Volume 4, Issue 9, September 2014 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

# Design of Low Pass FIR Filter using Artificial Bee Colony Optimization Technique and its Comparison with Particle Swarm Optimization

**Er. Karamjeet Singh**
ECE Department& BBSBEC,
Fatehgarh Sahib, India

**Gurpreet Kaur**
ECE Department& BBSBEC,
Fatehgarh, India

***Abstract— Digital Signal Processing (DSP) is one of the most powerful technologies that are shaping science and engineering in the twenty-first century. Revolutionary changes have already been made in a broad range of fields: communications, medical imaging, radar and sonar, and high fidelity music reproduction, to name just a few. Each of these areas has developed a comprehensive DSP technology, with its own algorithms, mathematics, and specialized techniques. Analog (electronic) filters can be used for these tasks, as these are cheap, fast, and have a large dynamic range in both amplitude and frequency; however, digital filters are vastly superior in the level of performance. Digital filters are an essential part of DSP. The purpose of the filters is to allow some frequencies to pass unaltered, while completely blocking others. The digital filters are mainly used for two purposes: separation of signals that have been combined, and restoration of signals that have been distorted in some way. In this present work, Digital FIR filter is designed using Artificial Bee Colony (ABC) optimization technique and its comparison is done with particle swarm optimization.***

***Keywords—Digital filters, Low pass filters, Finite impulse response filter, Artificial Bee Colony (ABC) optimization technique, fitness function.***

## 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 digital filter is a discrete system, and it can do a series of mathematic processing to the input signal, and therefore obtain the desired information from the input signal. The transfer function for a linear, time-invariant, digital filter is usually expressed as:

$$H(Z) = \sum_{j=0}^{M} \frac{b^j \ z^{-j}}{1 + \sum_{i=1}^{N} a_i \ z^{-i}}$$

where $a_i$and $b_i$are coefficients of the filter in Z-transform.

According to the formula above, if $a_i$s always zero, then it is a FIR filter, otherwise, if there is at least one none-zero $a_i$, then it is an IIR filter.

## II. TYPES OF DIGITAL FILTERS

The two types of Digital filters are:

A. . FIR (Finite Impulse Response) Filters
B. . IIR (Infinite Impulse Response) Filters

### *A. FIR (Finite Impulse Response Filter)*

The finite impulse response (FIR) filter is one of the most basic elements in a digital signal processing system, and it can guarantee a strict linear phase frequency characteristic with any kind of amplitude frequencycharacteristic. Besides, the unit impulse response is finite; therefore, FIR filters are stable system. The FIR filter has a broad application in many fields, such as telecommunication, image processing, and so on. The Response of FIR filters decays with increase in time. The output depends upon present and previous inputs. It is a non recursive filter characterized by equation:-

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

……………. (1)

where h (k) are the impulse response coefficients of the Filter, N is the filter Length, y (n) is the output at discrete time instance n and x (n-k) is the filter input delayed by k samples.

### B. *IIR (Infinite Impulse Response filter)*

The infinite impulse response (IIR) filter is recursive structure, and it has a feedback loop. The precision of amplitude frequency characteristic is very high, and IIR filters are not linear phase. The Response of IIR filter does not die out with the increase in the value of time and extends up till infinity. The output depends upon one or more previous output values. It is characterized by equation as follows:-

$$y(n) = \sum_{k=0}^{\infty} h(k)x\,(n-k)$$

…………… (2)

The advantage of using FIR filter in comparison to IIR filter can be summarized as:-

(1) They have exactly linear Phase Response which is an important requirement for data transmission as phase response of IIR filters is nonlinear at the edges.
(2) They have guaranteed Stability.
(3) The FIR filter is non-recursive structure, finite precision arithmetic error is very small. While IIR filter is recursive structure, and parasitic oscillation may occur in the operation of IIR filter.
(4) Fast Fourier Transformation can be used in FIR filter, while IIR cannot.
(5) Lesser effect of Round off Noise.
(6) Lesser Severe effect of Coefficient quantization errors than in IIR.

## III. FIR FILTER DESIGN

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

$$y(n) = -\sum_{k=1}^{Q} a_k\ y(n-k) + \sum_{k=0}^{P} b_k\ x(n-k) \ldots\ldots\ldots\ldots\ldots (3)$$

For a FIR filter, *ak = 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,

$Zero\ Order, y(n) = a_o x(n)$
$First\ Order, y(n) = a_o x(n) + a_1 x(n-1)$
$Second\ Order, y(n) = a_o x(n) + a_1\, x(n-1) + a_2 x(n-2)$

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

$Y(z) = a_o X(z) + a_1 z^{-1} X(z) + a_2 z^{-2} X(z) + \cdots + a_{N-1} z^{-(N-1)} X(z) \ldots\ldots\ldots\ldots\ldots (4)$

The corresponding transfer function will be,

$T(z) = \frac{Y(z)}{X(z)} = a_o + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_{N-1} z^{-(N-1)} \ldots\ldots\ldots\ldots\ldots (5)$

To design a filter, various filter parameters come intopicture. They are passband and stopband cut-off frequencies(wp and ws), passband and stopband ripple (δp and δs) attenuation factor in stopband and transition width. Theparameters are totally determined by the filter coefficients . For a design problem, a certain set of desired parameterswill be given. The job is to find the filter coefficients in such away so that the deviation of these parameters from the desiredvalue will be minimum. In present case the job is to design aLow pass filter with a given set of parameters. Figure 1 showsa typical example of how a designed filter deviates incharacteristics from an ideal filter. For the filter shown in figure1, the passband and the stopband frequencies are 0.3 and 0.4(normalized with respect to π) and the transition width (ws-wp)is 0.1. The passband and stopband ripples are shown.

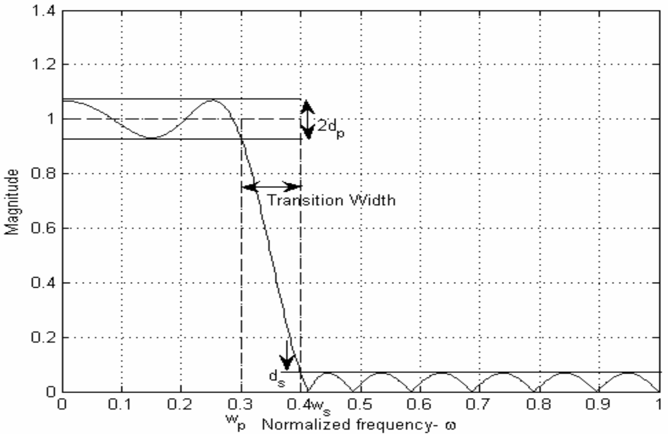



Figure 1. Ideal and Actual filter Magnitude response showing passband andstopband ripples and transition band in actual filter

## IV. ABC ALGORITHM

### 1. Introduction

In recent years, many swarm intelligence methods have been proposed for solving hard optimization problemsdue to their simple structures and production of effective solutions for problems in a reasonable time. Theartificial bee colony (ABC) algorithm, which is one of the most popular swarm intelligence algorithms, wasfirst introduced by Karaboga in 2005 for numerical optimization problems and was inspired by the intelligentbehaviors of real honey bee colonies. For solving numerical optimization problems, the ABC was designedusing the interaction in the swarm and there are 2 types of bees in the ABC, the employed and unemployedbees. Employed bees try to find food sources that represent a feasible solution for the optimization problemusing the interaction in the employed bee population. Employed bees also move the position information aboutthe food sources to the ABC hive. Unemployed foragers consist of onlooker bees and scout bees. The onlookerbees try to improve the solution of the employed bees by considering the information shared by the employedbees. If a food source could not be improved in a certain time, the employed bee of this food source becomes ascout bee. After a new food source is randomly generated for the scout bee, this scout bee becomes an employedbee. The basic ABC algorithm consists of 4 phases, named initialization, employed, onlooker, andscout bee phases.

#### *1.1. Initialization phase*

In this phase, a feasible solution is produced for each employed bee using Eq. (6) and the abandonment countersof the employed bees that will be used for testing the limit are reset.

$$X_i^j = X_j^{min} + R \times \left(X_j^{max} - X_j^{min}\right) \; i = 1,2,\ldots\ldots,N \; and \; j = 1,2,\ldots\ldots D \ldots\ldots\ldots\ldots\ldots (6)$$

Here, $X_i^j$ is the jth dimension of the ith employed bee, and $X_j^{min}$ and $X_j^{max}$ are the lower and upper bounds of the jth dimension, respectively. R is a random number in the range of [0,1], N is the number of employedbees, and D is the dimensionality of the problem.After producing new solutions for the employed bees, the fitness values of the solutions are calculatedusing Eq. (7):

$$fit_i = \begin{cases} \frac{1}{1+f_i} & if \; (f_i \geq 0) \\ 1 + abs\,(f_i) & if \; (f_{i<} \; 0) \end{cases} \quad ,\ldots\ldots\ldots\ldots\ldots (7)$$

Where $fit_i$the fitness value of the ith is employed bee (or solution) and $f_i$ is the objective function value,specific for the optimization problem, of the ith employed bee.

#### *1.2. Employed bee phase*

Each employed bee tries to improve its self-solution using Eq. (8). If the fitness value of the new candidatesolution obtained by Eq. (3) is better than the old one, then the employed bee memorizes the new solution andits abandonment counter is reset; otherwise, the abandonment counter of the employed bee is increased by 1 (let $V_i = X_i$) :

$$V_j^i = X_j^i + \varphi \times \left(X_j^i - X_k^j\right), \quad ik\epsilon \{1,2,\ldots\ldots,N\}, \; j\epsilon \{1,2,\ldots\ldots,D\} \; and \; i \neq k \,, \; \ldots\ldots\ldots\ldots\ldots (8)$$

where $V_j^i$is the j th dimension of the ith candidate solution, $X_j^i$is the j th dimension of the ith employed bee,$X_k^j$is the j th dimension of the neighbor employed bee, and φ is a random number in the range of [{1, + 1}]. Itshould be mentioned that only 1 dimension of the solution of an employed bee is updated at each iteration, andthis dimension and neighbor bee chosen from the employed bee population are randomly selected in Eq. (8).

#### *1.3. Onlooker bee phase*

The onlooker bees wait for information about food sources that will be shared by the employed bee in the hive.After the employed bees return to the hive, the employed bees share the position information of the self-solutionswith onlooker bees in the dance area of the hive. The onlooker bees select an employed bee in order to improveits solution using Eq. (9):

$$p_i = \frac{0.9 \times fit_i}{fit_{best}} + 0.1\ldots\ldots\ldots\ldots\ldots (9)$$

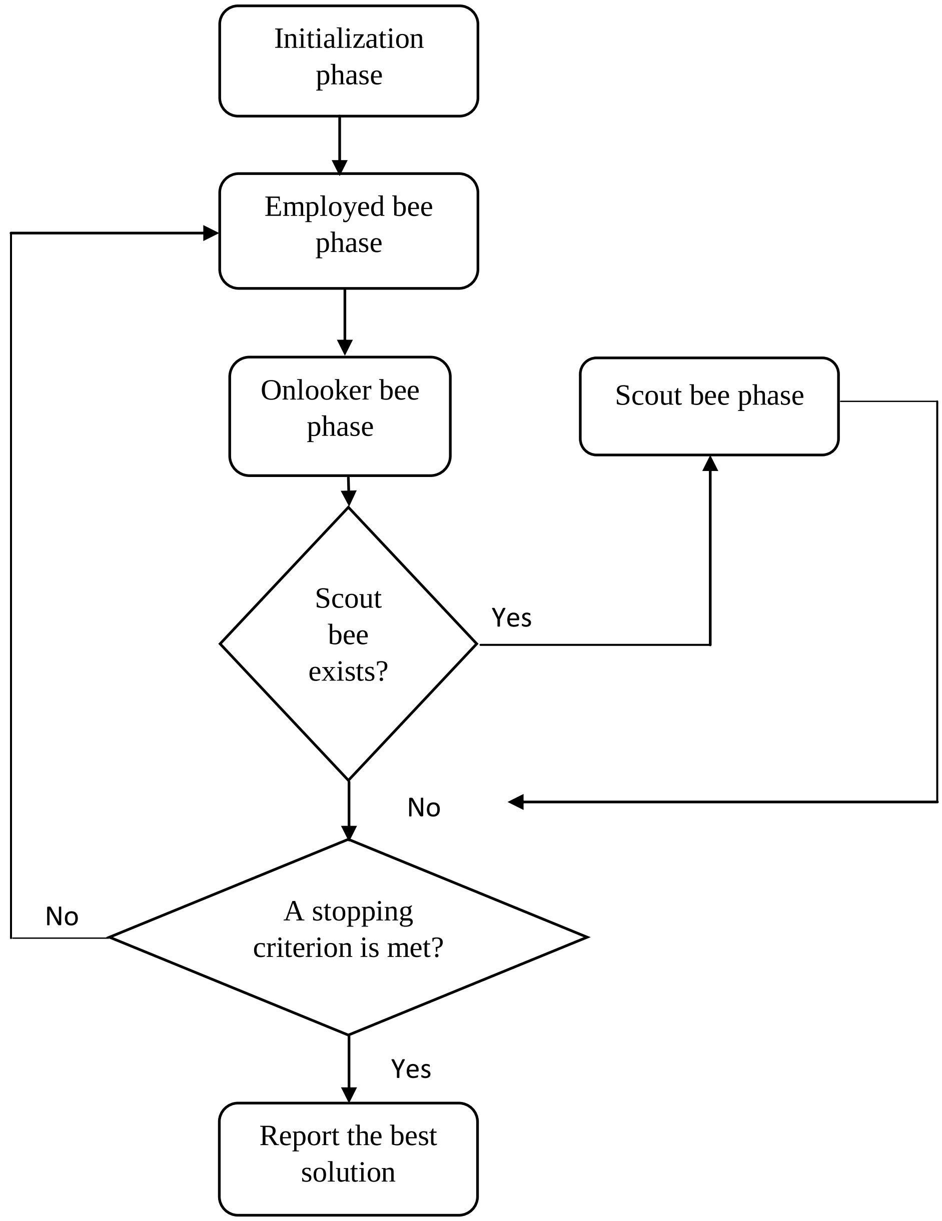
Where $p_i$ is the selected probability of the ith employed bee, $fit_i$ is the fitness value of the ith employed bee,$fit_{best}$ is the best solution in the employed bee population, and N is the number of employed bees.After an employed bee is selected, the solution of the employed bee is updated using Eq. (3). If thenew solution is better than the old solution, the solutions are replaced and the abandonment counter of theemployed bee is reset. Otherwise, the abandonment counter of the employed bee is increased by 1.

#### *1.4. Scout bee phase*

In this phase, the abandonment counter with the maximum content is fixed and this content is compared withthe limit. If the content is higher than the limit, the employed bee of this counter becomes a scout bee. If a scoutbee occurs at the ABC iteration, a new solution is generated for the scout bee using Eq. (6), the abandonmentcounter is reset, and the scout bee becomes an employed bee.Despite the fact that the best solution of the population is not directly used for updating the solution bythe employed or onlooker bees for the population, the best solution obtained so far is stored at each iteration.The ABC is an iterative algorithm and the maximum evaluation number or maximum iteration number can beused for termination of the algorithm. In addition, the number of employed bees and onlooker bees is equal toeach other and only 1 scout bee can occur at each iteration.

**2. Flowchart of ABC Algorithm:**




## V. PSO (PARTICLE SWARM OPTIMIZATION) BASED DESIGN APPROACH

A swarm is basically a disorganized collection of moving individuals which tend to cluster together while each individual seems to be moving in some random direction. PSO is a population based optimization approach introduced in 1995 by James Kennedy and Russell C. Eberhart (Parsopoulos and Vrahatis, 2010). It is inspired by the observation of natural habits of bird flocking and fish schooling. Particle swarm optimization (PSO) is a computational method that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality. PSO optimizes a problem by having a population of candidate solutions, here dubbed particles, and moving these particles around in the search-space according to simple mathematical formulae over the particle's position and velocity. Each particle's movement is influenced by its local best known position but, is also guided toward the best known positions in the search-space, which are updated as better positions are found by other particles. This is expected to move the swarm toward the best solutions.

It consists of a number of particles called as agents moving in search space looking for best solution i.e. best value of fitness. Each particle is treated as a point in D dimensional search space which tries to adjust its flying according to its own flying experience as well as flying experience of neighboring particles. The particles in the search process are the potential solutions, which move around the defined search space with some velocity until the error is minimized or the solution is reached, as decided by the fitness function. PSO is a metaheuristic as it makes few or no assumptions about the problem being optimized and can search very large spaces of candidate solutions. However, metaheuristics such as PSO do not guarantee an optimal solution is ever found. More specifically, PSO is a pattern search method which does not use the gradient of the problem being optimized, which means PSO does not require that the optimization problem be differentiable as is required by classic optimization methods such as gradient descent and quasi-newton methods. PSO can therefore also be used on optimization problems that are partially irregular, noisy, change over time, etc.

## VI. 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. In this paper, the desired ripple on the passband $d_p$is 0.1 and that on the stopband $d_s$ is 0.01. In each case, passband and stopband cut off frequencies are 0.25 and 0.3 respectively. The passband ripple is 0.1 and stopband ripple is 0.01. Filters with 20 coefficients are designed.The experiment has been implemented in MATLAB. The coefficients obtained from the filter design have been listed in Table 1

### 1. Artificial Bee Colony optimization results:-

#### *1.1Control Parameters of ABC algorithm:*

Table 1 control parameters of ABC

| Parameter | Values |
|---|---|
| Colony size | 30 |
| Food number | 15 |
| Limit | 100 |
| Maximum cycle | 200 |

The coefficients obtained from the filter design have been listed in Table 2.
The table indicates the value of coefficients obtained using ABC. Since the first and last coefficients are same, second and second last coefficients are same and so on, thereby satisfying linearity condition.

Table 2 Filter coefficients using ABC

| **Filter coefficients** | **Values** |
|---|---|
| h(1)=h(20) | -0.0200 |
| h(2)=h(19) | -0.0300 |
| h(3)=h(18) | -0.0162 |
| h(4)=h(17) | 0.0240 |
| h(5)=h(16) | 0.0544 |
| h(6)=h(15) | 0.0476 |
| h(7)=h(14) | -0.0035 |
| h(8)=h(13) | -0.1052 |
| h(9)=h(12) | -0.2082 |
| h(10)=h(11) | -0.2681 |

The magnitude and gain plots thus obtained from the design have been plotted. Fig.2 shows the error graph. It can be observed that ABC converges to a much lower error in lesser number of iterations.

### 2. Calculation of Mean Squared Error:

Mean Squared Error is calculated using the formula given below:

$$J_3 = 1\Big/N_f \sum_{k=1}^{N_f} (ideal(k) - actual(k))^2$$

(Luitel and Venayagamoorthy, 2008)
…… (10)

Where $ideal(k)$ and $actual(k)$ are the magnitude response of the ideal and the actual filter, and $N_f$ is the number of samples used to calculate the error.
Here, the value of Mean Squared Error using ABC optimization technique comes out to be 0.0029.
Error graph for the case of ABC:

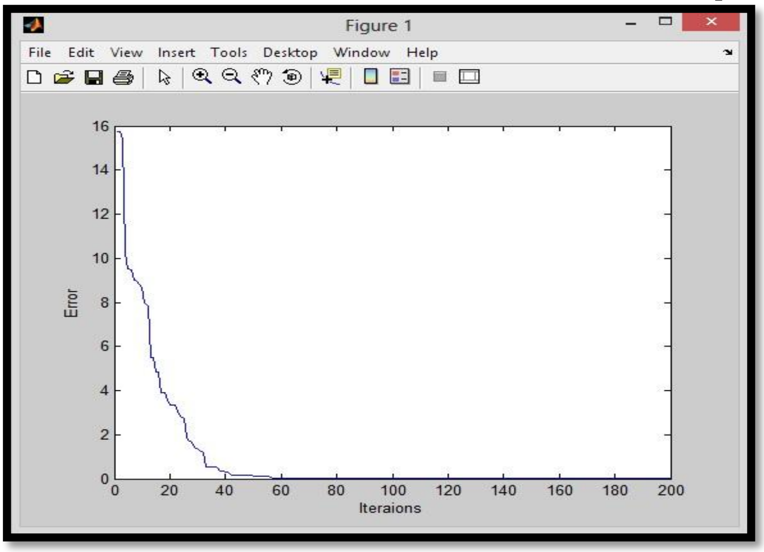



Figure 2 Error graph for ABC

From the Figure 2 it is inferred that ABC has converged to a much lower error in lesser number of iterations.
The magnitude and gain plots thus obtained from the design have been plotted are shown below:
From the Figure 3 it is inferred that there are some ripples in passband of low-pass FIR filter and very lesser amounts of ripples in stopband of low-pass FIR filter. The transition bandwidth is near to the optimal method.
From the Figure 4 it is inferred that there is a flat passband and ripples are present in the stopband of low-pass FIR filter.

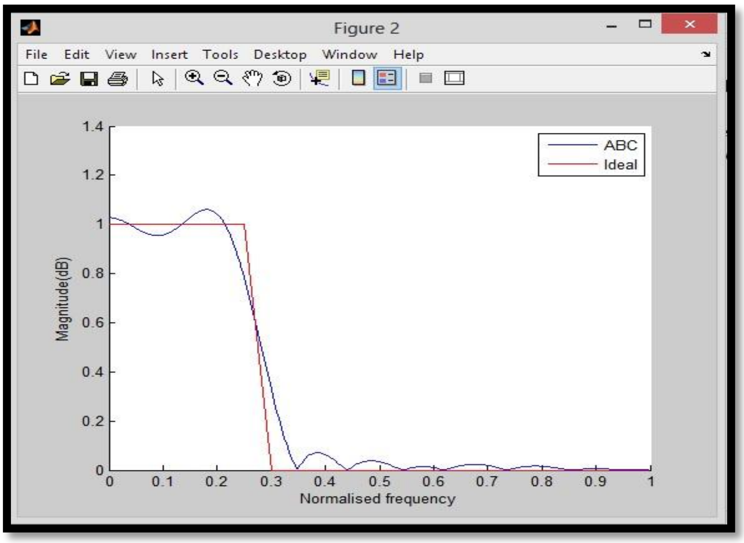



Figure 3 Magnitude response of Low-pass filter designed using ABC

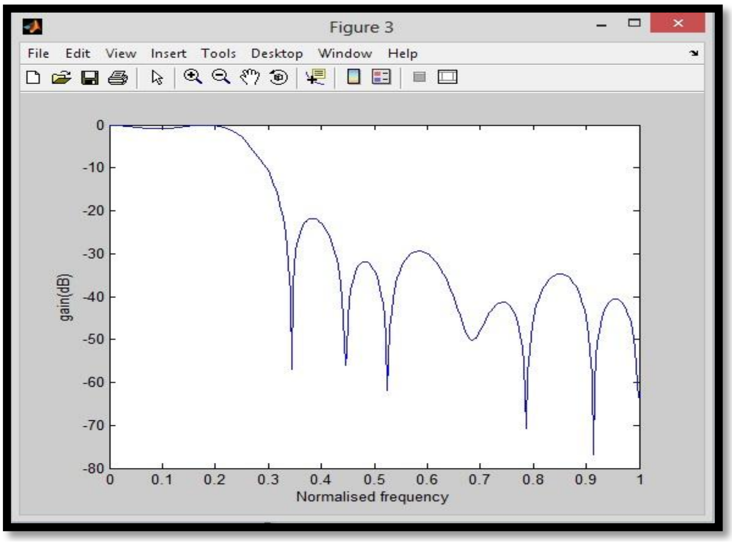



Figure 4 Gain plot of Low-pass filter designed using ABC

Table 3 Error values of ABC for different iterations.

| Iterations | MSE value |
|---|---|
| 200 | 0.0029 |
| 40 | 0.2251 |
| 10 | 3.2362 |

## 3. Comparison of ABC and PSO:

Comparison of ABC and PSO is done by plotting magnitude response and gain plot of low-pass FIR filter designed by using ABC optimization technique and PSO.

### *3.1 Comparison of ABC and PSO on the basis of gain plotting:*

In ideal case there should be flat pass band, flat stop band and a very small transition band width.
From the gain comparison plot figure, it is indicated that in case of:

1. Artificial Bee Colony (ABC) optimization technique: There is a flat passband and ripples are present in stopband region of low-pass FIR filter. Transition bandwidth is near to optimal value.
2. Particle Swarm Optimization (PSO): There are small ripples in passband region of low-pass FIR filter and large amount of ripples are present in stopband region of low-pass FIR filter. Transition bandwidth is same as that for the ABC.

Hence ABC has relatively better gain response as compare to PSO.

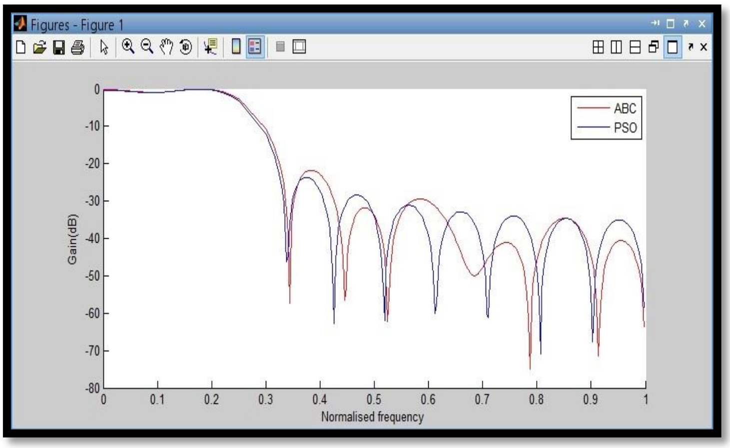



Figure 5 comparison plot for the gain of ABC and PSO

### *3.2 Comparison of ABC and PSO on the basis of magnitude plotting:*

In ideal case there should be flat pass band, flat stop band and a very small transition band width.
From the gain comparison plot figure, it is indicated that in case of:

1. Artificial Bee Colony (ABC) optimization technique: There are small amount of ripples in passband and stopband region of low-pass FIR filter. Transition bandwidth is near to optimal value.
2. Particle Swarm Optimization (PSO): There are large amount of ripples in passband region of low-pass FIR filter and large amount of ripples are present in stopband region of low-pass FIR filter. Transition bandwidth is same as that for the ABC.Amplitude of ripples present both in passband and stopband region in case of PSO is greater as compare to ABC.Hence ABC has relatively better magnitude response as compare to PSO.

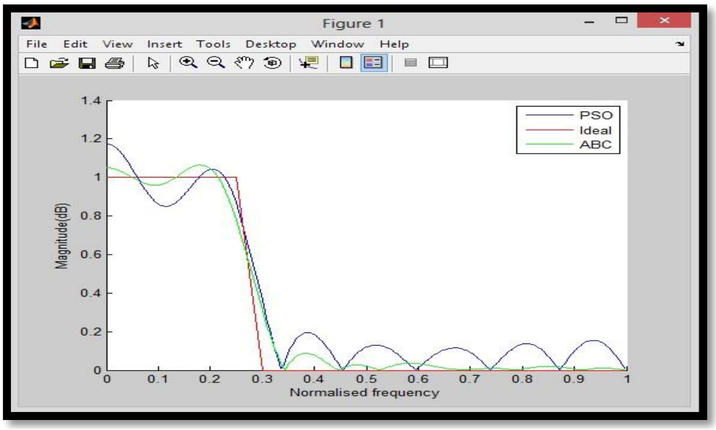



Figure 6 comparison plot for the magnitude of ABC and PSO

***3.3 Mean squared error comparison:***

Mean squared error values for both ABC and PSO is shown below in the table

Table 4 Error values of ABC and PSO

| Algorithm | MSE value |
|---|---|
| ABC | 0.0029 |
| PSO | 0.0030 |

## VII. CONCLUSIONS AND FUTURE SCOPE

**Conclusion**

The main focus of the thesis is on the design of digital FIR filters. In digital filter design, Low-pass FIR filter has been designed. In this work, Artificial Bee Colony (ABC) optimization technique is used as the baseline algorithm. Present work is compared with Particle Swarm optimization (PSO). The following conclusions are drawn from the results of FIR filter.

- Mean squared error achieved for the case of ABC is less as compare to PSO.
- ABC has relatively better magnitude response as compare to PSO.
- Gain response of ABC is also better as compare to PSO.
- The performance of the ABC is better than or similar to those of other evolutionary-based algorithms with the advantage of employing fewer control parameters and it can be efficiently used for solving multimodal and multidimensional optimization problems.

**Future Scope**

- Design of High pass Filters can be done.
- Design of Adaptive Filters can be done.
- Design of Higher order Filters and Filters with multiple passband and Stopband can be explored.

**REFERENCES**

[1] D. Karaboga, An Idea Based On Honey Bee Swarm For Numerical Optimization, Technical Report TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.

[2] B. Basturk, D.Karaboga, An Artificial Bee Colony (ABC) Algorithm for Numeric function Optimization, IEEE Swarm Intelligence Symposium 2006, May 12-14, 2006, Indianapolis, Indiana, USA.

[3] D. Karaboga, B. Basturk, A. powerful, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm, Journal of Global Optimization 39 (3) (2007) 459–471.

[4] D. Karaboga, B. Basturk, On the performance of artificial bee colony (abc) algorithm, Applied Soft Computing 8 (1) (2008) 687–697.

[5] D. Karaboga, B. Basturk Akay, An artificial bee colony (abc) algorithm on training artificial neural networks, in: 15th IEEE Signal Processing and Communications Applications, SIU 2007, Eskisehir, Turkiye, June, pp. 1–4.

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

[7] Alok Singh, An artificial bee colony algorithm for the leaf-constrained minimum spanning tree problem, Applied Soft Computing 9 (2) (2009) 625–631.

[8] D. Karaboga, C. Ozturk, Neural networks training by Artificial Bee Colony Algorithm on pattern classification, Neural Network World 19 (3) (2009) 279–292.

[9] J. Kennedy, R. C. Eberhart, “Particle swarm optimization”, In Proceedings of the 1995 IEEE International Conference on Neural Networks”, Vol. 4, pp. 1942–1948,1995.

[10] Ababneh, J. I. and Bataineh, M.H. (2008), “Linear phase FIR filter design using particle swarm optimization and genetic algorithms”, Journal of Digital Signal Processing, vol.18, 4, pp. 657–668.

[11] Arumugam, M. S.; Loo, C.K.; Murthy, G. R. and Rao, M.V.C. (2007), “A Novel Effective Particle Swarm Optimization Like Algorithm via Extrapolation Technique”, Proceedings of International Conference on Intelligent and Advanced Systems, Kuala Lumpur, Malaysia, pp. 516-521.

[12] Ayatollahi, A. and Najjarzadeh, M. (2008), “A Comparison between Genetic Algorithm and PSO for Linear Phase FIR Digital Filter Design”, Proceedings of International conference on Signal Processing (ICSP), Beijing, China, pp. 2134-2137.

[13] Bansal, J.C. and Deep, K. (2009), “Mean particle swarm optimisation for function optimisation”, International Journal Computational Intelligence Studies, vol.1, 1, pp. 72-92.

[14] Clerc, M. and Kennedy, J. (2002), “The Particle Swarm—Explosion, Stability and Convergence in a Multidimensional Complex Space”, IEEE Transactions on Evolutionary Computation, vol. 6, 1, pp. 58-73.

[15] Ghoshal, S.P.; Kar, R. and Mandal, D. (2011), "Linear phase FIR High pass filter design using PSO-CFIWA with Least Square Approach", Proceedings of IEEE Symposium on Industrial Electronics and Applications (ISIEA), Langkawi, Malaysia, pp. 314-319.
[16] Ghoshal, S.P.; Kar; R.; Mandal, S.; Mandal, D.; Mukherjee, P. and Sengupta, D. (2011), "Design of Optimal Linear Phase FIR High Pass Filter using Improved Particle Swarm Optimization", Proceedings of International Conference on Advances in Computer Science, Noida, India, pp. 5-9.
[17] Hashim, S.Z.M; Malik, R.F.; Ngah, R. and Rahman, T.A. (2007), "New Particle Swarm Optimizer with Sigmoid Increasing Inertia Weight", International Journal of Computer Science and Security, vol.1, pp. 35-44.
[18] Ifeachor, E.C. and Jervis, B.W. (2001), "Digital Signal Processing: A Practical Approach", Prentice Hall Publishers, U.S.A.
[19] Lin, M. and Hua, Z. (2009), "Improved PSO Algorithm with Adaptive Inertia Weight and Mutation", Proceedings of World Congress on Computer Science and Information Engineering, Los Angels, U.S.A, pp. 622-625.
[20] Luitel, B. and Venayagamoorthy, G. K. (2008), "Particle Swarm Optimization with Quantum Infusion for the Design of Digital Filters", Proceedings of IEEE Swarm Intelligence Symposium, St. Louis MO, USA, pp. 1031-1035.
[21] Luitel, B. and Venayagamoorthy, G. K. (2008), "Differential Evolution Particle Swarm Optimization for Digital Filter Design", Proceedings of IEEE Congress on Evolutionary Computation, Hong Kong, China, vol.2, 3, pp. 3954-3961.
[22] Ni, Q. and Xing, H. (2006), "An Improved Gaussian Dynamic Particle Swarm Optimization Algorithm", Proceedings of International Conference on computational Intelligence and Security, Guangzhou, China, pp. 316-319.
[23] Parsopoulos, E. and Vrahatis, M. N. (2010), "Particle Swarm Optimization and Intelligence: Advances and Applications", Information Science Reference, Hershey, New York.
[24] Tang, J. and Zhao, X. (2010), "A Hybrid Particle Swarm Optimization with Adaptive Local Search", Journal of Networks, vol. 5, 4, pp. 411-418.
[25] Teixeira, A.C.F. (2007), "Optimum Finite Impulse Response Digital Filter Design using Computational Intelligence Based Optimization Algorithms", Proceedings of International Conference Intelligent Systems and Agents, Lisbon, Portugal, pp.212- 216.
[26] Valle, Y.D.; Mohagheghi, S.; Hernandez, J.C. and Harley, R.G. and Venayagamoorthy, G. K. (2008), "Particle Swarm Optimization: Basic Concepts, Variants and Applications in Power Systems", IEEE Transactions on Evolutionary Computation, vol. 12, 2, pp. 171-195.
[27] Wang, H. and Liu, Y. (2007), "A Hybrid Particle Swarm Algorithm with Cauchy Mutation", Proceedings of IEEE Swarm Intelligence Symposium, Honolulu, Hawaii, pp.356-360.
[28] A.R. Baig, M. Rashid, Foraging for fitness: A Honey Bee Behaviour Based Algorithm for Function Optimization, Technical Report, NUCES, Pakistan, November 2006.
[29] P-W. Tsai, J-S. Pan, B-Y. Liao, S-C. Chu, Interactive artificial bee colony (IABC) optimization, ISI2008 (Taiwan), December 12, 2008.
[30] V. Nayak, H. A. Suthar, and J. Gadit, "Implementation of Artificial Bee Colony Algorithm," IAES International Journal of Artificial Intelligence (IJ-AI), vol. 1, no. 3, pp.112–120, September 2012.