ISSN No.: 2455-6130
Volume 1, Issue 11, 2016, pp. 23-32

website: www.ijoetr.com

International Journal of Emerging Trends in Research

Optimal Tuning of FIR Filter Design

Saroj[1], **Parveen Khanchi**[2]

[1,2]*Rayat Bahara Royal Institute of Management and Technology , Chidana, Sonipat, Haryana, India*

Abstract

The Finite Impulse Response (FIR) filter is a digital filter widely used in Digital Signal Processing applications in various fields like imaging, instrumentation, communications, etc. Programmable digital processors signal (PDSPs) can be used in implementing the FIR filter. However, in realizing a large-order filter many complex computations are needed which affects the performance of the common digital signal processors in terms of speed, cost, flexibility, etc.

Keywords: : Impulse Response:; PDSPs; FIR filter;

1. Introduction

There are many conventional methods for the design ofa FIR filter such as window method (Kaiser, Blackmann, Hanning and Hamming etc.), frequency sampling method etc. (Parks and McClellan, 1972; Rabiner, 1973; Litwin,2000). But these methods do not allow sufficient and precise control of various frequencies like pass band and stop band cut-off frequencies and the transition width. For last few years the works have been done continuously to evolve new methods for the filter design. The most frequently used method for the design of exact linear phase FIR filter is Chebyshev approximation method based on the Remez exchange algorithm developed by Parks McClellan (PM)(Parks and McClellan, 1972). A well-defined computer programme reported in McClellan et al. (1973) shows the further improvements in their results. These filter design methods are based on classical optimisation techniques and have greater tendency to get struck at local minima as they are highly dependent on their starting solutions. Slow convergence speed also limits the usefulness of the classical optimisation techniques. The objective function for the design of optimal digital filters involves accurate control of various parameters of frequency spectrum and is highly on-uniform, non-linear, non-differentiable and multimodal in nature. Several disadvantages of classical optimisation techniques are:

- *Highly sensitive to starting points when the number of solution variables and hence the size of the solution space increases*

- *Frequent convergence to local optimum solution or divergence or revisiting the same suboptimal solution*
- *Requirement of continuous and differentiable objective cost function (gradient search methods)*
- *Requirement of the piecewise linear cost approximation(linear programming)*
- *Problem of convergence and algorithm complexity(non-linear programming).*

1.1 Digital Filter Types

There are two basic types of digital filters, Finite Impulse Response (FIR) and Infinite Impulse Response (IIR) filters. The general form of the digital filter difference equation is

$$y(n) = \sum_{i=0}^{N} a_i x(n-i) - \sum_{i=0}^{N} b_i y(n-i) \tag{1}$$

where, *y(n)* is the current filter output, the $y(n-i)$ are previous filter outputs, the $x(n-i)$ are current or previous filter inputs, the $a_i$are the filter's feed forward coefficients corresponding to the zeros of the filter, the $b_i$are the filter's feedback coefficients corresponding to the poles of the filter, and N is the filter's order.

IIR filters have one or more nonzero feedback coefficients. That is, as a result of the feedback term, if the filter has one or more poles, once the filter has been excited with an impulse there is always an output. FIR filters have no non-zero feedback coefficient. That is, the filter has only zeros, and once it has been excited with an impulse, the output is present for only a finite (N) number of computational cycles [3].Because an IIR filter uses both a feed-forward polynomial (zeros as the roots) and a feedback polynomial (poles as the roots), it has a much sharper transition characteristic for a given filter order. Like analog filters with poles, an IIR filter usually has nonlinear phase characteristics. Also, the feedback loop makes IIR filters difficult to use in adaptive filter applications.

A digital filter is characterized in terms of difference equations .There are two types of digital filters, they are non-recursive, and recursive filters which are characterized based on their responses [1].

The response of a non-recursive filter at any instant depends on the present, past and future values of the input. At any specific instant *nT*. The response is of the form

$$y(nT) = f(\ldots.., x(nT-T), x(nT), x(nT+T) \ldots..) \tag{2}$$

Assuming linearity and time-invariance *y(nT)* can be expressed as

$$y(nT) = \sum_{i=-\infty}^{\infty} a_i x(nT-iT) \tag{3}$$

where .$a_i$represents constants. Now assuming causality for the filter we have

$a_{-1}= a_{-2} =\ldots\ldots\ldots.=0$

In addition, assuming a_i=0 for i> N the response can be written as Nth-order linear difference equation given as:

$$y(nT) = \sum_{i=0}^{N} a_i x(nT-iT) \tag{4}$$

Such a linear, time-invariant, causal, non-recursive filter represented as Nth-order linear difference equation is called the Finite Impulse Response (FIR) filter.

When a unit impulse defined as

$$\delta(nT) = \begin{cases} 0 \; for \; n \neq 0 \\ 1 \; for \; n = 0 \end{cases}$$

is applied to the system described by Equation (1.4), then the response, which is nothing but the impulse response *h(nT)* is given as

$$y(nT) = \sum_{i=0}^{N} a_i \delta(nT - iT) \tag{5}$$

From the above equation it can be inferred that the impulse response is finite and from the property of the impulse function we can see that the constants .a_i are nothing but the samples of the impulse response. That means

$$h(0) = a_0, h(T) = a_1 \ldots\ldots\ldots\ldots h(nT) = a_n \tag{6}$$

From these set of difference equations we can construct a block diagram consisting of an interconnection including delay elements, multipliers, and adders. Such a block diagram can be further analyzed in terms of signal flow diagrams. Such a block diagram can be referred as a *realization* of the system or in other words as a *structure* for realizing the system. These structures are nothing but the filter structures. One of the limitations of the FIR filter is that the order of the filter is generally large in order to meet the desired specifications of the filter. As the filter order is increased, the computational complexity is more which may limit the frequency of operation.

Traditionally, a DSP algorithms are implemented either using general purpose DSP processors (low speed, less expensive, flexible) or using Application Specific Integrated Circuits (ASIC) which offer high speed but are expensive and less flexible.

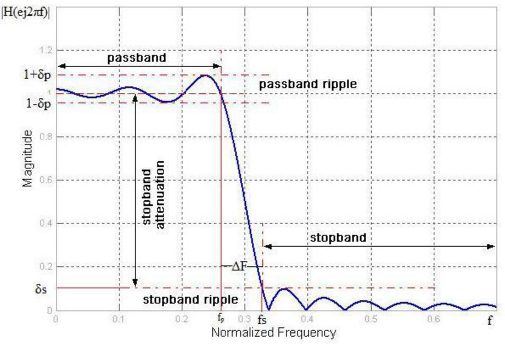

Figure 1: Magnitude frequency response specifications for a low pass filter.

The following are the key parameters of interest:

δ_p = *peak pass band deviation (or ripples)*
δ_s = *stop band deviation.*
fs = stop band edge frequency.
Fp = pass band edge frequency.
Fs = sampling frequency.

The edge frequencies are often given n the normalized form, that is as the fraction of the sampling frequency (f/Fs). Pass band and stop band deviation may be expressed in decibels. When they specify the pass band ripples and minimum stop band attenuation respectively. Thus the minimum stop band attenuation, As and the peak pass band ripple, Ap, in decibels are given as

As (stop band attenuation) = -20log10 δ_s

Ap (pass band ripple) = 20 log10 (1+δ_p)

The difference between fs and fp gives the transition width of the filter. Another important parameter is the filter length, N, which defines the number of filter.

2. Proposed Solution

In our work we are designing digital FIR filter by frequency sampling method. The frequency sampling method will work in following way, we start in the frequency domain, and sample the desired frequency response H($e^{j\Omega}$) with N evenly-spaced samples instead of a continuous frequency, and get Hd(k)= Hd($e^{j\Omega}$)|Ω=2πk/N, (k=0,1,…, N-1). Then, let H(k)= Hd(k)= Hd($e^{j\Omega}$)|Ω=2πk/N, we get the unit impulse response, h(n)=IDFT[H(k)], where IDFT is Inverse Discrete Fourier Transform. The inverse DFT then yields an impulse response which will lead to a filter whose frequency response the same as that of the specification exactly at the location of the frequency samples.

Here optimization of filter coefficients is used to design the low pass filter. Bio inspired optimization serves our purpose. Bacterial foraging optimization has been used earlier but backed by drawbacks of slow convergence. So improvement in BFO is brought by addition of particle swarm optimization (PSO). This BPSO optimization calls an objective function each time. This objective function is the backbone of whole algorithm. It calculates the error between designed filters frequency response and ideal filter response and optimization tends to minimize that error. Many constraints are put in objective function to reach the destination. The number of variables to optimize by BPSO is the order of filter divided by 2, if it is even else order +1 divided by 2. This is because digital filter coefficients exhibits a property of similarity of $[h(1) = h(N), h(2) = h(N-1) \dots \dots \dots]$. As per the property of BFO and PSO, initial positions of bacteria are random, but we have to fix some of them to give the control of defined pass band frequency. Various filter parameters which are responsible for the optimal filter design are the stop band and pass band normalized frequencies (ω_s, ω_p) the pass band and stop band ripples $(\delta_{s,}\delta_p)$, the stop band attenuation and the transition width. These parameters are mainly decided by the filter coefficients which are evident from transfer function in equation 1.5. Different kind of error fitness function calculation is used in literature but in our work following error calculation is used:

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

Where $H_d(e^{j\omega})$ is designed filter response and $H_i(e^{j\omega})]$ is ideal filter response, $G(\omega)$ is the weighting factor. For ideal low pass filter

$$H_i(e^{j\omega}) = \begin{cases} 1\ for\ 0 \le \omega \le \omega_c \\ 0 \quad otherwise \end{cases}$$

The major drawback of this algorithm is that $\frac{\delta_p}{\delta_{s,}}$ is fixed. To give control of user over pass band and stop band ripples the error function is updated as:

$$J = \max_{\omega \le \omega_p}[|E(\omega) - \delta_p|] + \max_{\omega \ge \omega_s}[|E(\omega) - \delta_s|]$$

This equation provides the constraint to the objective function for minimization of error. This error is compared to previous error after every iteration and if condition is met then optimization stops. The complete flow chart of the method step by step is shown in the appendix-A

3. Results

In our work as discussed above FIR filters are designed by frequency sampling method. Frequency sampling method gives more freedom to control the undesired ripples in pass band and stop band. It is a bitter truth to accept that ideal filter response is impossible to design, but it is always tried to minimize the ripples in stop band. Our work is also a step forward in that direction. We have chosen bio inspired optimization to set frequency response of filter so that ripples can be minimized. Here particle swarm optimization (PSO) in collaboration with bacterial foraging optimization (BFO) is used and results are compared with only BFO and practical response of filter. For this purpose MATLAB is used as a tool as it provides a wide range of toolboxes and freedom to design desired filter. The frequencies selected here for stop band and pass band are normalized. The order of filter is set to M=21. Since filter coefficients shows a similar pattern so we need to calculate only 11 coefficients here as order is odd, if order is to be even then calculation points will be M/2. This reduces the computation time. 1000 sampling points have been selected. To reduce the side lobs in the filter, two methods can be adopted: either increase the samples or set the coefficients value to optimum value. Increasing the sampling points still not gives control over filter coefficients and increases the computation time also. So, later is selected for our work. Initially BFO is used for computation and sets the coefficient value. In BFO as discussed above an objective function is called in each iterations and that objective function measure the error after calculating by the formula discussed in chapter 4. Initial values considered for filter, BFO and BPSO optimization is listed in table 1.

Table 1: Initial Parameters considered for filter design

Order of filter	21
Pass band frequency	0.5*pi
Transition band frequency	0.1*pi
Pass band ripples	0.1
Stop band ripples	0.01
Sampling points	1000
BFO Parameters	
Number of bacteria	26
Dimension of searching space	11
No of chemo tactic step	5
Swimming steps	15
Reproduction steps	4
Elimination/dispersal steps	2
Step size	0.5
PSO Parameters	
C1	0.5
C2	0.05
R1& R2	random

Initial positions of bacteria in both optimizations is taken random conditionally the positions for which pass band frequency is greater than the sampling points frequency, are set to 1 in every iteration so that control over cut off frequency can be provided during optimization. In BFO each bacteria move to gain optimum position which has been animated in our script and final positions attained by bacteria are shown in figure 2.

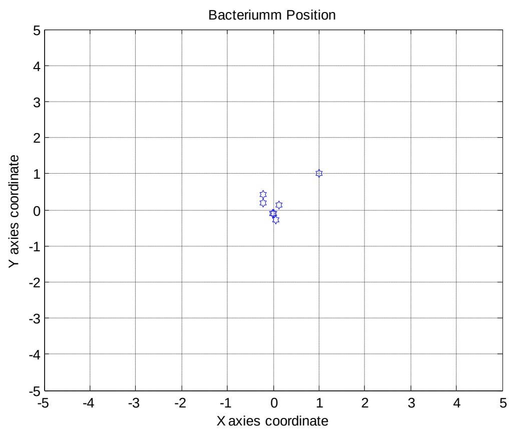

Figure2: bacteria final position after BFO optimization

Since random positions are used every time as per the BFO algorithm, results may be different each time script is executed. Here results pasted are after execution of 20 times.Any optimization works best if the error function or fitness function decreases after iterations. Decreasing fitness function indicates the error is reducing which automatically enhances results. Since ideal filter response is 1 for pass band and 0 for stop band, so decrease in error proves results are approaching the ideal values. The fitness value in case of BFO is shown in figure 3 below. Which shows the decrease in error value and improvement in results.The output of BFO is multiplied with exponential function, which gives the impulse response of filter. DFT is applied to outcome and real part of it is considered as filter coefficients. Frequency response is obtained for these filter coefficients.

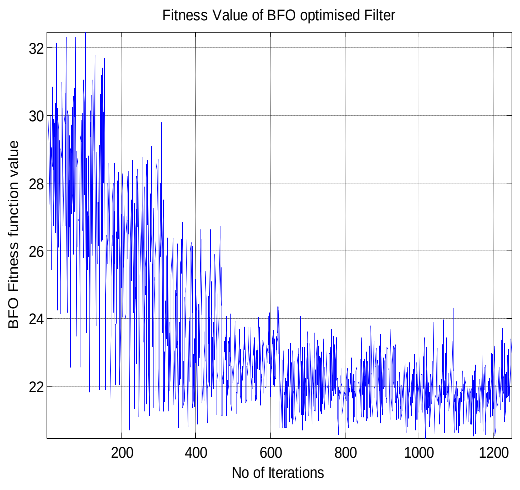

Figure 3: Fitness value graph of BFO optimization

This creates a low pass filter. To check the feasibility of this filter a signal of three different frequencies of 100, 300 & 700 Hz is constructed and sampling frequency of that signal is taken as 2000 Hz. Since we have taken normalized pass band frequency as 0.5*pi (500 Hz), so signal with frequency 700 Hz is undesired for our filter and designed filter should suppress that frequency.

Figure 4 shows the combined three frequency’s signal in time domain and frequency domain. For conversion of signal to frequency domain, fourier transform is used.

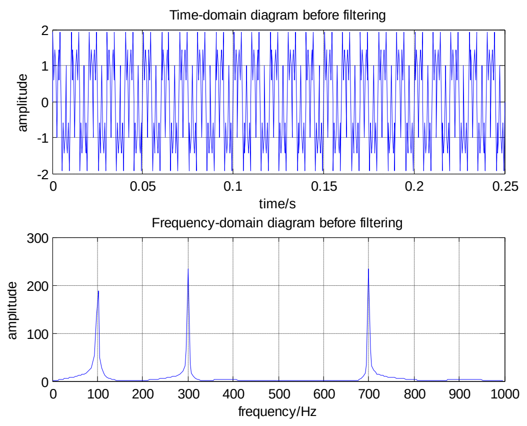

Figure 4: Test signal in time and frequency domain before filtering

We have mixed the signal with three frequency signal of 100, 300 and 700 Hz, which is clearly visible in frequency domain signal with main lobes around these central frequencies and time domain signal is chirped rather than sinusoidal shape because of mixing. If this signal is passed through filter designed by BFO then outcome is shown in figure 5. This figure shows BFO filter stop the signal of 700 Hz frequency and signal in time domain is also in somewhat sinusoidal shape. BFO filter gives good response as it kept the lobes of other frequency signals unchanged.

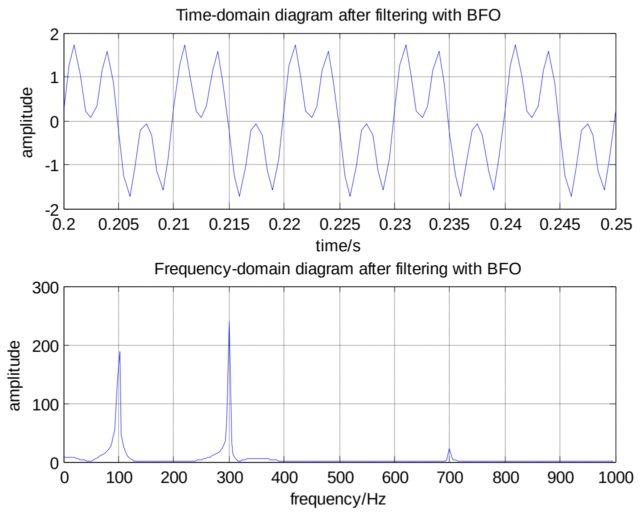

Figure 5: Filtered signal by BFO designed low pass filter

The frequency response of BFO filter is shown in figure 5 which is compared with proposed optimization technique also. Here BFO filter response shows that ripples are very much less in pass band and filter response changes at pass band frequency and a transition band is visible till stop band frequency. This filter exactly follows the pass band frequency and stop band frequency limit. Side lobs are also of less magnitude than practical filter's side lobs. The filter coefficients values are shown in table 3 along with PBFO optimized coefficients.

Figure6 clearly shows that although side lobs are reduced by BFO but still it is very far away from ideal response. So PSO is added to BFO optimization to improve results.

In PBFO, PSO is used to update the random directions of bacteria. In this also efficiency of algorithm is shown by the decrease in the fitness function value which is shown in figure 7. Number of iterations in both the cases is kept same and BFO parameters are considered as given in table1. Figure 6 shows that fitness function is settled to a more minimum value than BFO optimization which is also reflected in frequency response graph o PBFO, shown in figure 6. Side lobs in PBFO are more reduced than both practical and BFO designed filter. Figure 8 shows the output of filter after considering the same signal as above as input. Results show that PBFO clearly suppress the frequency signal of 700 Hz which was not in case of BFO. So PBFO performs better than BFO in every aspect.

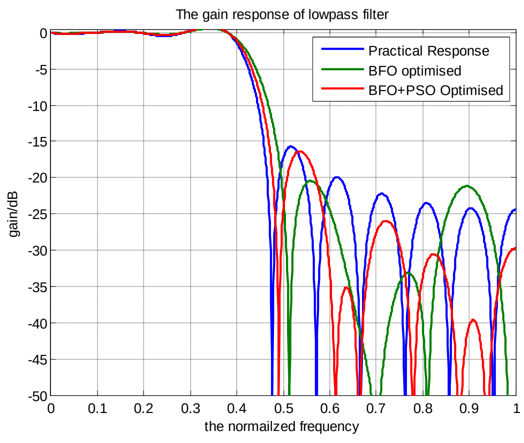

Figure 6: frequency response of practical, BFO, BPSO designed filter

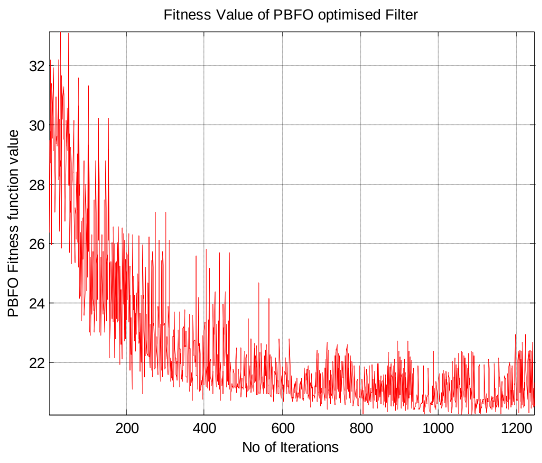

Figure 7: fitness values in case of PBFO optimization

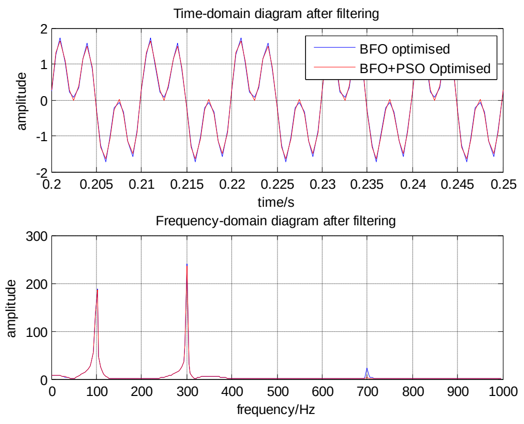

Figure 8: Filter output of BFO and PBFO designed filter

Above tablel show that alnmost for every filter coefficient magnitude is lesser in case of PBFO than BFO. A more clear demonstartion of this table is shown in figure 8 in form of bar graph.

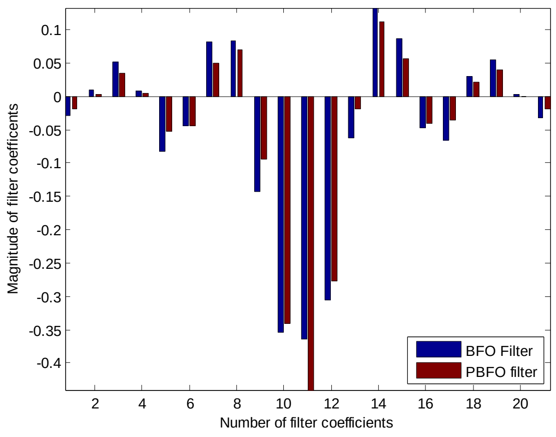

Figure 8: Bra graph comparison of BFO and PBFO filter coefficients

This algorithm is checked for other pass band frequencies for same hybrid input signal to the filter.

4. Conclusions

The designing of FIR filter by frequency sampling method gives more control over ripples suppression than window method. Frequency sampling method also gives freedom to optimize the filter coefficients to approach the ideal filter response. For this purpose bacteria foraging optimization (BFO) along with particle swarm optimization (PSO) is used and filters response was compared with only BFO designed filter which was our one of the objective. Low pass filter is designed to prove our results. Results show that frequency response by PBSO is better than BFO. Side lobs are more suppressed in PBFO than BFO and filter thus designed is checked for a signal mix of various central frequencies signals. Our designed filter performs better in case of filtration of undesired signal. Results considered here are for filter order of 21. These can be checked for more filter orders and results will be found improved for our desired filter.

References

[1]. Ouadi, A., Bentarzi, H., & Recioui, A. (2014). Optimal Multiobjective Design Of Digital Filters Using Taguchi Optimization Technique. *Journal of Electrical Engineering*, *65*(1), 21-29.

[2]. Kaur, R., & Sharma, P. Low Pass Filter Design Using Bacterial Foraging with Particle Swarm Optimization.'

[3]. Singh, B., Dhillon, J. S., & Brar, Y. S. (2013). A hybrid differential evolution method for the design of IIR digital filter. *ACEEE International Journal of Signal & Image Processing*, *4*(1), 1-10.

[4]. Saha, S. K., Dutta, R., Choudhury, R., Kar, R., Mandal, D., & Ghoshal, S. P. (2013). Efficient and accurate optimal linear phase fir filter design using opposition-based harmony search algorithm. *The Scientific World Journal*, *2013*.

[5]. Liu, X., Cai, Z., & Yu, C. (2011). A hybrid harmony search approach based on differential evolution. *Journal of Information & Computational Science, Available at http://www. joics. com, Binary Information Press*.

[6]. Kaur, H., & Dhaliwal, B. (2013). Design of Low Pass FIR Filter Using Artificial NeuralNetwork. *International Journal of Information and Electronics Engineering*, *3*(2), 204.

[7]. Ülker, E. D., & Haydar, A. (2013, September). A hybrid algorithm based on differential

evolution, particle swarm optimization and harmony search algorithms. In *Computer Science and Information Systems (FedCSIS), 2013 Federated Conference on* (pp. 417-420). IEEE.

[8]. Singh, S., Agarwal, R., & Sharma, S. (2013). Optimization of FIR Filter Usi.

[9]. Bhattacharya, A. (2013). A modified window function for FIR filter design with an improved frequency response and its comparison with the Hamming window. *International Journal of Science, Engineering and Technology Research*, *2*(5), pp-1122.

[10]. Vijay. P , M. Sudhakar V. (2012). Comparison of Various FIR Low pass Filter Design Techniques With PSO Algotirhm. International Journal of Advanced Research in Computer and Communication Engineering, 1(9).

[11]. Mondal, S., Chakraborty, D., Kar, R., Mandal, D., & Ghoshal, S. P. (2012, June). Novel particle swarm optimization for high pass FIR filter design. In 2012 IEEE Symposium on Humanities, Science and Engineering Research(pp. 413-418). IEEE.

[12]. Kar, R., Mandal, D., Roy, D., & Ghoshal, S. P. (2011). FIR Filter Design using Particle Swarm Optimization with Constriction Factor and Inertia Weight Approach 1.

[13]. Mondal, S., Vasundhara, R. K., Mandal, D., & Ghoshal, S. P. (2011). Linear phase high pass FIR filter design using improved particle swarm optimization. *World academy of science, engineering and technology*, *60*, 1621-1627.

[14]. Sharma, A., & Kaur, R. DESIGN OF OPTIMUM LINEAR PHASE LOW PASS FIR FILTER USING HYBRID PSO AND GSA EVOLUTIONARY ALGORITHM.

[15]. Kaur, P., & Kaur, S. (2012). Optimization of FIR Filters Design using Genetic Algorithm. *International journal of Emerging Trends& technology in computer science (IJETTSC)*, *1*(3), 228-232.

[16]. Saha, S. K., Kar, R., Mandal, D., & Ghoshal, S. P. (2013). Bacteria foraging optimisation algorithm for optimal FIR filter design. International Journal of Bio-Inspired Computation, 5(1), 52-66.

[17]. Singh M. & Khanchi P. (2016). APR in LTE OFDM: A Survey. International Journal of Emerging Trends in research, 1(9), 11-16.

[18]. Das, S., Biswas, A., Dasgupta, S., & Abraham, A. (2009). Bacterial foraging optimization algorithm: theoretical foundations, analysis, and applications. In Foundations of Computational Intelligence Volume 3 (pp. 23-55). Springer Berlin Heidelberg.

[19]. Pedersen, M. E. H. (2010). Good parameters for particle swarm optimization. Hvass Lab., Copenhagen, Denmark, Tech. Rep. HL1001.

[20]. Rini, D. P., Shamsuddin, S. M., & Yuhaniz, S. S. (2011). Particle swarm optimization: technique, system and challenges. International Journal of Computer Applications, 14(1), 19-26.

[21]. Del Valle, Y., Venayagamoorthy, G. K., Mohagheghi, S., Hernandez, J. C., & Harley, R. G. (2008). Particle swarm optimization: basic concepts, variants and applications in power systems. IEEE Transactions on evolutionary computation, 12(2), 171-195.