Design and optimization of digital FIR filter coefficients using Genetic algorithm

Putta Prabhu Thapaswini, Umadevi.S, Seerangasamy.V

Abstract— **This research work presents a method to design FIR filter with the optimized filter coefficients using Genetic Algorithm (GA). The coefficient is been represented by using Canonical Signed Digit (CSD) format. The CSD format representation is been chosen to reduce the hardware complexity and hardware cost. The goal of this research is to select a best coefficient for the filter design and the wordlength reduction in the coefficient, so that the desirable magnitude response is been met with the hardware complexity. The obtained Genetic Algorithm (GA) results are compared with windowing technique's results & park.Mc.clellan's or equiripple algorithm results. The implementation is done using MATLAB.**

Index Terms— **Canonical Signed Digit (CSD), Equiripple algorithm, Genetic Algorithm(GA), optimization.**

I. INTRODUCTION

In this paper, FIR filters would be favored on account of their features like stability& linear phase. A FIR's usage noise characteristics are easy to model & because of its all zero structure, it has a linear phase response. However for administering this, one must pay those value in the form of vast number of multipliers, due to this vast number for multipliers the speed of the processor will be reduced, thereby expanding the equipment cosset additionally. [1]. To beat this, the coefficients are represented in CSD format. This design need no multipliers so, hardware complexity may be decreased and cosset may be also lessened What's more additionally is high speed requirements are satisfied.

CSD format diminishes the multiplier operation to shift and add operations. Similarly as the multiplier is reduced to a minimum shift and add operations, the hardware complexity is also reduced, thereby reducing the hardware cost and automatically speed will be increased [1]. Therefore, in this paper we utilize CSD format to represent the coefficients. Along with the reduction of hardware complexity & hardware cost, filter coefficients optimization is a critical task. In this paper, for the optimization, a method was proposed, first of all the initial set of coefficients which are given as input to genetic algorithm are optimized filte coefficients obtained from equiripple algorithm because compared to coefficients obtained from windows technique equiripple algorithm results are more optimum according to the results obtained. Equiripple algorithm is used because equiripple outputs and genetic algorithm outputs are practically same if hardware

cost & complexity are not taken into consideration [2], that's where genetic algorithm is needed.

Idea behind the suggested technique is to optimize the filter coefficients simultaneously and also the hardware cost & complexity are taken consideration at the same time guaranteeing high speeds.

II. WINDOW METHOD

The simplest design of FIR filters is attained utilizing the windowing technique. In this technique, the design starts with an ideal desired frequency response which is given by[2]

$$H_d(e^{j\omega}) = \sum_{n=-\infty}^{\infty} h_d[n] e^{-j\omega n} \tag{1}$$

$$h_w[n] = h_d[n] d[n] \tag{2}$$

Where $h_d[n]$ is the impulse response of the filter
$d[n]$ is the window function
$h_w[n]$ is the windowed impulse response

As a result, the frequency response of the windowed impulse response is the periodic convolution of the desired frequency response with the Fourier transform of the window and which is given by

$$H_w(e^{j\omega}) = \frac{1}{2\pi} \int_{-\pi}^{\pi} H_d(e^{j\varphi}) D(e^{j(\omega-\varphi)}) d\varphi \tag{3}$$

The frequency response of the approximation filter will be given by

$$H(e^{j\omega}) = e^{-j\omega n_d} H_w(e^{j\omega}) \tag{4}$$

where n_d will be the essential time delay to present causality in the approximated filter. The filter designed by windowing technique will have greatest error on either side of the discontinuity of the ideal frequency response[2]. The length of the estimated filter (N) should be as short as possible. Furthermore, the windowing technique doesn't tolerate singular control over the close estimation errors in various bands.

Preferred filters result from minimization of maximum error in both, the stop band and the pass band of the filter which leads to equiripple filters[2].

III. EQUIRIPPLE ALGORITHM

Equiripple algorithm is also known as Park.Mc.Clellan's algorithm. It is used to optimize the filter coefficients as it provides the optimum result when the hardware cost & complexity are not taken into consideration. In this algorithm the design of linear phase FIR filter is predicated on the min-max error paradigm i.e.., minimizing the maximum error possible [2], which there by produces the optimum result. The method implemented is an iterative numerical algorithm. It

converges very rapidly to the optimal solution. This method is very robust and is broadly utilized because of the optimum results (in spite of hardware complexity). In equiripple algorithm N (filter length) ω_p, ω_s, and the ratio $\delta p/\delta s$ are fixed and it is considered as most popular approach because of its flexibility and computational efficiency [2].

The weighted error function is given by

$$E(w) = (A(w) - D(w)) \qquad (5)$$

Given,

N $\longrightarrow$ filter length

$D(\omega)$ $\longrightarrow$ desired (real valued) amplitude function

$W(\omega)$ $\longrightarrow$ non negative weighting function

FIR filter's amplitude response can be formulated as

$$A(w) = \sum_{n=0}^{N} a(n) \cos(nw) \qquad (6)$$

$$D(\omega) = \begin{cases} 1, & 0 < \omega < \omega_0 \ (pass\ band) \\ 0, & \omega_0 < \omega < \pi \ (stop\ band) \end{cases}$$

$$W(\omega) = \begin{cases} k_p & 0 \le \omega \le \omega_p \\ 0 & \omega_p < \omega < \omega_s \\ k_s & \omega_s \le \omega \le \pi \end{cases}$$

Where $0 < w_p < w_0 < w_s < \prod$

Maximum error

$$\|E(w)\|_\infty = \max_{w \in [0,\pi]} |W(w)(A(w) - D(w))| \qquad (7)$$

The solution to this quandary is called best weighted approximation to $D(\omega)$, similarly as it provides the optimum result and it also minimizes the maximum value of error (minmax solution)[4].

So, for computing this solution alteration theorem is used. Alteration theorem states that, $|E(w)|$ attains its maximum value at a minimum of M+2 points & the weighted error function alternates sign on at least M+2 of those points[4].

Let us say,

R=M+2

Where, R is a reference set

$$E(w_i) = c(-1)^i \|E(w)\|_\infty \qquad (8)$$

From eqn. (2), we can write the above equation as

$$W(w_i)(A(w_i) - D(w_i)) = (-1)^i \delta \qquad (9)$$

$$A(w_i) - D(w_i) = \frac{(-1)^i \delta}{W(w_i)} \qquad (10)$$

The above equation represents interpolation problem, which can be represented in the matrix form as

$$\begin{bmatrix} 1 & \cos w_1 \cdots \cos M_{w1} & 1/W(w_1) \\ 1 & \cos w_2 \ldots \cos M_{w2} & -1/W(w2) \\ 1 & \cos w_R \ldots \cos M_{wR} & -(-1)^R/W(w_R) \end{bmatrix} \begin{bmatrix} a(0) \\ a(1) \\ \vdots \\ \delta \end{bmatrix} = \begin{bmatrix} D(w_1) \\ D(w_2) \\ \vdots \\ D(w_R) \end{bmatrix}$$

There will be an unique solution to this linear system of equations, that solution is taken as optimal minmax solution[4]. This utilization takes excessive amount of computer time and memory, so it is not very useful [3] & we cannot guarantee convergence in equiripple algorithm where as in genetic algorithm convergence is guaranteed.

Algorithm steps are

1. Interpolation can be repeated by taking new set of R points.

2. Updating the reference (R) set till the desired optimized result is met.

Those acquired equiripple coefficients are converted into Canonical Signed Digit (CSD) format. CSD format is used to represent the binary numbers. In this format, binary numbers are encoded in such a way that word has fewest number of non zero digits. This technique maps a tenary system to a binary system [5].

Properties of CSD are:

1. No two consecutive bits in the encoded bit are non zeros.

2. The encoded word should contain a minimum number of non-zero digits.

For example, to represent 23 in CSD, csdigit(23) returns +0-00-. Where csdigit is a in-built function in MATLAB. CSD contains 33% fewer non-zero bits than 2's complement digits [5].

IV. GENETIC ALGORITHM

Offspring can be produced in more number by the organisms and the offspring produced can be similar to the parent organisms but some offspring can have similarities that none of the parent organism have due to mutations (random changes) and some characteristics are inherited through reproduction from parent organisms. The characteristics it has inherited through mutations and reproduction helps it to survive [6] or not to survive while undergoing through natural changes in the environment. To understand this adaptive process of nature and to apply it so that it'll be desensitized to changes, John Holland (1970) developed algorithm called Genetic Algorithm (GA).It is a heuristic algorithm and it finds the optimal results by decreasing the value of objective function continuously. It provides both efficient & effective techniques for optimization. The major hindrance in the conventional application of GA is about the premature convergence, Whether those body of evidence happens. GA produces a sub-optimal result In there, may be no outside 'strength' or 'instruction' which lead GA out of local optima[7] which is dealt in this method by iterating over population many times and simultaneously iterating different regions of population.

Basic steps of GA are:

A. A.Population

Encode each bit as a gene and a string of genes are called as chromosome and set of chromosomes is called population. First step to start with is to initialize the population i.e., is called as initial population.

B.Evaluation

Each chromosome has to be assessed and to be assigned a value called fitness value, larger the fitness value, (that says i.e.., a good gene) probability will be more to select it for reproduction. Fitness is the measure of goodness of a chromosome.

C.Selection

The individual chromosomes which have best fitness values are selected and proceeded for next step called reproduction where blending of the both guardians would be carried to process new offspring's [8]. Two regularly utilized

methods are 'roulette wheel' & 'tournament' selection Over roulette wheel, every individual will be allocated a sector (slice) size proportional to their fitness evaluated The wheel is then spun and the individual inverse to the marker turns into a standout amongst those as parents[2].In this paper roulette wheel selection is used.

D. Reproduction

Two chromosomes which are selected based on the fitness value [9] from the population undergo a process called reproduction to produce offspring's. Parents who have better fitness values have superior possibilities to be selected for production of finer offsprings.

E. Crossover

Choosing a random point and Splitting the parents at this crossover point and Creating children by trading their tails is called crossover process. Crossover probability is typically in the range (0.6, 0.9) [10]. Fig .1 describes the crossover process. It can be classified as

a. Simple Crossover: The Process we discussed above is a simple crossover process[2].

b. Arithmetic Crossover : It is a process where two complimentary linear combinations of the parents were produced[2].

c. Heuristic Crossover : It gets profit from fitness information with furthermore extrapolation of the two individuals to enhance the offspring[2].

In this paper, simple crossover was used.

Fig 1. Crossover Process

I	I	I	0	0	I	0	0	I

I	I	I	0	0	I	0	I	0

I	I	I	0	0	I	0	I	0

| I	I	I	0	0	I	0	0	I
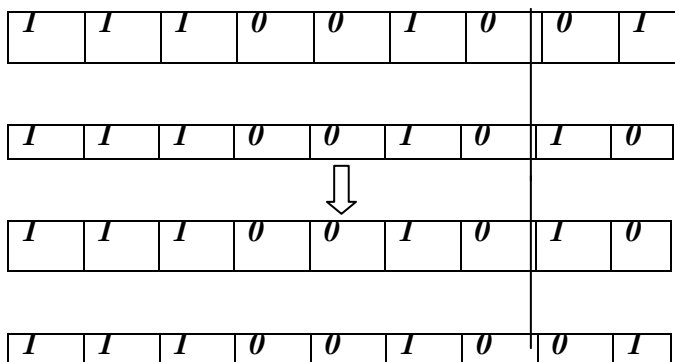
F. Mutation

Mutations are random changes done to a chromosome in order to get a good fitness value as shown in Fig. 2. Mutations are random changes which reintroduces the genetic diversity into the population [9].

Mutation can also be classified as
a. Uniform mutation
b. Non uniform mutation
c. Multi-non uniform mutation
In this paper, non uniform method is used where one variable (gene in a chromosome) is selected randomly and its equivalent is set to a non uniform random number.

Fig 2. Mutation

I	0	0	0	I

I	0	0	0	I
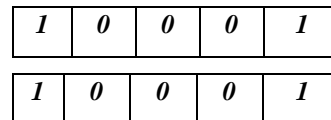
Then afterward finishing every last one of above steps, if the desired criteria are met by the coefficients, then it'll stop, otherwise the process will be continued along with the new generated offsprings as a population until the required specifications met.

V. DESIGN PROCEDURE

Here, the aim is to design a FIR filter with desired frequency, magnitude response, and to optimize the filter coefficients this can be achieved by the following methods of proposed technique.

A. A. Window method

B. 1. On deciding the specifications of the filter that has to be kept constant for equiripple & genetic algorithms

2. Design a filter using any window lets say hamming window
3. Calculate the magnitude response & filter coefficients for particular filter length (in the paper N= 23).

C. B. Implementation of equiripple algorithm

1. We have to decide the filter specifications and the length.
2. After deciding the length, ω_p (normalized pass band edge) & ω_s (normalized stop band edge), the main idea of equiripple algorithm starts with taking reference set as explained in section II before.
3. The magnitude response of the filter designed is compared with the desired magnitude response and error E(ω) is calculated.
4. 4.If error is more i.e. magnitude response of filter observed are not coinciding with the desired magnitude response then a new reference set will be taken.
5. 5. Again, the step repeats until error function is less than smallest number defined as SN in the program(i.e. taken SN=1e-8)
6. 6.If it is less than SN i.e. max (err) < SN, then result is displayed as converged.

C. CSD conversion

1. The filter coefficients thus obtained are converted into CSD format.
2. They are converted into CSD format by using the MATLAB built-in function (csdigit)

[a,p,n] = csdigit (num, range, resolution)
Where; a is the CSD digit
p is the positive part
n is the negative part
num = input number (decimal)
range = maximum digits to the left of the decimal point

resolution = digits to the right of decimal point
for example,
csdigit(.25,2,2) represents xx.xx binary
csdigit(.25,0,4) represents .xxxx binary

D.Genetic algorithm

1. The first step in genetic algorithm is to define the objective function

$$f=max(err)$$
$$err = W(\omega)(A(\omega)-D(\omega)) \qquad (12)$$

2. The coefficients obtained in equiripple method is taken as initial population

3. Then, we've to calculate fitness value for each individual in the population.

4. Fitness value is calculated by the fitness function given below

$$ff = \frac{1}{f} + k(SONZ) \qquad (13)$$

where, f is the objective function defined at the starting of the algorithm

$$k=0.02$$

SONZ=sum of non zeros in CSD digit.

5. Two individuals (parents) who have fitness value more are selected for reproduction to produce offsprings.

Eg: P1 00+00-
             P2 0+0000
             O1 00+00
             O2 0+00-

where P1, P2 are parents
& O1, O2 are offsprings.

6. After crossover, the next step is mutations (random changes of variables), so that it can produce better fitness value.

7. If this population gives the desired frequency response with optimized filter coefficients, stop the algorithm otherwise include the offsprings & create a new generation & repeat all the steps as mentioned above.

Fig 3. Shows the typical flow of the proposed algorithm
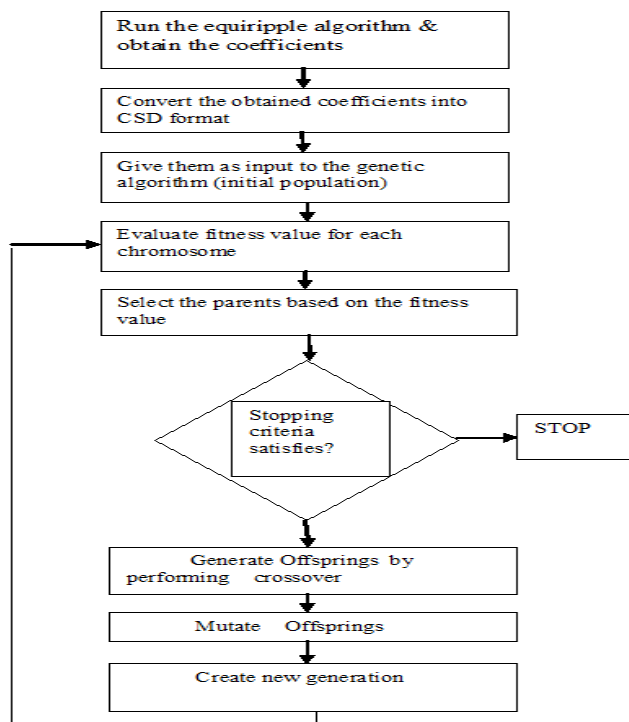
TABLE I. FILTER COEFFICIENTS

Coefficients	Values
1, 23	$2^{-10} + 2^{-13} + 2^{-15}$
2, 22	$2^{-9} + 2^{-12} + 2^{-15}$
3, 21	$2^{-8} - 2^{-12} + 2^{-14}$
4, 20	$-2^{-8} - 2^{-11}$
5, 19	$-2^{-6} + 2^{-8} - 2^{-10} - 2^{-13} - 2^{-15}$
6, 18	$2^{-8} + 2^{-10} + 2^{-12} - 2^{-15}$
7, 17	$2^{-5} + 2^{-9} - 2^{-12} - 2^{-15}$
8, 16	$2^{-8} - 2^{-14}$
9, 15	$-2^{-4} - 2^{-7} - 2^{-10} + 2^{-13}$
10, 14	$-2^{-4} + 2^{-6} + 2^{-8} - 2^{-10} + 2^{-12} + 2^{-15}$
11, 13	$2^{-2} - 2^{-4} - 2^{-6} + 2^{-9} + 2^{-13}$
12	$2^{-1} - 2^{-3} + 2^{-5} + 2^{-8} + 2^{-10} + 2^{-12} + 2^{-14}$

The TABLE I shows the filter coefficients obtained from windowing technique.

TABLE II. FILTER COEFFICIENTS

Coefficients	Values
1, 23	$2^{-7} + 2^{-10}$
2, 22	0
3, 21	$2^{-6} - 2^{-9} - 2^{-11}$
4, 20	0
5, 19	$-2^{-5} + 2^{-7} - 2^{-9} - 2^{-11}$
6, 18	0
7, 17	$2^{-4} - 2^{-6} + 2^{-9} - 2^{-11}$
8, 16	0
9, 15	$-2^{-3} + 2^{-5} - 2^{-9} - 2^{-11}$
10, 14	0
11, 13	$2^{-2} + 2^{-4} + 2^{-9} - 2^{-11}$
12	2^{-1}

The TABLE II shows the filter coefficients obtained from equiripple algorithm.

TABLE III. FILTER COEFFICIENTS

Coefficients	Values
1,27	2^{-8}
2,26	0
3,25	2^{-7}
4,24	0
5,23	2^{-6}

6,22	0
7,21	$-2^{-5} + 2^{-8}$
8,20	0
9,19	$2^{-4} - 2^{-6} + 2^{-8}$
10,18	0
11,17	$-2^{-3} + 2^{-5} - 2^{-8}$
12,16	0
13,15	$2^{-2} + 2^{-5}$
14	2^{-1}

The TABLE I shows the filter coefficients of FIR filter designed according to the windowing technique.TABLE II. shows the filter coefficients of FIR filter designed according to the equiripple algorithm. TABLEIII. Shows the filter coefficients of FIR filter designed according to the genetic algorithm. On comparision of TABLE I, TABLEII & TABLEIII it can be observed that non zero digits are reduced in coefficients obtained from genetic algorithm than in the equiripple algorithm coefficients or coefficients obtained in windowing technique. That proves the reduction in hardware complexity.

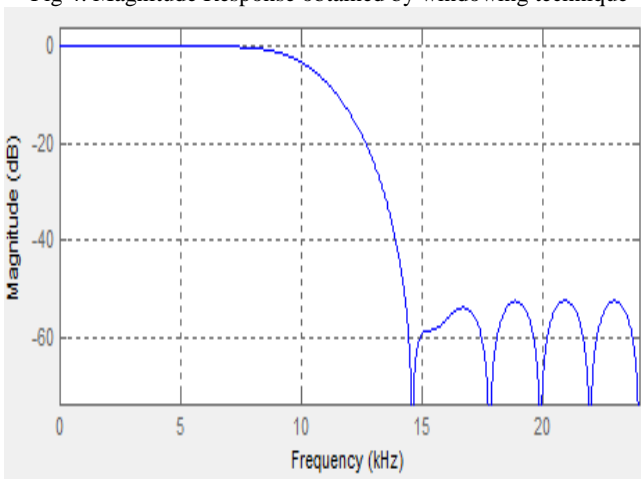Fig 4. Magnitude Response obtained by windowing technique
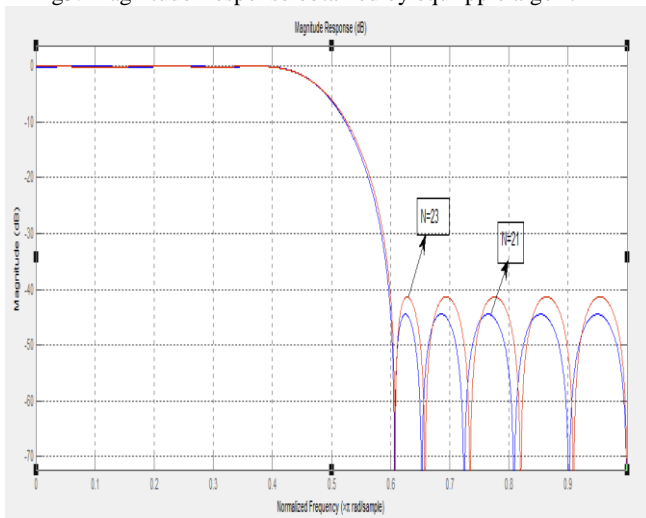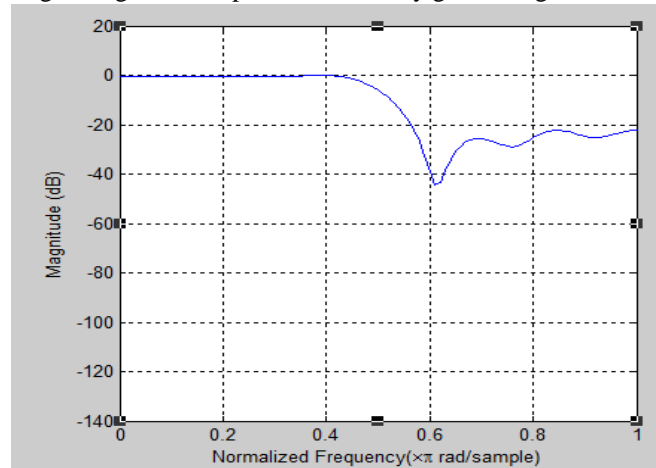

Fig5. Magnitude Response obtained by equripple algorithm

Fig5. Magnitude Response

Fig 6.Magnitude response obtained by genetic algorithm

The response shown in Fig 4. indicates the magnitude response of the FIR filter designed by windowing (hamming) method. It can be observed that the response has some fluctuations andpass band is not completely flat. The response shown in Fig 5. indicates the magnitude response of the FIR filter designed by equiripple method. It can be observed that the response is not purely flat in pass band and the Fig. 6. represents the magnitude response of filter designed according to genetic algorithm, in this it is obvious that the error has minimized to complete zero in the pass band i.e., the pass band is totally flat in Fig4. That means the error is completely minimized to zero in the pass band i.e., a flat response for pass band is a desired characteristic .

Using MATLAB software to design a FIR filter by window method, an FIR filter with 23 coefficients and of wordlength 15 satisfies the given specifications

The same filter to be designed using equiripple algorithm it'll cost 23 coefficients and wordlength 11.

If such a filter is designed using proposed method (GA) an FIR filter with 27 coefficients and wordlength 8 satisfies the given specifications.

It can be observed that GA has minimum wordlength so minimum hardware complexity and a desired frequency response. So it can be concluded that of all the three methods GA is the most optimum method.

VII. CONCLUSION

The proposed technique achieves the optimum number of coefficients required to get the desired frequency response with the optimum wordlength. Compared to window method & equiripple method the proposed technique results in about 35-40% reduction in hardware cost.

VIII. FUTURE SCOPE

The magnitude response precision can still be increased by giving this obtained best genetic algorithm coefficients as a input to the another genetic algorithm procedure which has a fitness function defined for improving the precision of magnitude response. Currently I am extending this work with the above mentioned method.

R<small>EFERENCES</small>

[1] A. R. Rezaee,"Using genetic algorithms for designing of fir digital filters," ICTACT journal on soft computing, july 2010, Issue: 01, pp. 18-22.

[2] Jehad I.Ababneh, Mohammad H. Bataineh , "Linear phase FIR filter design using particle swarm optimization and genetic algorithms",Elsevier, vol. 18 Issue:4.

[3] P. Janardhan, M. N. Neelakantan, "A fast perturbation method for the design of linear phase fir digital filters of finite wordlength using single frequency filters," springer journal on circuit systems and signal processing, vol. 10,Issue 2, pp. 233-244.

[4] I.Selesnick,"The Remez Algorithm".

[5] A. Rajolia, M. Kaur, "Finite impulse response (fir) filter design using canonical signed digits (csd)," International Journal of science and research (IJSR), india online issn: 2319-7064.

[6] R. Singh, S. K. Arya, " Genetic algorithm for the design of optimal iir digital filters," *Journal of Signal and Information Processing*, 2012, Issue: 3, pp. 286-292.

[7] Ling Cen,"A hybrid genetic algorithm for the design of FIR filters with SPoT coefficients",Elsevier vol.87,, Issue:3.

[8] P. Kaur, S. Kaur, "Optimization of fir filters design using gentic algorithm," IJETTCS, Issue :3, Vol. 1.

[9] S. T. Tzeng, "Genetic algorithm approach for designing 2-D FIR digital   filters with 2-D symmetric properties," Elsevier, vol. 84, Issue: 10.

[10] S. Aggarwal, A. Gagneja, A. panghal, "Design of fir filter using ga and its comparison with hamming window and parks mcclellan optimization techniques," IJARCSSE, vol. 2, Issue: 7.

[11] A. N. Kani, "Digital signal processing", second edition.

[12] A. Lee, M. Ahmadi, G.A. Jullien, W.C. Miller,   R.S.Lashkari," Digital filter design using genetic algorithm", in:Proceedings of IEEE Symposium on Advances in Digital Filtering and Signal Processing, June 1998.

[13] E. Hostetter, A.V. Oppenheim, J. Siegel, On optimum nonrecursive digital filters, in: Proc. 9th Allerton Conf. Circuit System Theory, October 1971.

[14] T.W. Parks, J.H. McClellan, Chebyshev approximation for nonrecursive digital filters with linear phase, IEEE Trans. Circuit Theory CT-19(1972) 189–194.

[15] T.W. Parks, J.H. McClellan, A program for the design of linear phase finite impulse response filters, IEEE Trans. Audio Electroacoust. AU-20 (3) (1972) 195–199.

[16]   J.H. McClellan, T.W. Parks, A unified approach to the design of optimum FIR linear phase digital filters, IEEE Trans. Circuit Theory CT-20(1973) 697–701.

[17] J.H. McClellan, T.W. Parks, L.R. Rabiner, A computer program for designing optimum FIR linear phase digital filters, IEEE Trans. Audio Electroacoust. AU-21 (1973) 506–526.

[18] A.V. Oppenheim, R.W. Schafer, J.R. Buck, Discrete-Time Signal Processing, Prentice Hall, Englewood Cliffs, NJ, 1999.

[19] J.H. Holland, Adaptation in Natural and   Artificial Systems,MIT Press, 1975.

[20] Mitra, Digital Signal Processing: A Computer- Based Approach, McGraw-Hill, 2001.

[21] K.F. Man, K.S. Tang, S. Kwong, Genetic Algorithms:Concepts and Designs, Springer, London, 1999.

[22] M. Srinivas, L.M. Patnaik, Genetic algorithms: a survey,Computer 27 (6) (June 1994) 17–26.

.