Design of Low-Pass FIR Digital Filter using Soft Computational Techniques: A Comparison

Sukhdeep Kaur Sohi
Giani Zail Singh College of Engineering & Technology, Bathinda

Balraj Singh
Giani Zail Singh College of Engineering & Technology, Bathinda

Darshan Singh
Government Polytechnic College, Bathinda

***Abstract*— This paper compares the performance of Hybrid Differential Evolution (HDE) algorithm with that of Differential Evolution (DE) and Particle Swarm Optimization (PSO) algorithm. Parameter tuning of these three algorithms has been walked around to optimize the results. Results show that HDE has an upper hand when compared with DE and PSO algorithm.**

Keywords*— *Exploratory move, mutation strategy, optimization

I. INTRODUCTION

A device which permits only a particular frequency range to pass while jamming the path of other frequencies is given the name 'filter'. Filters are one of the most imperative and dominant tool of DSP. Window method and frequency sampling method are the traditional methods for designing digital filters. But they failed to provide flexibility [1, 2].

Optimization algorithms are used to minimize or maximize a certain work. It is a simple operation for discovering the attractive outputs from a list of possible solutions. These are classified in to direct search, gradient search and nature inspired methods. Nature inspired methods have become a well known name in the field due to their close resemblance with real biological systems. Ant colony search, particle swarm optimization, predator prey optimization, genetic algorithm and differential evolution are the few drops of nature inspired techniques [3, 4].

But as every coin has two sides, all the above mentioned method offered some disadvantages like slow convergence speed, problem of local minima and dependency on initial parameters. So to overcome these disadvantages, hybrid evolution came in to existence. Hybrid means combining the better of two algorithms to sort out an optimization problem [5, 6].

This paper has been divided into V sections. Section II focuses on the FIR filter design problem where as Section III describe the algorithm of DE, HDE and PSO algorithm. Results are enclosed in Section IV. Finally Section V concludes the final work.

II. PROBLEM FORMULATION

FIR filters are digital filters with finite impulse response. Unlike IIR filters they do not have feedback and hence given the name recursive filters. Structure of FIR filters is simply composed of adders, multipliers and adders. FIR filters are described by a difference equation as described in Eq. (1) as follows:

$$y(n)=\sum_{k=0}^{N-1} h_k x(n-k) \tag{1}$$

where y(n) is the output produced by an input x(n). h_k is the impulse response and N is the order of filter [2,7]. Desired magnitude response for the ideal filter is given as described in Eq. (2)

$$H_d(\omega_i)=\begin{cases}1 & for \quad \omega_i \in pass-band \\ 0 & for \quad \omega_i \in stop-band\end{cases} \tag{2}$$

where H_d describe the desired magnitude response. Realizing an ideal filter is not possible, there is always a scope of error [8]. Magnitude errors are described as follows in Eq. (3) and Eq. (4) as below:

$e_1(x)$ – absolute error L_1-norm of magnitude response

$e_2(x)$ - squared error L_2 norm of magnitude response

$$e_1(x)=\sum_{i=0}^{K} \left| H_d(\omega_i)-|H(\omega_i,x)| \right| \tag{3}$$

$$e_2(x)=\sum_{i=0}^{K} \left(\left| H_d(\omega_i)-|H(\omega_i,x)| \right| \right)^2 \tag{4}$$

where $H(\omega_i,x)$ is the magnitude response of designed filter. The ripple magnitudes errors $\delta_1(x)$ and $\delta_2(x)$ of pass-band and stop-band respectively are to be minimized. Ripple magnitudes errors are described in Eq. (5) and Eq. (6) as follows:

$$\delta_1(x)=\max\{|H(\omega_i,x)|\}-\min\{|H(\omega_i,x)|\} \tag{5}$$

and $$\delta_2(x)=\max\{|H(\omega_i,x)|\} \tag{6}$$

Combining all objectives and stability constraints

Minimize $K_1(x)=e_1(x)$ (7)

Minimize $K_2(x)=e_2(x)$ (8)

Minimize $K_3(x)=e_3(x)$ (9)

Minimize $K_4(x)=e_4(x)$ (10)

$$\text{Minimize } K(x) = \sum_{q=1}^{4} w_q K_q(x) \quad (11)$$

where w_q indicates the different weights and K(x) stands for objective function whose value is to be minimized [6].

III. HYBRID DIFFERENTIAL EVOLUTION ALGORITHM

In 1995, Storn and price introduced a new optimization technique named "Differential Evolution Algorithm". It came as a blessing for non-linear and non differentiable function. Requirement of few control parameters, a simple technique with high convergence speed added stars to the fame of DE algorithm. [4, 9, 10]

DE algorithm is excellent in investigating the search space and tracing the region of global optima but sometimes it is time-consuming at fine tuning the solution. Some alteration on the basic DE algorithm can enhance its performance. Such as hybridizing DE algorithm with different optimizers such as additional local searchers and other population based met heuristics.. This paper DE algorithm has been hybridized with Hooke Jeeves exploratory move and its performance has been compared with traditional DE algorithm. HDE algorithm comprises of same operators as that of DE named mutation, crossover, and selection and with an addition of exploratory move [11, 12]. Let a function with H real parameters with population size P is to be optimized. The parameter is described by Eq. (12):

$$x_{i,G} = \left[x_{1,i,G}, x_{2,i,G}, \ldots\ldots\ldots, x_{H,i,G}\right] \quad (12)$$

A. Mutation

Addition of the weighted difference of two vectors to the third randomly chosen vector to generate a mutant or donor vector is named as mutation [9, 10]. It is mathematically described in Eq. (13):

$$v_{i,G+1} = x_{r_1,G} + f_m\left(x_{r_2,G} - x_{r_3,G}\right) \quad (13)$$

where f_m is the mutation factor having a value in the range of [0,1]. $x_{r_1,G}, x_{r_2,G}$ and $x_{r_3,G}$ are randomly chosen vectors which are different from the $x_{ij,g}$ known as target vector. DE makes use of different variants which are classified using the following notations named DE $\alpha|\beta|\delta$. α describe the method for selecting a parent chromosome that form the base foundation for the mutant vector. β points to the number of difference vectors used to work up the parent chromosome. δ recognizes the recombination mechanism for generating offspring population. DE variants could be addressed using a particular notation. For e.g. DE/best/1/bin, in this particular nomenclature DE stands for Differential Evolution algorithm, best means the vectors selected for mutation procedure is the best vector of current generation, 1 is the number of solution pairs selected and bin indicates the binomial crossover [9, 13, 14, 15]. In this paper five mutation strategies of DE has been explored which are listed below:

MS-1: DE/rand/1: $v_{ij}^t = x_{r_1}^t + f_m\left(x_{r_2}^t - x_{r_3}^t\right)$ (14)

MS-2: DE/best/1: $v_{ij}^t = x_{B_j}^t + f_m\left(x_{r_1}^t - x_{r_2,j}^t\right)$ (15)

MS-3: DE/current to best /1:

$$v_{ij}^t = x_{ij}^t + f_B\left(x_{Bj}^t - x_{ij}^t\right) + f_m\left(x_{r_1,j}^t - x_{r_2,j}^t\right) \quad (16)$$

MS-4: DE/best/2:

$$v_{ij}^t = x_{Bj}^t + f_m\left(x_{r_1 j}^t + x_{r_2 j}^t - x_{r_3 j}^t - x_{r_4 j}^t\right) \quad (17)$$

MS-5:DE/rand/2:

$$v_{ij}^t = x_{r_5 j}^t + f_m(x_{r_1 j}^t + x_{r_2 j}^t - x_{r_3 j}^t - x_{r_4 j}^t) \quad (18)$$

B. Crossover

Generation of trail vector by mixing the elements of donor vector with that of target vector is referred as the process of crossover. There are two types of crossover named binomial crossover and exponential crossover. In this paper binomial crossover has been discussed [9, 15]. Binomial crossover can be mathematically illustrated as shown in Eq. (19):

$$u_{ij,G+1} = \begin{cases} v_{ij,G} & if \quad (randb((j) \le CR) \quad or \quad j = rnbr(i) \\ x_{ij,G} & if \quad (randb(j) \le CR) \quad and \quad j \ne rnbr(i) \end{cases} \quad (19)$$

where i=1,2,……..,P and j= 1,2,……..,H

randb(j) is the jth evaluation of a uniform random generator with outcome ϵ [0,1]. CR is the crossover constant ϵ [0,1] which has to be determined by the user, rnbr(i) is a randomly chosen index ϵ 1,2,……H which ensures that $u_{ij,G+1}$ gets at least one parameter from $v_{ij,G+1}$. rnbr(i) ensures that $v_{ij,G+1} \ne x_{ij,G}$

C. Selection

Finally a choice has to be made between the trail vector and the target vector. The one which yields the lower value of objective function is kept, while the other is throw away [9, 10]. Selection can be mathematically expressed by following Eq.

$$x_{ij,G+1} = \begin{cases} u_{ij} & if \ f\left(u_{ij,G+1}\right) < f\left(x_{ij,G}\right) \\ x_{ij} & otherwise \end{cases} \quad (20)$$

D. Exploratory Move

This step is included to enhance the performance of traditional DE algorithm. In the exploratory move the current solution denoting filter coefficients is agitated in positive and negative directions and the best point is witnessed. This move is a success if it can yield the lower value of objective function otherwise it is a failure. It can be mathematically illustrated as follows in Eq. (21):

$x_i^n = x_i^o \pm \Delta_i u_i^j$ for (i=1,2,…,P: j=1,2,…,H); (21)

where $u_i^j = \begin{cases} 1 & if \quad i = j \\ 0 & if \quad i \ne j \end{cases}$

The objective function can be computed as described in Eq. (22) as follows:

$$x_i^n = \begin{cases} x_i^o + \Delta_i u_i & ; K\left(x_i^o + \Delta_i u_i\right) < K\left(x_i^o\right) \\ x_i^o - \Delta_i u_i & ; K\left(x_i^o - \Delta_i u_i\right) < K\left(x_i^o\right) \\ x_i^o & ; otherwise \end{cases} \quad (22)$$

IV. PSO ALGORITHM

In 1995 J. Kennedy and R. Ebherhart introduced a new nature inspired technique named, "Particle Swarm Optimization (PSO)".It is motivated from the behavior of animals such as bird flocking [16]. . PSO algorithm is initiated with a chosen population of random particles. Swarm is the name given to the population of PSO and each individual of that swarm is referred as particle. Each particle is allocated a random velocity initially. Mathematically the position and velocity matrix can be described by Eq. (23) and Eq. (24) as follows:

$$p_i = [p_{i_1}, p_{i_2}, \ldots\ldots \ldots, p_{i_Q}] \qquad (23)$$

$$v_i = [v_{i_i}, v_{i_2}, \ldots\ldots \ldots, v_{i_Q}] \qquad (24)$$

where Q is the dimension and i=[1,2,………L] is number of particles. p_{bset} is known as the local best and the best value in the group is termed as g_{best}. Each particle in the search space alter its position by calculating the distance between the current position and local best and distance between the group best and current position. Amendments in the velocity of particle according to the following Eq. (25) as follow:

$$v_{id}^{t+1} = wv_{id}^{t} + C_1 * rand() * (pbest_{id} - p_i^t) + C_2 * Rand() * (gbest_{id} - p_{id}^t)$$

$$p_{id}^{t+1} = p_{id}^{t} + v_{id}^{t} \qquad (25)$$

where C_1 and C_2 are acceleration constant and rand() and Rand() are the random numbers whose value lie in the range of 0 to 1.W is weighing function illustrated by Eq. (26)

$$w = w_{\max} - (w_{\max} - w_{\min})\left(\frac{It}{MAXIt}\right) \qquad (26)$$

where $w_{\max}$ and $w_{\min}$ are the maximum and minimum value of weighing function. It stands for number of iterations and MAXIt point to the maximum number of iterations chosen for an optimization problem [16, 17, 18].

PSO Algorithm:-

1. A random position and velocity is allocated to each particle of the population.
2. Objective Function of each particle of the population is calculated and is designated as K.
3. If K > p_{best} than p_{best} is set as S.
4. Then g_{best} is computed by choosing the minimum value of p_{best}.
5. Velocity of particles is repaired by using Eq. (25).
6. Check maximum number of runs accomplished or not, if yes continue to step 7 otherwise revisit step 2.
7. Check maximum number of runs carried out or not. If yes terminate the program, otherwise return to step 1.

V. RESULTS

A. DE and HDE algorithm results

A low-pass FIR digital filter has been designed using HDE and traditional DE algorithm. Five mutation strategies described in Eq. (14) to Eq. (18) has been implemented on filter order 22. The performance has been depicted in Fig. 1 as follows:

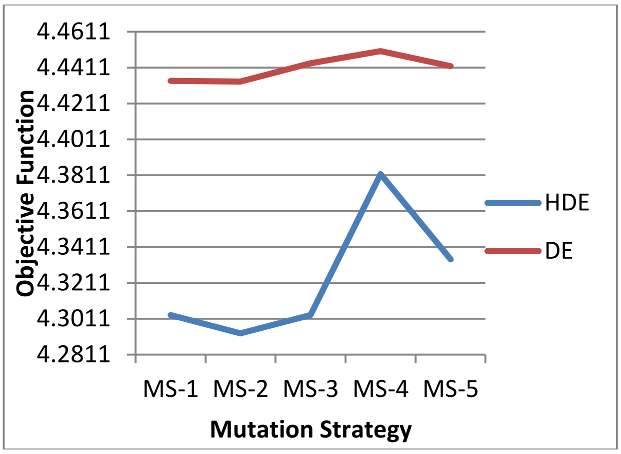

Fig.1: Objective Function obtained for different mutation strategies for HDE and DE algorithm on Filter Order 22

It is evident from the above Fig.1 that minimum value of objective function has been obtained as 4.43333 and 4.292996 for DE and HDE algorithm respectively with MS-2. Now parameters value of both DE and HDE algorithm has been varied to enhance the value of objective function. Keeping the MS-2 and filter order 22 population has been varied from 80 to 160 in steps of 20 for both HDE and DE algorithm. Results have been drawn in Fig.2 and Fig.3 as follows:

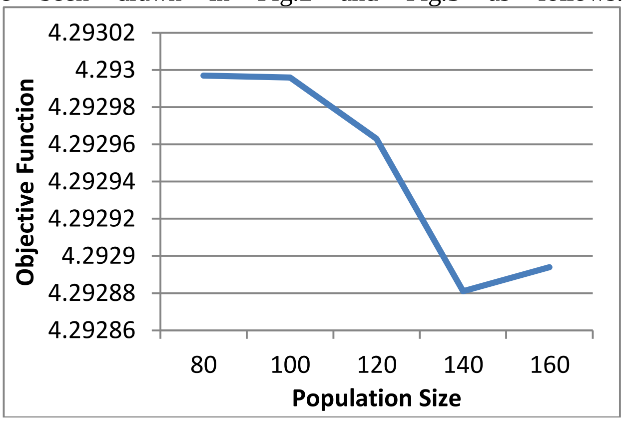

Fig.2: Objective Function versus Population Size with MS-2 for Filter Order 22 for HDE algorithm

It is evident from the above Fig.2 that minimum value of objective has been obtained with population sixe 140.

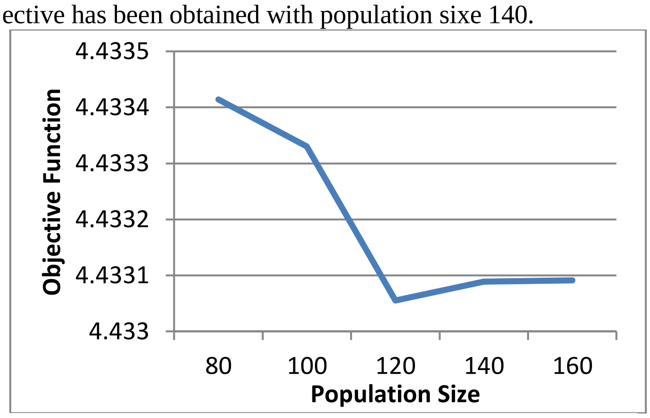

Fig.3: Objective Function versus Population Size with MS-2 for Filter Order 22 for DE algorithm

From the above Fig.3 it has been that minimum value of objective function has been obtained with population size 120. Now keeping the size of population as 120 and 140 for DE and HDE respectively, the value of mutation factor has been varied from 0.4 to 1 in steps of 0.2. The results have been shown in Fig.4 and Fig.5 as below:

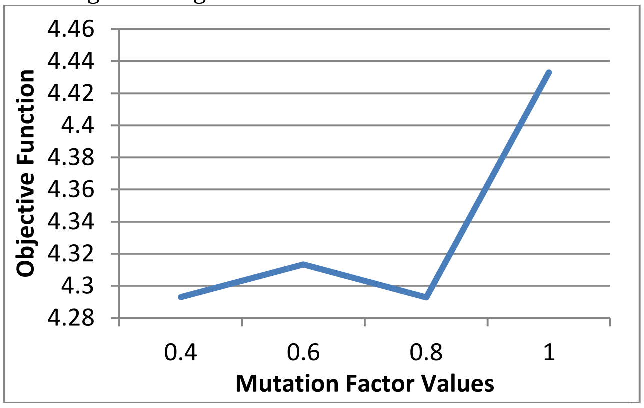

Fig.4: Objective Function Versus Mutation Factor values with Population Size 140 with MS-2 at Filter Order 22 for HDE algorithm

From the above Fig.4 it has been evident that minimum value has been obtained with mutation factor value 0.8.

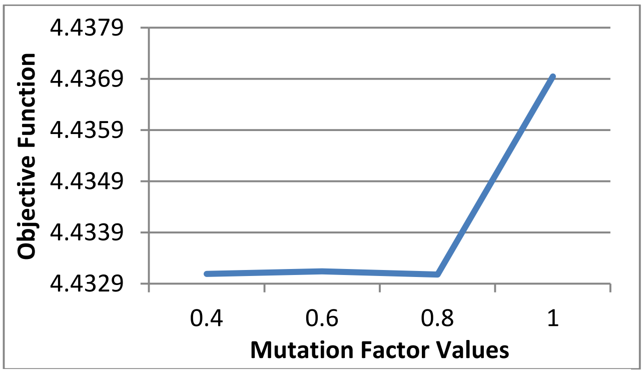

Fig.5: Objective Function Versus Mutation Factor values with Population Size 120 with MS-2 at Filter Order 22 for DE algorithm

So minimum value of objective function has been obtained with mutation factor value as 0.8. Now the value of crossover constant has been varied for both HDE and DE algorithm. Their performance has been drawn in Fig. 6 and Fig. 7 respectively.

It is evident from Fig.4 and Fig.5 that crossover rate 0.4 and 0.3 yields the best possible value of objective function for both HDE and DE algorithms respectively.

B. PSO algorithm Results

A low-pass FIR digital filter has been also designed using PSO algorithm. Selected order for the design of low-pass FIR filter is 22. Initially the objective function is achieved as 4.835267. To improve the value of objective function parameter tuning has been investigated. Initially the population size has been varied from 80 to 160 in steps of 20. The Fig.8 depicts the performance.

It is evident from the above Fig. 8 that optimum value of objective function has been attained with population size 80. Now keeping the population size as 80 the value of acceleration constant has been varied from 0.4 to 2 in steps of 0.4. The Fig.9 shows the variance in objective function with varied value of acceleration constant.

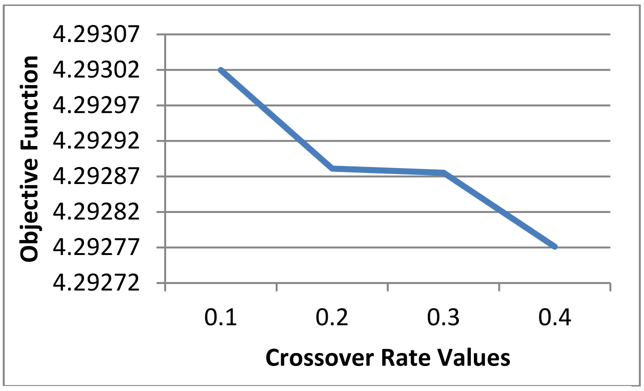

Fig.6: Objective Function attained for different values of CR with mutation factor as 0.8 with MS-2 at Filter Order 22 for HDE algorithm

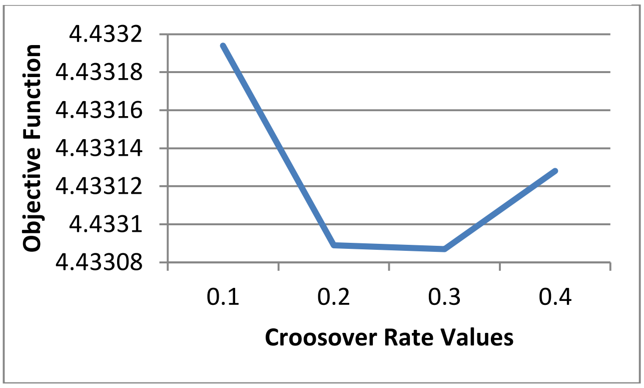

Fig.7: Objective Function attained for different values of CR with mutation factor as 0.8 with MS-2 at Filter Order 22 for DE algorithm

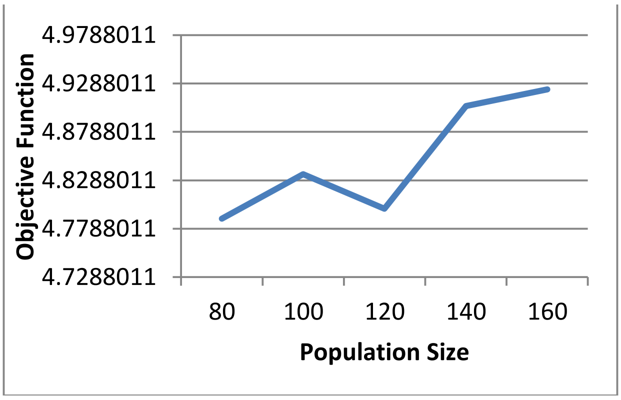

Fig.8: Objective Function versus Population Size for Filter Order 22 for PSO algorithm

C. Comparison between PSO, DE and HDE algorithm

A low-pass FIR digital filter has been designed using two nature inspired optimization techniques named DE and HDE algorithms. Table I draws a comparison between the parameter values of DE and HDE algorithm.

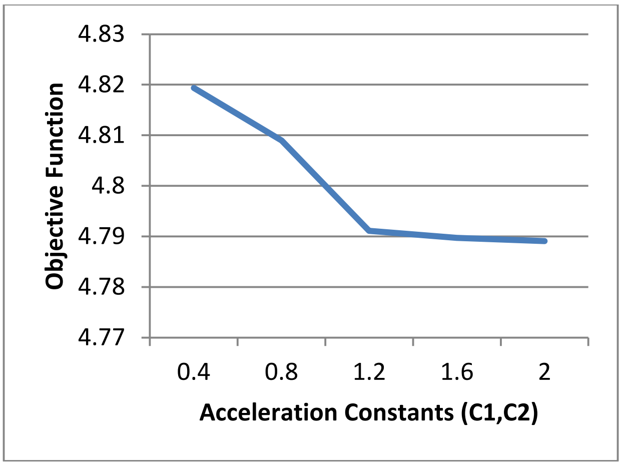

Fig.9: Objective Function versus Acceleration Constants with Population Size 80 for Filter Order 22 with PSO algorithm

Table-1: Parameters Value compared for PSO, DE and DE Algorithm

Parameters	PSO	DE	HDE
Filter Order	22	22	22
Mutation Strategy	-	DE/best/1	DE/best/1
Population Size	80	120	140
Mutation Factor	-	0.8	0.8
Crossover Rate	-	0.3	0.43
Acceleration Constants	2.0,2.0	-	-

The following tables compare PSO, DE and HDE algorithm's performance.

Table-2: Design Values For PSO, DE and HDE Algorithm

Parameters	PSO	DE	HDE
Magnitude Error-1	2.266943	2.0007	2.0066
Magnitude Error-2	0.284138	0.2554	0.2395
Pass-Band Performance	$.9014 \leq \lvert H(\omega) \rvert \leq 1.0208$	$.8926 \leq \lvert H(\omega) \rvert \leq 1.0262$	$.8846 \leq \lvert H(\omega) \rvert \leq 1.0285$
Stop-Band Performance	$\lvert H(\omega) \rvert \leq .0765$	$\lvert H(\omega) \rvert \leq 0.0864$	$\lvert H(\omega) \rvert \leq 0.0583$

To check the robustness of both algorithms standard deviation has been calculated and results have been depicted in Table-3 as follows:

Table-3: Maximum, Minimum and Average Value of Objective Function along with standard deviation

Objective Function Value	PSO	DE	HDE
Maximum Value	4.7946	4.4332	4.2943
Minimum Value	4.7890	4.3308	4.2927
Average Value	4.7893	4.3314	4.2943
Standard Deviation	0.0006	0.5574	0.00024

From the results depicted in Table-3 it has been evident that HDE performs better than DE and PSO algorithm. Magnitude and Phase response of HDE algorithm has been plotted using MATLAB as follows in Fig.10 and Fig.11 respectively:

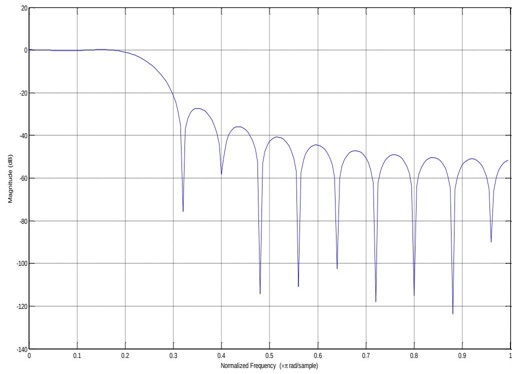

Fig.10: Magnitude Response for Low-Pass FIR digital Filter with MS-2 at Filter Order 22

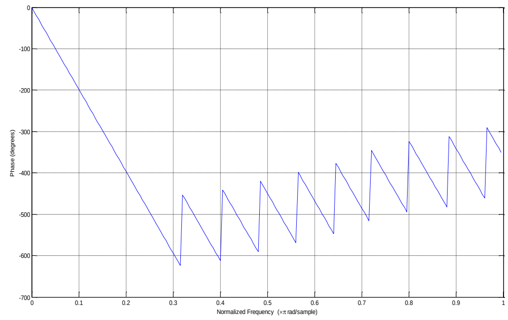

Fig.11: Phase Response for Low-Pass FIR digital Filter with MS-2 at Filter Order 22

VI. CONCLUSION

DE algorithm is an evolutionary algorithm with few control parameters, simple implementation and fast convergence speed. This paper compared traditional DE with hybrid differential algorithm and particle swarm optimization techniques for designing a low-pass FIR digital filter. Parameter tuning of all three algorithms has been investigated to improve the minimum value of objective function. Results depicted that HDE has a slight upper hand when compared to DE algorithm and PSO algorithm. Standard deviation of objective function for all three algorithms is less than 1 which shows the robustness of designed filter using HDE and DE and PSO algorithm.

REFERENCES

[1] E.C Ifeacher and B.W Jervis, "Digital Signal Processing: A Practical Approach", Prentice Hall, South Asia, Second Edition, 2002.

[2] John G. Proakis and Dimitris G. Manolakis, "Digital Signal Processing", Pearson Prentice Hall, Fourth Edition, 2007.

[3] Kalyanmoy Deb, "Optimization for engineering Design", PHI Learning Pvt. Ltd. New Delhi, First Edition, 2004.

[4] Balraj Singh, J.S. Dhillon and Y.S. Brar, "A Hybrid Differential Evolution Method for the Design of Digital FIR Filter", International Journal on Signal and Image Processing, vol. 1, no. 1, pp: 1-10, 2013.

[5] Bipuel Luitel and Ganesh K. Venayagamoorthy, "Differential Evolution, Particle Swarm Optimization for Digital Filter Design, IEEE world congress on computational intelligence, Honkong, pp: 3954-3961, 1-6 June 2008.

[6] Ranjit Kaur and Manjit Singh Patterh and J.S. Dhillon, "Digital IIR Filter Design using Real Coded Genetic Algorithm", International Journal Information Technology and Computer Science, vol. 5, no. 7, pp: 27-35, 2013.

[7] S. Chattopadhyay, S.K. Sanyal and A. Chandra, "Design of FIR Filter Using Differential Evolution Optimization & to Study its Effect as a Pulse-Shaping Filter in a QPSK Modulated System", vol. 10, no. 1, pp: 313-322, 2010.

[8] S.M. Shamsul alam and Md. Tariq Hasan, "Performance Analysis of FIR Filter Design by using Optimal, Blackman Window and Frequency Sampling Method", International Journal of Electrical and Computer Science (IJECS), vol. 10, pp: 9-14, February 2010.

[9] Rainer Storn and Kenneth Price, "Differential Evolution- A simple and efficient heuristic for global optimization over continuous spaces", Journal of Global Optimization, vol. 11, no. 4, pp: 341-359, 1997.

[10] Swagatam Das and Ponnuthuari Nagaratnam Suganthan, "Differential Evolution: A Survey of the State of the Art", IEEE Transactions on Evolutionary Computation, vol. 15, no.1, pp: 1-28, February 2011

[11] Chumeri Zhang, Jie Chen and Bin Xin, "Distributed mesmetic differential evolution with the synergy of Lamarckian and Baldwinian learning", Applied Soft Computing, vol. 13, pp: 2947-2959.

[12] Rajni and Balraj Singh, "A Hybrid Differential Evolution Method for the Design of High Pass Digital FIR Filter" International Journal of Computer Science and Mobile Computing, vol. 4, no. 7, pp: 438-445, 2015.

[13] D.P Kothari, J.S. Dhillon, "Power System Optimization", PHI learning Pvt. Ltd New Delhi, second edition.

[14] Efren Mezura-Montes, Jesus Velazquez-Reyes and Carlos A. Coello Coello, "Modified Differential Evolution for Constrained Optmization", IEEE Congress on Evolutionary Computation, Canada.

[15] Abhijit Chandra and Sudipta Chattopadhay, "Role of Mutation Strategies of Differential Evolution Algorithm in designing Hardware Efficient Multiplier-less Low–Pass FIR Filter", Journal of multimedia, vol. 7, no. 5, pp: 353-363, October 2012

[16] Meisam Najjarzadeh and Ahmad Ayatoolahi, "FIR Digital Filter Design, PSO: Utilizing LMS and Minimax Strategies", IEEE Symposium on Signal Processing and Information Technology, pp: 129-132, 2008.

[17] Ranjit Kaur and Damanpreet Singh, "Particle Swarm Optimization Algorithm for Designing Optimal IIR Digital Filter", International journal of emerging technologies in computational and applied sciences (IJETCAS), vol. 2, no. 7, pp: 225-230, 2014

[18] Sangeeta Mandal, Prabisha Mallick, Durbadal Manda, Rajib Kaur, Sabti Prasad Ghoshal, "Optimal FIR Band-Pass Filter design using Novel PSO Algorithm", IEEE Symposium on humanities, Science and Engineering Research, 2012.