Design of Linear-Phase Digital FIR Filter Using Differential Evolution Optimization with an Improved Ripple Constraint Handling Method

Shekhar Sharma[1], Sumant Katiyal[2] and **L. D. Arya[3]**

Submitted in April, 2014; Accepted in December, 2015

Abstract - For the optimal design of frequency-selective digital filters, evolutionary optimization algorithms have been applied. In these design methods the goal of the optimization process is to find the optimal filter coefficients which closely approximate the desired frequency response. In this paper, an efficient alternative method for the design of linear phase digital FIR filter with ripple constraint is discussed. This method of optimization uses DE algorithm with modified selection rule for ripple constraint handling. The results obtained using this method are compared with those obtained for another method of ripple constraint handling based on penalty function using DE algorithm. From the simulation results it is observed that ripple constraint handling method based on the modified selection rule of DE shows better performance than that obtained using DE with ripple constraint method based on penalty function when number of runs is applied.

Index Terms — Differential Evolution algorithm, FIR filters, Frequency response, Ripple constraint.

NOMENCLATURE

$H(e^{j\omega})$: The frequency response of a digital filter,
$h(n)$: The impulse response of a digital filter,
$A(\omega)$: The magnitude response,
$\theta(\omega)$: The phase response,
$a(k)$: The filter coefficients,
$E(\omega)$: The approximation error function,
$W(\omega)$: The weighting function,
$D(\omega)$: The desired frequency response,
$L_2(\vec{X})$: The discrete form of L_2 norm approximation error,
$\vec{X}_k$: The k^{th} solution vector ,
x_{kj} : The j^{th} component of the solution vector,
F : The scaling factor,
CR : The crossover factor.

[1]*Dept. of Electronics & Telecommunication Engg., S.G.S.I.T.S., Indore, Madhya Pradesh*
[2]*School of Electronics, Devi Ahilya Vishvavidyalaya, Indore, Madhya Pradesh*
[3]*Dept. of Electrical Engg., S.G.S.I.T.S., Indore, Madhya Pradesh*
E-mail: [1]shekhar.sgsits@gmail.com, [2]sumant578@yahoo.com and [3]ldarya@rediffmail.com

1.0 INTRODUCTION

Digital filters are recognized by great flexibility in design and implementation. This makes it is easier to implement complex signal processing schemes utilized in digital communication systems. The digital filters are classified according to the length of the impulse response as Finite Impulse Response (FIR) and Infinite Impulse Response (IIR) digital filters [1]. Special features of the FIR filters are their inherent stability and precise linear phase. The phase response of the FIR filter is linear if the coefficients of FIR are symmetric around the centre coefficient. Linear phase FIR filters have many applications such as in telecommunications, for demultiplexing the data that have been frequency-division multiplexed, without distorting the data in this process, and in systems, where it is necessary to have minimum signal distortion and signal dispersion so as to avoid inter symbol interference.

In order to seek better control over different parameters in the design of digital filters, the design methods based on optimization algorithms are developed. Thus, Evolutionary algorithms (EA), such as Genetic algorithm (GA), Particle swarm optimization (PSO), and Differential evolution (DE) and many others have been used for better individual control over the parameters of digital filters. Generally, in these design methods, the unconstrained optimization is used. In this paper, DE algorithm is used for the design of linear phase digital FIR filter with two ripple constraint handling methods. One method is based on penalty function [2] and the other is based on a method proposed by Lampinen [3]. The simulation results obtained for these two methods show that when number of runs is applied, the ripples obtained in different frequency bands using the second method have smaller ripple size and smaller value of error as compared to those obtained for the first method. Thus a better performance is exhibited by the second method.

2.0 LITERATURE SURVEY

For the design of the digital FIR filters two classical methods are used namely, windowing method and frequency sampling method [4]. In general, an approximation error norm is used in these methods for designing an FIR filter. The two most commonly used norms are the least-squares (L_2) norm and Chebyshev (L_∞) norm.

In the design method based on windowing, the decrease in the transition bandwidth causes increase in the magnitude of the side lobes and consequently an increase in the approximation error. Also this method does not offer individual control over the approximation errors in different bands with any constraint criterion. The design method based on frequency sampling, provides good control over the transition bandwidth; however, the approximation error is zero exactly at the sampling frequencies.

Recently, the design methods based on optimization algorithms are developed. Genetic Algorithm (GA) is one of the most useful, general purpose optimization algorithm [5]. GA has been used to solve a wide range of engineering design and testing optimization problems such as ATM network design [6], optimal testing of nonlinear allocation problems in modular softwares [7], etc. GA has also been applied for the design of digital FIR filters by Xu and Daley [8], Cen, [9] and others. Although GA is a good global searching algorithm, sometimes it gets trapped into the local minima, and is complex in coding. Also GA has slower convergence and takes more execution time.

Particle swarm optimization (PSO) algorithm developed by Kennedy and Eberhart [10] requires less parameters and is simple. PSO has been used for various engineering optimization problems such as, in malicious node detection and path optimization for wireless sensor networks [11]. PSO and its variants have also been applied for the design of digital filters [12]-[15]. It gives faster convergence as compared to GA as shown by Ababneh [12]. The modified PSO is applied by Sharma and Arya [13] for the design of linear phase digital FIR filter to control global exploration and local exploration.

Another optimization algorithm, Differential evolution (DE), developed by Storn and Price [16] has been used in power systems for optimization in planning, operation and distribution etc. [17]. DE has also been used for the design of digital FIR filters by Zhao and Meng [18], Albataineh et al. [19], Singh and Kaur [20], Sharma et al. [2] and others. DE algorithms provide good global optimization if its control parameters are adjusted properly. DE algorithm, with two ripple constraint handling methods, is used in this paper for the design of linear phase digital FIR filter. One method is based on penalty function [2], and another is based on the method proposed by Lampinen [3]. The comparison of the simulation results obtained for these two methods shows that the maximum error magnitude between desired frequency response and the designed frequency response is equal to or below the constraint in the specified frequency bands. However, when number of runs is applied, the ripples obtained in different frequency bands for the first method have almost constant magnitude; while the ripple size and final error value is less in the case of second method and thus a better performance is exhibited by this method.

This paper is organized in six sections as follows: In Section 3, Problem formulation of linear phase digital FIR filters is presented. Section 4 explains the DE algorithm for optimizing filter coefficients and describes the ripple constraint handling methods. In Section 5, design of linear phase FIR filters using DE with ripple constraint methods is given. Then, in Section 6, simulation results are discussed and analyzed. Finally, conclusion and future scope are discussed in Section 7.

3.0 PROBLEM FORMULATION

The frequency response of a linear-phase FIR filter is given by:

$$H(e^{j\omega}) = \sum_{n=0}^{N} h(n) e^{-j\omega n} \tag{1}$$

Where, $h(n)$ is the real-valued impulse response of filter, $(N+1)$ is the length of filter and ω is the frequency of interest. The linear phase is possible if the impulse response $h(n)$ is either symmetric {i.e. $h(n) = h(N - n)$}, or, is antisymmetric $\{h(n) = -h(N-n)\}$ for $0 \le n \le N$.

In general, for causal linear-phase FIR filters, the frequency response $H(e^{j\omega})$ can be expressed as:

$$H(e^{j\omega}) = e^{-jN\omega/2} e^{j\beta} A(\omega) \tag{2}$$

Thus, the magnitude response is:

$$\left|H(e^{j\omega})\right| = A(\omega) , \tag{3}$$

& the phase response is:

$$\theta(\omega) = \begin{cases} -N\dfrac{\omega}{2} + \beta, & for \quad A(\omega) \ge 0, \\ -N\dfrac{\omega}{2} + \beta - \pi, & for \quad A(\omega) < 0. \end{cases} \tag{4}$$

When $\beta = 0$, h(n) is symmetrical, and when $\beta = \pi/2$, $h(n)$ is antisymmetrical.

The amplitude response [1], for the case of type-I linear phase FIR filter, by substituting N = 2M, is given by:

$$A(\omega) = \sum_{k=0}^{M} a(k)\cos(\omega k), \tag{5}$$

where, $a(0) = h(M)$ and $a(k) = 2h(M-k)$, for $1 \le k \le M$.

For the design of low pass digital FIR filters, the objective of the algorithm used for computation, is to determine the vector $\vec{X}$ of coefficients $a(k)$, so as to minimize the difference between the desired frequency response, $D(\omega)$, and the realized amplitude frequency response, $A(\omega)$. Generally this difference is specified as a weighted error function $E(\omega)$ given by:

$$E(\omega) = W(\omega)[A(\omega) - D(\omega)] \tag{6}$$

Where, $W(\omega)$ is a non-negative weighting function and is accepted for the given pass band attenuation δ_p and stop band attenuation δ_s, as:

$$W(\omega) = \begin{cases} (\delta_s / \delta_p), & in \quad the \quad passband, \\ 1, & in \quad the \quad stopband. \end{cases} \tag{7}$$

And $D(\omega)$, the desired magnitude response for the low pass filter given by:

$$D(\omega) = \begin{cases} 1, & in \quad the \quad passband, \\ 0, & in \quad the \quad stopband. \end{cases} \tag{8}$$

The least-squares, or, L_2 norm; which considers error energy, is defined in the integral form [2] as:

$$\|\varepsilon\|_2 = \left(\frac{1}{\pi} \int_0^{\pi} \left| W(\omega)[A(\omega) - D(\omega)] \right|^2 d\omega \right)^{1/2} \tag{9}$$

In practice, the discretized version of integral scalar error used in L_2 norm is approximated by a finite sum given by:

$$L_2(\vec{X}) = \left(\sum_{i=1}^{K} \left| W(\omega_i)[A(\omega_i) - D(\omega_i)] \right|^2 \right)^{1/2} \quad (10)$$

Where, $L_2(\vec{X})$ is L_2 norm approximation error determined for a vector $\vec{X}$ and ω_i is suitably chosen grid of digital angular frequencies for the range $0 \le \omega \le \pi$ for $1 \le i \le K$.

4.0 DE ALGORITHM

DE algorithm introduced by Storn and Price [16] is a simple population based stochastic search algorithm for objective function minimization. Application of DE for the design of digital filters has been described in [2], [18][19][20]. In basic DE algorithm, the initial NP population vectors are formed randomly from the vectors having bounded parameter values. Each of these vectors has D-parameters and belongs to a D-dimensional vector space. The optimization task is to successively improve these vectors by applying mutation, crossover and selection operators; similar to those used by GA. DE generates new solution vectors in the D-dimensional vector space using mutation. To produce mutated vector the weighted difference between two randomly chosen, distinct population vectors, is added to another distinct vector. Then crossover is performed to produce a trial vector from target vector and mutated vector. By evaluating objective function for target vector and trial vector, either one is then selected on the basis of their fitness.

4.1 The Steps of DE Algorithm

Step 1:

Initialization: An initial population of 'NP' solution vectors is generated as follows:

$$P^0 = [\vec{X}_1^0, \vec{X}_2^0, ..., \vec{X}_{NP}^0] \quad (11)$$

Where, P^0 is the initial population of solution vectors, $\vec{X}_i^0$, for $1 \le i \le NP$ given by:

$$\vec{X}_i^0 = [x_{i1}^0, x_{i2}^0, ..., x_{iD}^0] \quad (12)$$

The jth component, or parameter, x_{ij}^0, for $1 \le i \le NP$, $1 \le j \le D$, is obtained from uniform distribution as follows:

$$x_{ij}^0 = x_j^L + (x_j^U - x_j^L) * rand_j, \quad (13)$$

Where, x_j^L and x_j^U are lower and upper bounds on parameter x_j and $rand_j$ is a random number in the range [0, 1].

Step 2:

Mutation: A mutant vector in the generation (G + 1) is created for each population vector by mutation:

$$\vec{V}_i^{G+1} = [v_{i1}^{G+1}, v_{i2}^{G+1}, ..., v_{iD}^{G+1}] \quad (14)$$

In this paper DE/best/1/bin is implemented for the design of FIR digital filter, hence a single difference of vectors is utilized. In DE/best/1/bin, a mutant vector $\vec{V}_i^{G+1}$ is generated for each target vector $\vec{X}_i^G$ by adding a weighted difference between two randomly selected distinct population vectors, $\vec{X}_{r1}^G$ and $\vec{X}_{r2}^G$, to the best vector $\vec{X}_{best}^G$ as follows:

$$\vec{V}_i^{G+1} = \vec{X}_{best}^G + F(\vec{X}_{r1}^G - \vec{X}_{r2}^G) \quad (15)$$

Where, $\vec{V}_i^{G+1}$ is a mutant vector, $\vec{X}_{best}^G$ is the best vector of the current population which gives the lowest cost function value; $r1$ and $r2$ are randomly chosen integers such that $r1, r2 \in \{1,2,...,NP\}$, $r1 \ne r2$; and F is a real and constant scaling factor which usually lies in the range [0, 1].

Step 3:

Crossover: To increase the diversity of population, crossover operation is used. This operation causes crossover or exchange of parameters of mutant vector with those of the target vector and generates trial vector $\vec{T}_i^{G+1}$ given by:

$$\vec{T}_i^{G+1} = [t_{i1}^{G+1}, t_{i2}^{G+1}, ..., t_{iD}^{G+1}] \quad (16)$$

In the binomial crossover scheme, uniform crossover is performed as follows:

$$t_{ij}^{G+1} = \begin{cases} v_{ij}^{G+1}, & if\ (rand_j \le CR \quad or \quad j = j_{rand}) \\ x_{ij}^G, & if\ (rand_j > CR \quad and \quad j \ne j_{rand}) \end{cases} \quad (17)$$

Where, t_{ij}^{G+1} is jth component of trial vector $\vec{T}_i^{G+1}$, v_{ij}^{G+1} is jth component of mutant vector $\vec{V}_i^{G+1}$ and x_{ij}^G is jth component of target vector $\vec{X}_i^G$. $rand_j$ is the jth evaluation of the random number in the range [0, 1]. CR is the crossover constant in the range [0, 1] and j_{rand} is randomly chosen index within the range [1, D]. As shown above, the trial vector component is adopted from the mutant vector $\vec{V}_i^{G+1}$, if the random number $rand_j$ is less than or equal to CR, or j is equal to index j_{rand}. Otherwise, the trial vector component is adopted from target vector $\vec{X}_i^G$. The index j_{rand} ensures that the trial vector $\vec{T}_i^{G+1}$ contains at least one parameter from mutant vector $\vec{V}_i^{G+1}$ and does not duplicate the target vector.

Step 4:

Selection: In order to decide whether trial vector $\vec{T}_i^{G+1}$, or, the target vector $\vec{X}_i^G$, is to be selected as the member of population vectors in next generation G + 1, the objective function is evaluated for target vector and trial vector. If the trial vector gives a smaller value of objective function, then this vector replaces the target vector for the next generation; otherwise, the old target vector is retained as follows:

$$\vec{X}_i^{G+1} = \begin{cases} \vec{T}_i^{G+1}, & if \quad f(\vec{T}_i^{G+1}) \le f(\vec{X}_i^G), \\ \vec{X}_i^G, & otherwise. \end{cases} \quad (18)$$

The process of mutation, crossover and selection is executed for all target vector index i and new population is created till the optimal solution is achieved. The procedure is terminated if maximum number of generations has been executed.

4.2 Bounce Back Technique for Handling Bounds on Parameters of Mutant Vector:

In the process of generating mutant vector, some of the components of this vector may cross the lower or upper bounds. In such cases bounce back mechanism [17] is adopted to bring such elements of the mutant vector within limit. In this method the element, which has violated the limits, is replaced by a new element whose value lies within the best vector value and the bound being violated. The following relations are used for violated mutant vector elements:

$$v_{ij}^{G+1} = \begin{cases} x_{best,j} + rand \cdot (x_j^L - x_{best,j}), & if\ v_{ij}^{G+1} \le x_j^L \\ x_{best,j} + rand \cdot (x_j^U - x_{best,j}), & if\ v_{ij}^{G+1} > x_j^U \end{cases} \quad (19)$$

Where, v_{ij}^{G+1} is j^th element of mutant vector, $\vec{V}_i^{G+1}$, $x_{best,j}$ is j^th element of the best vector, $\vec{X}_{best}^{G}$ and x_j^L, x_j^U are lower and upper bounds on parameter x_j respectively and $rand$ is a random number in the range [0, 1].

4.3 Ripple Constraint Handling methods:

In this paper two methods are used for ripple constraint handling. First we have discussed the method based on penalty function used by Sharma et al. [2]. Another method is based on modified selection rule as proposed by Lampinen [3].

4.3.1 Method # 1: Method Based on Penalty Function:

This method is based on penalty function, which penalizes infeasible frequency response ripple values obtained in the pass band and stop band. Thus, for a vector $\vec{X}$, the objective function with ripple constraint $J_2(\vec{X})$ is developed as follows:

$$J_2(\vec{X}) = c_L L_2(\vec{X}) + c_p \delta_p(\vec{X}) + c_s \delta_s(\vec{X}), \quad (20)$$
$$with\ \ c_L + c_p + c_s = 1.$$

Where, c_L, c_p and c_s are suitable weight parameters for $L_2(\vec{X})$, $\delta_p(\vec{X})$ and $\delta_s(\vec{X})$ respectively. $\delta_p(\vec{X})$ and $\delta_s(\vec{X})$ are the maximum pass band and stop band ripples given as follows:

$$\delta_p(\vec{X}) = \max_{\omega_i \in Passband} |1 - A(\omega_i)| \quad (21)$$

$$\delta_s(\vec{X}) = \max_{\omega_i \in Stopband} [A(\omega_i)] \quad (22)$$

Where, $A(\omega_i)$ is the magnitude of the frequency response of the filter, defined earlier in (3), for the suitable set of frequencies ω_i.

4.3.2 Method # 2: Method Based on Modified Selection Rule of DE:

The penalty function method uses additional control parameters, which are termed as the weight parameters. Setting the weight (or, penalty) parameters for getting their appropriate values by trial and error method, is a laborious task. The penalty function method effectively converts a constrained problem into an unconstrained one as shown by Lampinen [3]. This is seen from Equation (20) where, objective function $J_2(\vec{X})$ is used instead of $L_2(\vec{X})$.

In this subsection an improved version of constraint handling method used by Lampinen [3] is described. It allows to get rid of setting of the weight parameters for individual constraints. In this method, only the selection operation of the basic Differential Evolution algorithm is modified, for handling the ripple constraints. The selection criteria of Equation (18) to select either trial vector $\vec{T}_i^{G+1}$ or, target vector $\vec{X}_i^{G}$ for the next generation vector $\vec{X}_i^{G+1}$ is changed as follows:

- If both solution vectors satisfy all ripple constraints, then the one with lower objective function value is selected, OR,
- If target vector satisfies all ripple constraints, while trial vector does not satisfy, and if target vector also has lower objective function value then it is selected, OR,
- If target vector does not satisfy all ripple constraints but provides lower or equal value for all ripple constraints as compared to the trial vector, and also if target vector has lower objective function value then it is selected.
- Else, trial vector is selected.

5.0 DESIGN OF LINEAR PHASE FIR FILTERS USING DE WITH RIPPLE CONSTRAINTS

This section is divided into two subsections. Subsection 5.1 describes the specifications of the digital low pass FIR filter. The design parameters of DE algorithm with ripple constraints for Method # 1 and Method # 2 are discussed in subsection 5.2.

5.1 Specifications of The Digital Low Pass FIR Filter Designed:

Type-I linear phase FIR filter is designed with the filter length taken as N+1 = 31, and the grid of digital angular frequencies as K = 180. The cut-off frequency of the pass band is $\omega_p = 0.3\pi$ and cut-off frequency of the stop band is $\omega_s = 0.4\pi$. The desired ideal frequency response *D(ω)* has unity gain in the pass band and zero gain in the stop band and is given by:

$$D(\omega) = \begin{cases} 1, & 0 \le \omega \le 0.3\pi \\ 0, & 0.4\pi \le \omega \le \pi \end{cases} \quad (23)$$

For passband attenuation δ_p = 0.06 and stopband attenuation δ_s = 0.06, the weighting function $W(\omega)$ used, is given by:

$$W(\omega) = \begin{cases} (\delta_s / \delta_p) = 1, & in\ \ the\ \ passband \\ 1, & in\ \ the\ \ stopband \end{cases} \quad (24)$$

5.2 Design Parameters of Differential Evolution Algorithm with Ripple Constraint for Method # 1:

For applying DE with ripple constraint for obtaining the coefficients of the filter, $a(k)$; the size of each solution vector, $\vec{X}$, is taken as $D = M + 1 = 16$. The scaling factor F

and crossover constant *CR* used are taken as, *F* = 0.5 and *CR* = 0.7. The population size is taken as *NP* = 50. The numbers of generations used are 200. The weight parameters c_L, c_p and c_s used in the objective function of (20) are arbitrarily set as 0.0006, 0.2894 and 0.71respectively so as to reject larger values of objective function for larger size of ripples. It is observed that the amplitudes of the ripples are larger in the stop band, so more weight is assigned to c_s as compared to c_p.

5.3 Design Parameters of Differential Evolution Algorithm with Ripple Constraint for Method # 2:

There is no specific parameter used for ripple constraint handling Method # 2. All other parameters of DE algorithm are kept same as mentioned in the subsection 5.2.

6.0 SIMULATION RESULTS AND ANALYSIS

The summary of the parameters obtained for the best of 30 runs, for Type-I FIR low pass filter (LPF) design, using DE with ripple constraint Method # 1 and Method # 2, is shown in the TABLE-1 and TABLE-2 for the time-domain and the frequency domain respectively. In TABLE-3, the statistical parameters obtained for the two cases are compared.

In Fig. 1, the error plots for the best run of the two cases are shown. The frequency responses obtained for the filter design using DE with ripple constraint methods are compared in Fig. 2. It is observed from Fig. 1, that the absolute value of error obtained for Method # 2 is lower than that obtained for Method # 1. This fact is also exhibited in TABLE-3.

By comparing the frequency responses in Fig. 2, it is observed that ripple constraint handling Method # 2 shows a better frequency response as compared to Method # 1. It is also obse-

Table 1: Time-Domain Parameters

DE with ripple constraint Method # 1	DE with ripple Constraint Method # 2
Impulse response h(n)	**Impulse response h(n)**
h(0) = – 0.0033 = h(30)	h(0) = – 0.0058 = h(30)
h(1) = 0.0081 = h(29)	h(1) = 0.0025 = h(29)
h(2) = 0.0170 = h(28)	h(2) = 0.0104 = h(28)
h(3) = 0.0090 = h(27)	h(3) = 0.0071 = h(27)
h(4) = – 0.0080 = h(26)	h(4) = – 0.0051 = h(26)
h(5) = – 0.0226 = h(25)	h(5) = – 0.0201 = h(25)
h(6) = – 0.0104 = h(24)	h(6) = – 0.0136 = h(24)
h(7) = 0.0229 = h(23)	h(7) = 0.0154 = h(23)
h(8) = 0.0356 = h(22)	h(8) = 0.0367 = h(22)
h(9) = 0.0120 = h(21)	h(9) = 0.0195 = h(21)
h(10) = – 0.0412 = h(20)	h(10) = – 0.0385 = h(20)
h(11) = – 0.0708 = h(19)	h(11) = – 0.0717 = h(19)
h(12) = – 0.0167 = h(18)	h(12) = – 0.0195 = h(18)
h(13) = 0.1316 = h(17)	h(13) = 0.1250 = h(17)
h(14) = 0.2804 = h(16)	h(14) = 0.2839 = h(16)
h(15) = 0.3493	h(15) = 0.3535

-rved that the stop band attenuation is decreased further with the increase in frequencies in the case of Method # 2. Finally, from the TABLE-3 it is observed that the frequency of convergence of Method # 2 is higher than that of Method # 1; however, the value of standard deviation obtained for Method # 2 is larger.

Table 2: Frequency-Domain Parameters

Parameter	DE with ripple constraint Method # 1		DE with ripple constraint Method# 2	
	Pass Band	**Stop Band**	**Pass Band**	**Stop Band**
Lower Band Edge	0.0000	0.4000 π	0.0000	0.400 π
Upper Band Edge	0.3500 π	1.0000 π	0.3500 π	1.000 π
Desired Value: D(ω)	1.0000	0.0000	1.0000	0.0000
Maximum ripple	0.0243	0.0231	**0.0229**	**0.0226**
Maximum ripple (dB)	0.2087	– 32.70	**0.1972**	**– 32.88**
Minimum ripple	0.0010	0.0057	0.0010	**0.0022**
Minimum ripple (dB)	0.0094	– 44.78	**0.0091**	**– 52.81**

Table 3: Statistical Parameters of DE With Ripple Constraint Method # 1 And Method # 2 for the Low Pass Filter

(Number of runs = 30; Number of generations = 200; Population Size = 50.)

Sr. No.	Parameters	DE with ripple constraint Method # 1	DE with ripple constraint Method # 2
1	Best fitness value of error of all runs	0.2711	**0.1552**
2	Average value of minimum error of all runs	0.4599	**0.3174**
3	Worst minimum value of all runs	0.9015	**0.7021**
4	Standard deviation of minimum error from average	**0.1389**	0.1440
5	Frequency of convergence*	0.5000	**0.6000**

*(Frequency of convergence = number of better fitness values than mean out of all runs / total no. of runs)

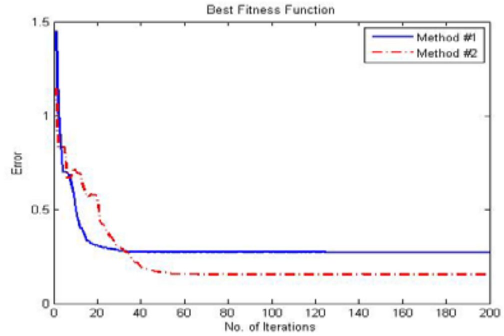

Figure 1: Error plots of Type-I FIR LPF obtained using DE with ripple constraint Method # 1 and Method # 2.

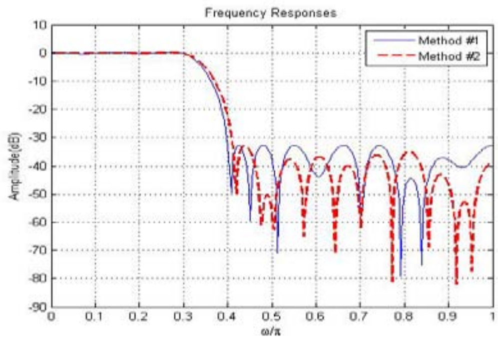

Figure 2: Comparison of Magnitude Frequency responses of Type-I FIR LPF for DE with ripple constraint Method # 1 and Method # 2.

7.0 CONCLUSION AND FUTURE SCOPE

In this paper an efficient alternative method for the design of linear phase digital FIR filter with ripple constraint is discussed. This method of optimization of the filter coefficients uses DE algorithm with modified selection rule for ripple constraint handling. The results obtained using this method are compared with those obtained for another method of ripple constraint handling based on penalty function. From the simulation results it can be concluded that ripple constraint handling method based on the modified selection rule of DE shows better performance than that obtained using DE with ripple constraint method based on penalty function. Thus this method is seen as an efficient alternative method for ripple constraint handling with DE algorithm for FIR filter design. In future the population size of the DE algorithm can be varied and statistically better results are expected.

8.0 REFERENCES

[1]. A. V. Oppenheim and R. W. Schafer, Discrete-Time Signal Processing, 3rd ed. Prentice-Hall of India, New Delhi, 2013.

[2]. S. Sharma, L. D. Arya and S. Katiyal, "Design of linear-phase digital FIR filter using Differential evolutional optimization with ripple constraint", International conference on Sustainable Global Development, INDIACom-2014, pp. 474-480, March 2014.

[3]. J. Lampinen, "Multi-Constrained Nonlinear Optimization by the Differential Evolution Algorithm", Journal of Soft Computing and Industry, Springer, pp. 305-318, 2002.

[4]. J. G. Proakis and D. G. Manolakis, Digital Signal Processing: Principles, Algorithms and Applications, 4th ed., Pearson Education, 2007.

[5]. S. V. Chande1 and Dr. M. Sinha, "Genetic Algorithm: A Versatile Optimization Tool", BIJIT – 2009, Vol. 1 No. 1, January – June, 2009.

[6]. S. Routray, "An Enhanced Genetic Algorithm Approach to ATM Network Design", BIJIT – 2011, Vol. 3 No. 1, January – June, 2011.

[7]. A. G. Aggarwal, P. K. Kapur, G. Kaur and R. Kumar "Genetic Algorithm Based Optimal Testing Effort Allocation Problem for Modular Software", BIJIT – 2012, Vol. 4 No. 1, January - June, 2012.

[8]. D.J. Xu and M.L. Daley, "Design of optimal digital filter using a parallel genetic algorithm", IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, Vol. 42, No. 10, pp. 673-675, October 1995.

[9]. L. Cen, "A hybrid genetic algorithm for the design of FIR filters with SPoT coefficients", Signal Processing, Vol. 87, pp. 528-540, March 2007.

[10]. Kennedy J. and Eberhart R., "Particle Swarm Optimization", In: Proc. IEEE Int. Conf. on Neural Networks, Piscataway, pp. 1942-1948, 1995.

[11]. D. Goyal, M. R. Tripathy, "Path optimization using APSO", BIJIT – 2013, Vol. 5 No. 2, July – Dec., 2013.

[12]. J. I. Ababneh and M. H. Bataineh, "Linear phase FIR filter design using particle swarm optimization and genetic algorithms", Digital Signal Processing, Vol. 18(4), pp. 657-668, July 2008.

[13]. S. Sharma and L. D. Arya, "Design of FIR digital filters using particle swarm optimization", IE(I) Journal-ET, Vol. 91, pp.11-19, July 2010.

[14]. S. Mandal, S. P. Ghoshal, R. Kar, D. Mandal, " Design of optimal linear phase FIR high pass filter using craziness based particle swarm optimization technique", Journal of King Saud University – Computer and Information Sciences, Vol.24, pp. 83–92, 2012.

[15]. M. Shukla, G. R. Mishra, "DAPSO and PSO-VAF in Linear Phase Digital Low Pass FIR Filter Design", Circuits and Systems, Vol. 5, pp. 57-67, 2014.

[16]. R. Stron and K. Price, "Differential Evolution – A simple and efficient heuristic for global optimization over continuous spaces", Journal of Global Optimization, Vol. 11, pp. 341-359, 1997.

[17]. L. D. Arya, S. C. Chaube and R. Arya, "Differential evolution applied for reliability optimization of radial distribution systems", IJEPES, Elsevier , Vol. 33, Issue 2, Feb. 2011.

[18]. Q. Zhao, G. Meng, "Design of Digital FIR Filters using Differential Evolution algorithm based on Reserved Gene", International Conference of Information Science and Management Engineering, pp.177-180, 2010.

[19]. Z. Albataineh, F. Salem and J. I. Ababneh, "Linear phase FIR Low Pass Filter Design Using Hybrid Differential Evolution", International Journal of Research in Wireless Systems (IJRWS), vol. 1, issue 2, pp. 43-49, 2012.

[20]. K. Singh N. Kaur, "Design of Digital FIR Filters Using Differential Evolution Algorithm Based on Reserved Genes", Vol. 4, Issue 7, pp. 813-819, July 2014.