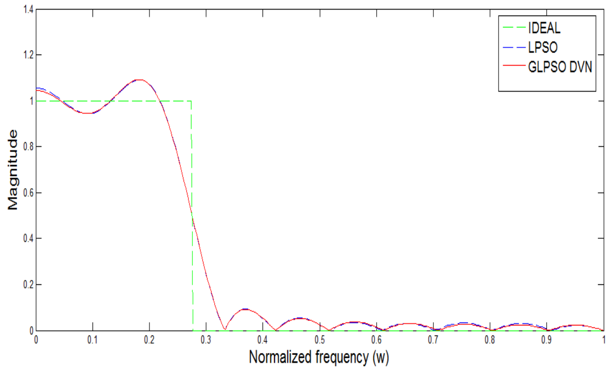

Figure 10. Magnitude response of filter for Case 2

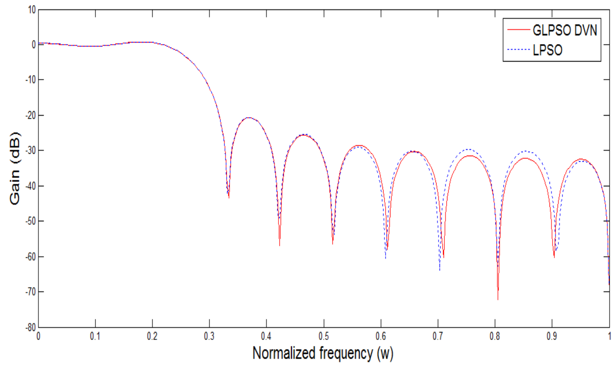

Figure 11. Gain plot of filter for Case 2

In fig 12-13, the comparison between the two objective functions is made with the ideal response and it is found that the second objective function (case 2) leads to fitter solution as verified from fig 12-13.

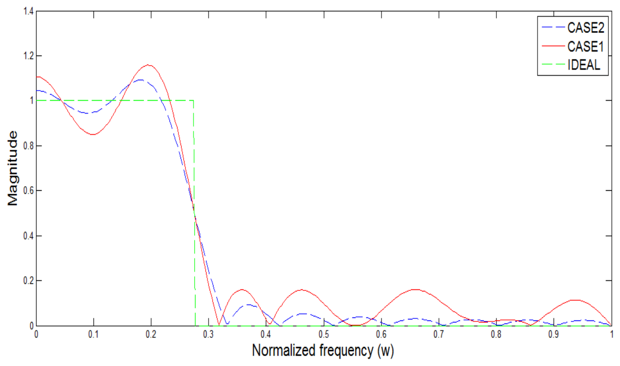

Figure 12. Comparison between Case1 and Case2 in magnitude plot

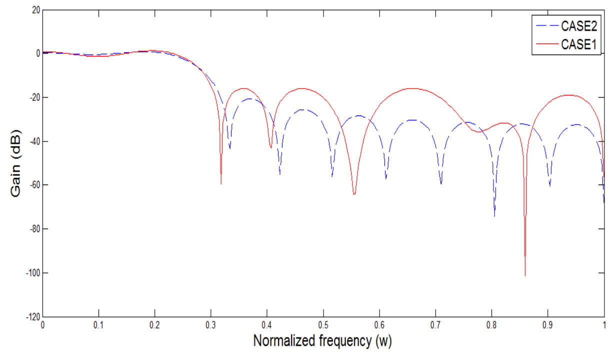

Figure 13. Comparison between Case1 and Case2 in gain plot

VI. Conclusion

In the present work it is shown that hybrid optimization algorithms lead to better solution with less error than the individual algorithm. Also the convergence of hybrid algorithms is fast with better accuracy. The fitness is significantly improved as the dynamically varying neighbourhood topology prevents the particles to be trapped in local minima. Thus, Local best Particle Swarm optimization (Lbest PSO) with Dynamically Varying Neighbourhood (DVN) topology hybridized with Genetic Algorithm (GLPSO) gives better accuracy, lower error rates and faster convergence than Classical Lbest PSO or the LPSO DVN algorithm. With the two objective functions, the design using the second objective function leads to fewer ripples and close to ideal response than the first objective function. It is to be noted that the transition width of the designed filter using the second objective function is more than the design using the first one. Although low transition width is a design criterion, lower value of pass band and stop band ripple is a stronger design aspect and so the design using second objective is considered superior than the first one. It can be concluded that swarm and evolutionary techniques in filter design can be achieved in dynamic environment which is useful for adaptive filtering. Hybridization of other types of algorithms to improve the performance will be our future research goal. Also development of dynamic algorithms to track changing minima will be our future endeavor.

References

[1] L. Litwin, “FIR and IIR digital filters,” IEEE Potential, 0278-6648, 2000, 28–31.

[2] Sangeeta Mondal, Vasundhara,Rajib Kar, Durbadal Mandal, S. P.Ghoshal, Linear Phase High Pass FIR Filter Design using Improved Particle Swarm Optimization, World Academy of Science, Engineering and Technology ,60 ,2011,pp-1620-1627

[3] T.W. Parks, J.H. McClellan, “Chebyshev approximation fornon-recursive digital filters with linear phase,” IEEE Trans. Circuits Theory CT-19 (1972), pp. 189–194.

[4] 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.

[5] 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.

[6] Rajib Kar, Durbadal Mandal, Dibbendu Roy, Sakti Prasad Ghoshal, FIR Filter Design using Particle Swarm Optimization with Constriction Factor and Inertia Weight Approach, Proc. of Int. Conf. on Advances in Computer Engineering 2011,pp-55-59.

[7] Karaboga, N., “Digital Filter Design Using Differential Evolution Algorithm,” EURASIP Journal of Applied Signal Processing, 2005:8, pp. 1269-1276.

[8] Krusienski, D. J., Jenkins, W. K., “Particle Swarm Optimization for Adaptive IIR Filter Structures,” in Congress on Evolutionary Computation , June 2004, vol. 1, pp. 965-970.

[9] Goldberg DE (1989) Genetic algorithm in search, optimization and machine learning, Machine Learning, Addison-Wesley, New York

[10] J. Kennedy, "Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance". Proc. of IEEE Congress on Evolutionary Computation (CEC 1999), Piscataway, NJ. pp. 1931-1938, 1999.

[11] P. Larranaga, C.M.H. Kuijpers, R.H. Murga, I. Inza and S. Dizdarevic, "Genetic Algorithms for the Travelling Salesman Problem: A Review of Representations and Operators". Artificial Intelligence Review 13: 129–170, 1999, Kluwer Academic Publishers, Netherlands.

[12] J. Kennedy, "Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance". Proc. of IEEE Congress on Evolutionary Computation (CEC 1999), Piscataway, NJ. pp. 1931-1938, 1999.

[13] J. Kennedy and R. Mendes, "Population structure and particle swarm performance ". Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2002),Honolulu, Hawaii USA. 2002).

[14] R. C. Eberhart, and J. Kennedy, "A new optimizer using particle swarm theory", Proc. of the Sixth Int. Symposium on Micromachine and Human Science, Nagoya, Japan. pp. 39-43, 1995.

[15] J. Kennedy, and R. C. Eberhart, "Particle swarm optimization ". Proceedings of IEEE International Conference on Neural Networks, Piscataway, NJ. pp. 1942-1948, 1995.

[16] P. N. Suganthan, "Particle swarm optimizer with neighborhood operator," Proc. of the IEEE Congress on Evolutionary Computation (CEC 1999), Piscataway, NJ, pp. 1958-1962, 1999.

[17] X. Hu and R. C. Eberhart, "Multiobjective optimization using dynamic neighborhood particle swarm optimization," Proc. IEEE Congress on Evolutionary Computation, Hawaii, pp. 1677-1681, 2002.

[18] R. Mendes, J. Kennedy, and J. Neves, "The fully informed particle swarm: simpler, maybe better”. IEEE Transactions on Evolutionary Computation, 8(3):204 - 210, June 2004

[19] 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.

[20] D.Mandal, S.P.Ghoshal and A.K.Bhattacharjee, “Swarm Intelligence based Optimal Design of Concentric Circular Antenna Array,” Journal of Electrical Engineering, vol.10, no.3,pp 30-39,2010.