RESEARCH ARTICLE OPEN ACCESS

Design of Low Pass Digital FIR Filter Using Cuckoo Search Algorithm

Taranjit Singh*, Harvinder Singh Josan**
*(Research Scholar, Department of Electronics and Communication, RIMT-IET, Mandi Gobindgarh, Punjab)
**(Assistant Professor, Department of Electronics and Communication, RIMT-IET, Mandi Gobindgarh, Punjab)

ABSTRACT
This paper presents a novel approach of designing linear phase FIR low pass filter using cuckoo Search Algorithm (CSA). FIR filter design is a multi-modal optimization problem. The conventional optimization techniques are not efficient for digital filter design. An iterative method is introduced to find the best solution of FIR filter design problem.Flat passband and high stopband attenuation are the major characteristics required in FIR filter design. To achieve these characteristics, a Cuckoo Search algorithm (CSA) is proposed in this paper. CSA have been used here for the design of linear phase finite impulse response (FIR) filters. Results are presented in this paper that seems to be promising tool for FIR filter design

***Keywords* -** Convergence, CSA, Evolutionary Optimization Technique, Magnitude Response, Parks and McClellan Algorithm.

I. INTRODUCTION

A filter is a frequency selective circuit that allows a certain frequency to pass while attenuating the others. Filter could be analog or digital. Analog filters use electronic components such as resistor, capacitor, transistor etc. to perform the filtering operations. The applications of filters are such as they are used for noise reduction, video/audio signal enhancement etc. On other hand, digital filters adopt digital processors which perform mathematical calculations on the sampled values of the signal in order to perform the filter operation. A computer or a dedicated digital signal processor may be used implementing digital filters.

Conventionally, different techniques exist for the design of digital filters. The simplest design of FIR filter is achieved using window method. In this method, ideal impulse response is multiplied with a window function. These various windows limit the infinite length impulse response of ideal filter into finite window to design an actual response. But windowing methods do not allow sufficient control of frequency response in the various frequency bands and other filter parameters such as transition width. Furthermore, the windowing method does not permit individual control over approximate error in various bands. So, better filter result from minimization of maximum error in both stopband and passband of the filter which leads to equiripple filters. Such filters can be achieved using evolutionary methods. Since population based stochastic search methods have proven to be effective in multidimensional nonlinear environment, all of the constraints of filter design can be effectively taken care of by the use of these algorithms i.e. the filter Design can be viewed as optimization problem. There are two types of filter, FIR and IIR filter. FIR filter are known as non-recursive filters and IIR filters are known as recursive filters. These names came from the nature of algorithms used for these filters. Implementation of FIR filters is easy, but it is slower when compared to IIR filters. Though IIR filters are fast, practical implementation is a bit complicated compared to FIR filters [1]. FIR filter is an attractive choice because of the ease in design and stability. By designing the filter taps to be symmetrical about the centre tap position, the FIR filter can be guaranteed to have linear phase. Finite impulse response (FIR) digital filters are known to have many desirable features such as guaranteed stability, the possibility of exact linear phase characteristic at all frequencies and digital implementation as non-recursive structures. Linear phase FIR filters are also required when time domain specifications are given [2]. Traditionally, different techniques exist for the design of digital filters.

Out of these, windowing method is the most popular. In this method, ideal impulse response is multiplied with a window function. There are various kinds of window functions (Butterworth, Chebyshev, Kaiser etc.), depending on the requirements of ripples on the pass band and stop band, stop band attenuation and the transition width. These various windows limit the infinite length impulse response of ideal filter into a finite window to design an actual response. But windowing methods do not allow sufficient control of the frequency response in the various frequency bands and other filter parameters such as transition width. The most frequently used method for the design of exact linear phase weighted Chebyshev FIR digital filter is the one based on the Remez-exchange algorithm proposed by Parks and McClellan [3]. Further improvements in their results have been reported in [4]. The main limitation of this procedure is that the relative values of the amplitude error in the frequency bands are specified by means of the weighting function, and not by the deviations themselves. Therefore, in case of designing high-pass filters with the given stop band deviation, filter length and cut-off frequency, the program has to be iterated many times [5]. A number of models have been developed for the finite impulse response (FIR) filter techniques and design methods. This is a thrust research area, aiming at obtaining more general and innovative techniques that are able to solve and/or optimize new and complex engineering problems [6]. The trade-off has to be made by the designer on one or the other of the design specifications. So, evolutionary methods have been employed in the design of digital filters to design with better parameter control and to better approximate the ideal filter [7]. Different heuristic optimization algorithms such as genetic algorithm (GA) [7] simulated annealing algorithms [8] etc. have been widely used to the synthesis of design methods capable of satisfying constraints which would be unattainable. When considering global optimization methods for digital filter design, the GA seems to be the promising one. Filters designed by GA have the potential of obtaining near global optimum solution. Although standard GA (mostly referred to as Real Coded GA (RGA)) have a good performance for finding the promising regions of the search space, they are inefficient in determining the local minimum in terms of convergence speed and solution quality [9-10]. The authors have clearly indicated PSO to be a better performer. PSO with quantum infusion for the design of Digital Filters. Particle Swarm Optimization (PSO) is an evolutionary algorithm developed by Eberhart *et al.* [11- 12]. Several attempts have been made towards the optimization of the FIR Filter [10] using PSO algorithm.

The approach detailed in this paper takes advantage of the power of the Meta-heuristic optimization technique called Cuckoo search algorithm [13]. The authors have chosen to focus on real-coefficient FIR filters, in view of their importance in engineering practice. The CSA is simple to implement and its convergence may be controlled via few parameters. Common filtering objectives are to improve the quality of a signal (for example, to remove or reduce noise), to extract information from the signals. The advantage of using Cuckoo Search Algorithm for coefficient calculation lies in fact that filter designed using this technique offers improved characteristics such as flat passband and higher stopband attenuation. In achieve these characteristics, the CS algorithm has been implemented in this paper and is employed for FIR low pass filter design.

This paper describes an technique for the FIR low pass digital filter design using Cuckoo Search algorithm (CSA). CSA algorithm tries to find the best coefficients that closely match the ideal frequency response. Based upon the CSA approach, this paper presents a good and comprehensive set of results, and states arguments for the superiority of the algorithm. Simulation result demonstrates the effectiveness and better performance of the proposed designed method.

The rest of the paper is arranged as follows. In section II, the FIR low pass filter design problem is formulated. Section III briefly discusses on the algorithm of Cuckoo search algorithm. Section IV describes the simulation results obtained for low pass FIR digital filter using CS algorithm. Finally, section V concludes the paper.

II. LOW PASS FIR FILTER DESIGN

A digital FIR filter is characterized by:

$$\sum_{n=0}^{N} h(n)z^{-n} \text{ , n=0,1.....N} \qquad (1)$$

Where N is the order of the filter which has (N+1) number of coefficients. h(n) is the filter's impulse response. It is calculated by applying an impulse signal at the input. The values of h(n) will find the type of the filter e.g. low pass, high pass,band pass etc. The values of h(n) are to be determined in the design process and N represents the order of the polynomial function. This paper presents the most widely used FIR with h(n) as even symmetric and the order is even. The length of h(n) is N+1 and the number of coefficients is also N+1. In the algorithm, the individual represents h(n). In each iteration, these individuals are updated. Fitness of particles is calculated using the new coefficients. In each iteration, this fitness is used to improve the search and results obtained after a certain number of iterations or after the error is below a certain limit is considered to be the optimal result. Because its coefficients are symmetrical, the dimension of the problem reduces by a factor of 2. The (N+1)/2 coefficients are then flipped and concatenated to find the required N+1 coefficients. The least error is used to evaluate the fitness of the individual. It takes the error between the frequency response of the ideal and the actual filter. An ideal filter has a magnitude of one on the pass band and a magnitude of zero on the stop band. So, the error for this fitness function is the difference between the magnitudes of this filter and the filter designed using the evolutionary algorithms GA(Genetic Algorithm), PSO (Particle Swarm Optimization) and IPSO (Improved Paticle swarm Optimization). The individuals that have lower error values represent the better filter i.e., the filter with better frequency response. Various filter parameters which are responsible for the optimal filter design are the stop band and pass band normalized frequencies ($\omega_{s,}$ ω_p) the pass band and stop band ripples δp and δs , the stop band attenuation and the transition width. These parameters are mainly decided by the filter coefficients which are evident from transfer function in (1). Several scholars have investigated and developed algorithms in which *N*, δp, and δs are fixed while the remaining parameters are optimized [6]. Other algorithms were originally developed by Parks and McClellan (PM) [3] in which *N*, ω_p ,ω_s, and the ratio $\delta p/\delta s$ are fixed.

In this paper, swarm and evolutionary optimization algorithms are applied in order to obtain the actual filter response as close as possible to the ideal response. Now for (1), coefficient vector is represented in N+1 dimensions. The particles are distributed in a D dimensional search space, where D = N+1 for the case of FIR filter. The frequency response of the FIR digital filter can be calculated as:

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

Where ω_k=2πk/N ; H($e^{j\omega_k}$) is the fourier transform complex vector. This is the FIR filter frequency response. The frequency is sampled in [0,π] with N points; the positions of the particles in this D dimensional search space represent the coefficients of the transfer function. In each iteration, these particles find a new position, which is the new set of coefficients. Fitnesses of particles are calculated using the new coefficients. These fitnesses are used to improve the search in each iteration, and result obtained after a certain number of iterations or after the error is below a certain limit is considered to be the final result. Different kinds of fitness functions have been used in different literatures. An error function given by (3) is the approximate error used in Parks–McClellan algorithm for filter design [3]:

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

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

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

The major drawback of PM algorithm is that the ratio of $\delta p/\delta s$ is fixed. To improve the flexibility in the error function to be minimized, so that the desired level of δp and δs may be specified. The error function given in (5) has been considered as fitness function in many literatures [16]. The error to be minimized is defined as:

$$J_1 = \max_{\omega \le \omega_p}(|E(\omega) - \delta_p) + \max_{\omega \ge \omega_p}(|E(\omega) - \delta_s) \quad (5)$$

Where δ_s *and* δ_p are ripple in pass band and stop band; and ω_p *and* $\omega_{s,}$ are pass band and stop band normalized cut off frequencies, respectively. Equation (5) represents the fitness function to be minimized using the evolutionary algorithms. The algorithms try to minimize this error and thus increase the fitness. Since the coefficients of the linear phase filter are matched, meaning the first and the last coefficients are the same; the dimension of the problem is reduced by one-half. By only determining one half of the coefficients, the filter could be designed. This greatly reduced the computational complexity of the algorithms.

III. EVOLUTIONARY TECHNIQUES EMPLOYED

3.1. Cuckoo Search Algorithm (CSA)

Cuckoo-inspired algorithms are population based new optimization purposes. In this section, cuckoo inspired algorithms which have been developed are discussed. These procedures are based on the parasitic behavior observed in some cuckoo species, in combination with the Levy flight behavior discovered in some birds and fruit flies. Cuckoo Search Algorithm was developed by Xin-She Yang and Suash Deb in 2009 [13]. The authors made a new meta-heuristic algorithm, known as cuckoo search (CS) for solving optimization problems.

Two important characteristics are selection of the fittest and adaptation to the environment. Numerically speaking, these can be translated into two crucial characteristics of the modern metaheuristics: intensification and diversification [13],[14]. Intensification intends to search around the current best solutions and select the best candidates or solutions, while diversification makes sure the algorithm can explore the search space efficiently. Many problems are continuous in the real world and finding the solutions is difficult. Heuristic methods have been tackling the problems within reasonable computational time. Heuristic methods give an approximate solution. The later studies of the researchers have led to development of the algorithms which have been based on the natural phenomenon. Several Meta-Heuristic algorithms have been developed during recent decades. For solving the optimization problems, the Meta-Heuristic methods have been efficient in finding the solution.

Cuckoo Optimization Algorithm is based on the life of a bird called cuckoo [13],[14]. It was followed by the obligate brood parasitism of some cuckoo species by laying their eggs in the nests of other host birds (of other species). Some host birds can engage direct conflict with the intruding cuckoos. For example, if a host bird discovers the eggs are not their own, it will either throw these alien eggs away or simply abandon its nest and build a new nest elsewhere. Some cuckoo species such as New World brood-parasitic Tapera have evolved in such a way that female parasitic cuckoos are often very specialized in the mimicry in colors and pattern of the eggs of a few chosen host species. Cuckoo search idealized such breeding behaviour, and thus can be applied for various optimization problems. It seems that it can outperform other metaheuristic algorithms in applications.

Cuckoo search (CS) uses the following representations [13]:
Each egg in a nest represents a solution, and a cuckoo egg represents a new solution. The aim is to use the new and potentially better solutions (cuckoos) to replace a not-so-good solution in the nests. In the simplest form, each nest has one egg. The algorithm can be extended to more complicated cases in which each nest has multiple eggs representing a set of solutions
CS is based on three main rules:

1. Each cuckoo lays one egg at a time, and dumps its egg in a randomly chosen nest;
2. The best nests with high quality of eggs will carry over to the next generation;
3. The number of available hosts nests is fixed, and the egg laid by a cuckoo is discovered by the host bird with a probability.

On the other hand, various studies have shown that flight behaviour of many animals and insects has demonstrated the typical characteristics of Levy flights [13],[14]. In nature, animals search for food in a random or quasi-random manner. In general, the foraging path of an animal is effectively a random walk because the next move is based on the current location/state and the transition probability to the next location. Which direction it chooses depends implicitly on a probability which can be modeled mathematically. For example, various studies have shown that the flight behaviour of many animals and insects has demonstrated the typical characteristics of Levy flight. Even light can be related to levy flight [15].

A recent study shows that fruit flies or Drosophila melanogaster, explore their landscape using a series of straight flight paths punctuated by a sudden 90 degree turn, leading to a Levy-flight-style intermittent scale free search pattern[14].

When generating new solutions $x^{(t+1)}$ for, say cuckoo i, a Levy flight is performed[13],[14]:

$$x_i^{(t+1)}=x_i^{(t)} + \alpha \oplus \text{Levy}\ (\lambda) \tag{6}$$

where $\alpha > 0$ is the step size which should be related to the scales of the problem of interest. In most cases, the product means entry-wise multiplications. Levy flights essentially provide a random walk while their random steps are drawn from a Levy distribution for large steps:

$$\text{Levy} \sim \text{u} = t^{-\lambda} \quad (1 < \lambda \leq 3) \tag{7}$$

which has an infinite variance with an infinite mean. Here the consecutive jumps/steps of a cuckoo essentially form a random walk process which obeys a power-law step-length distribution with a heavy tail. It is worth pointing out that, in the real world, if a cuckoo's egg is very similar to a host's eggs, then this cuckoo's egg is less likely to be discovered, thus the fitness should be related to the difference in solutions. Therefore, it is a good idea to do a random walk in a biased way with some random step sizes.

The operation of Cuckoo Search algorithm depends on the pseudo code. Pseudo code is an informal high-level description of the principle of algorithm. This code uses the structural conventions of a programming language that are helpful for human reading The purpose of using pseudo code is that it is easier for people to understand than conventional programming language code and it is an efficient and environment-independent description of the key principles of an algorithm. The Pseudo code for Cuckoo search operation can be summarized as:

begin
Objective function f(x), $x = (x_1 \ldots\ldots\ldots.. x_d)^T$
Generate Initial Population of
n host nests x_i ***(i= 1,2,3……..n)***
while (t < MaxGeneration) or (Stop Criterion)
Get a Cuckoo randomly by Levy Flights
Evaluate its quality/fitness F_i
Choose a nest among n, (say j) randomly
If $(F_i > F_j)$,
Replace j by the new solutions ;
end
A fraction (p_a ***) of worst nests***
are abandoned and new nests are built ;
Keep the best solutions
(or nest with quality solutions) ;
Rank the solutions and find the current best
end while
Postprocessed results and Visualization
end

It show the operation for cuckoo search algorithm which contains following steps:

1. Define optimization parameters, fitness function and population size.
2. Generate the initial population of host nests.
3. Evaluate the fitness of all solutions and identify the best solutions.
4. Generate new solution by levy flights (6): $x_i^{(t+1)}=x_i^{(t)}+\alpha\oplus\text{Levy}\ (\lambda)$
5. Evaluate the fitness value for new solutions.
6. Replace a fraction of bad solutions with good solutions.
7. Compare the fitness value and keep the best nest with best quality solutions.
8. Determine new alien nests with probability and local step size.
9. Evaluate the fitness value for new alien solutions.
10. Compare the fitness value for each new alien nest and best nest.
11. If termination required is yes, then it means best solutions are found and if termination required is

no, then again for new iteration for finding the best solution.

IV. RESULTS AND DISCUSSIONS

4.1 Analysis of Magnitude Response of Low Pass FIR Filters

In order to demonstrate effectiveness of proposed filter are constructed using CSA. The MATLAB simulation has been performed realize the low pass filter of order of 20. The Specifications for Low pass filter are taken from Mandal et al. in 2011 [16] are taken as reference.

The parameter of the filter that has been derived are pass band ripple (δ_p)=0.1, stop band rippl (δ_s) =0.01 for low pass filter , pass band edge frequency(ω_p)=0.25, stop band edge(ω_s)= 0.35 and transition width=0.1. The filter is designed using the objective function J1 (5) using cuckoo search algorithm. The numbers of cuckoos are taken as 30 and number of iterations are 1000. The MATLAB simulation has been performed to realize the low-pass FIR filter of length 20.The sampling frequency has been chosen as f_s = 1Hz. For the designing of low-pass filter the response of the actual filter is matched at 128 frequency points and it works well for any set of arbitrary 512 frequency points in the range from 0 to f_s/2.

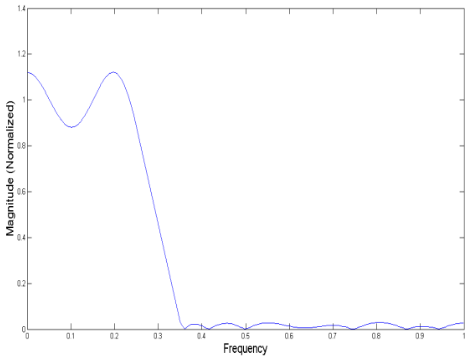

Figure 1. Magnitude (Normalized) Plot of the FIR Low Pass Filter of Order 20

Fig.1 shows the Magnitude (Normalized) response plot of low pass filter. It shows the graph between magnitude in absolute form and frequency in xπ radian/samples.

Table 1. Different values obtained using CSA

Pass band Ripple (Normalized)	Stop band Ripple (Normalized)	Transition Bandwidth
0.8808	0.02937	0.1

Fig. 2 shows the Magnitude (dB) response plot of low pass filter. It shows the graph between magnitude in decibel form and frequency in x π radian/samples.

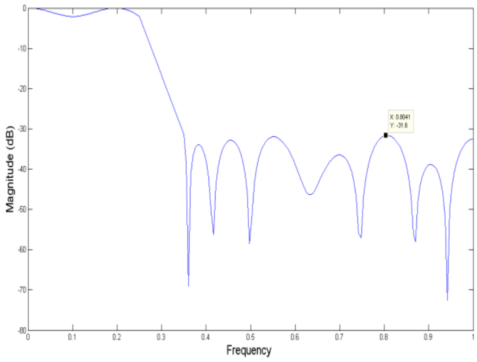

Figure 2. Magnitude (dB) Plot of the FIR Low Pass Filter of Order 20

From Fig. 2, the various different values obtained such as passband ripple, stopband attenuation, transition width and convergence error or error to be minimized are described in the following table:

Table 2. Different values obtained using CSA

Convergence error	Pass band Ripple (dB)	Stop band Attenuation (dB)	Transition Bandwidth
0.415	-2.086	-31.6	0.1

From the Table 2, it is inferred that using CS algorithm the value of pass band ripple and stop band attenuation are obtained as -2.086 dB and -31.6 dB respectively in dB. The convergence error or error to be minimized comes out as 0.0415 and transition bandwidth as 0.1.

4.2 Effectiveness and convergence profiles

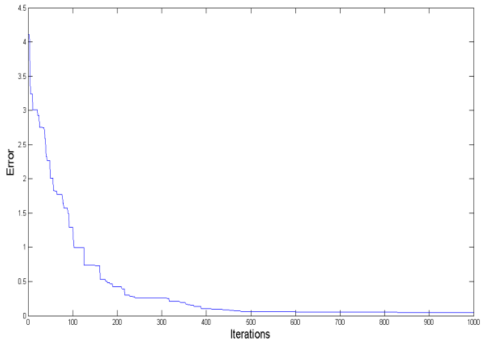

Figure 3. Convergence Profile for CSA in case of 20th order low Pass FIR Filters.

Fig. 3 shows the plot of minimum error value against the number of iteration cycles (1000) when the CSA is employed. The convergence profile has been shown for the filter order of 20. It shows that using cuckoo search algorithm, error reduces when the number of iterations increases. At one point, error value remains constant with respect to their iterations.

4.3 Best Coefficients for Filter Design

By using CS algorithm, there are various best nests obtained which are taken as coefficients for LPF.. These are the best solutions obtained in design of LPF by using CS algorithm method. The various best coefficients btained using Cuckoo search algorithm are as follows:

Table 3. Best Coefficients for filter design

Coefficients	CS algorithm
h(1) = h(21)	0.0234
h(2) = h(20)	0.0348
h(3) = h(19)	0.0357
h(4) = h(18)	0.0057
h(5) = h(17)	-0.0344
h(6) = h(16)	-0.0546
h(7) = h(15)	-0.0386
h(8) = h(14)	0.0364
h(9) = h(13)	0.1542
h(10) = h(12)	0.2507
h(11)	0.2918

The above are the coefficients for low pass filter by using cuckoo search algorithm. The values of these coefficients vary from different values of iterations.

V. CONCLUSIONS

This paper presents a novel and accurate method for designing linear phase digital low pass FIR filters. Filter of order 20 has been realized using CSA. Extensive simulation results justify that the proposed algorithm provide better results. The advantage of using Cuckoo Search Algorithm for coefficient calculation lies in fact that filter designed using this technique offers improved characteristics such as flat passband and higher stopband attenuation. and is adequate for use in other related design problems. By using this algorithm, we can also design high pass filter and adaptive filters.

REFERENCES

[1.] L. Litwin, *FIR and IIR digital filters*, *IEEE Potentials. 0278-6648*,2000, 28–31.
[2.] T.W. Parks, C.S. Burrus, *Digital Filter Design, Wiley*, New York, 1987.
[3.] T.W. Parks, J.H. McClellan, *Chebyshev approximation for non recursive digital filters with linear phase*, *IEEE Trans Circuit Theory, CT-19* (1972) 189–194.
[4.] J.H. McClellan, T.W. Parks, L.R. Rabiner, *A computer program for designing optimum FIR linear phase digital filters*, *IEEE Trans.Audio Electro acoust.*, *AU-21* (1973) 506–526.
[5.] L.R. Rabiner, *Approximate design relationships for low-pass FIR digital filters*, *IEEE Trans. Audio Electro acoust.*, *AU-21* (1973) 456–460.
[6.] O. Herrmann, W. Schussler, *Design of non-recursive digital filters with linear phase*, *Electron. Lett.*, *6* (1970), 329–330.
[7.] 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) 634–639.
[8.] S. Chen, *IIR Model Identification Using Batch-Recursive Adaptive Simulated Annealing Algorithm, In Proceedings of 6th Annual Chinese Automation and Computer Science Conference*, 2000,pp.151–155.
[9.] B. Luitel, G.K. Venayagamoorthy, *Differential Evolution Particle Swarm Optimization for Digital Filter Design*, 2008 *IEEE Congress on Evolutionary Computation* (CEC 2008), PP. 3954-3961, 2008.
[10.] J.I. Ababneh, M. H. Bataineh, *Linear phase FIR filter design using particle swarm optimization and genetic algorithms*, *Journal of Digital Signal Processing, 18,* 657–668, 2008.
[11.] J. Kennedy, R. Eberhart, *Particle Swarm Optimization*, in *Proc. IEEE int. Conf. On Neural Network*, 1995.
[12.] R. Eberhart, Y. Shi, *Comparison between Genetic Algorithms and Particle Swarm Optimization*, *Proc. 7th Ann. Conf. on Evolutionary Computation, San Diego*, 2000.
[13.] X.S. Yang and S. Deb ,*Cuckoo Search via Levy Flights*”, *Proceedings of IEEE World Conference on Nature and Biologically Inspired Computing*, pp. 210–214,2009.
[14.] X.S. Yang and S. Deb, *Engineering Optimization by Cuckoo Search*, Proceedings of *Int. J. Mathematical Modelling and Numerical Optimization , Vol. 1, 4* , pp. 330–343,2010.
[15.] P. Barthelemy,J.Bertolotty and D.S. Wiersma, *A Levy flight for light*, *Proceedings of nature Publishing Group, vol.453, 4*, pp. 495–498,2008.
[16.] S.P. Ghoshal, R. Kar, S. Mandal, D. Mandal and S. Mukherjee, *Linear Phase Low Pass FIR Filter Design using Improved Particle Swarm Optimization*, *Proceedings of IEEE Student Conference on Research and Development*, pp. 358–363, 2011.