International Journal of Scientific Engineering and Technology (ISSN : 2277-1581)
Volume 2 Issue 5, pp : 398-403 1 May 2013

Performance Evaluation of Evolutionary Algorithms for Digital Filter Design

Ashutosh P., Kasambe P. V.
Department of Electronics Engineering, Sardar Patel Institute of Technology, Mumbai, Maharashtra, India.
ashp3050@gmail.com

Abstract- ***In this paper, Particle Swarm Optimization (PSO) and Genetic Algorithm (GA) have been used to design digital IIR filter and the filter responses have been compared with the desired responses. Simulation results show that PSO produces a response which is closest to the desired response. Hence, PSO algorithm can be effectively used to design digital filters.***

Keywords- Digital IIR Filters, Digital FIR Filters, Particle Swarm Optimization, Genetic Algorithm.

I. INTRODUCTION

Signals arise in almost every field of science and engineering. Two general classes of signals can be identified, namely, continuous-time and discrete-time signals. A discrete-time signal is one that is defined at discrete instants of time. The numerical manipulation of signals and data in discrete-time signals is called digital signal processing (DSP). Almost any DSP algorithm or processor can reasonably be described as a filter. Filtering is a process by which the frequency spectrum of a signal can be modified, reshaped, or manipulated according to some desired specifications. Digital filters can be broadly classified into two groups: recursive and nonrecursive. The response of nonrecursive (FIR) filters is dependent only on present and previous values of the input signal. However, the response of recursive (IIR) filters depends not only on the input data but also on one or more previous output values. The main advantage of an IIR filter is that it can provide a much better performance than the FIR filter having the same number of coefficients. The main advantage of a digital IIR filter is that it can provide a much better performance than the FIR filter having the same number of coefficients [1].

Design of a digital filter is the process of synthesizing and implementing a filter network so that a set of prescribed excitations results in a set of desired responses. Like most other engineering problems, the design of digital filters involves multiple, often conflicting, design criteria and specifications, and finding an optimum design is, therefore, not a simple task. Analytic or simple iterative methods usually lead to sub-optimal designs. Consequently, there is a need for optimization-based methods that can be used to design digital filters that would satisfy prescribed specifications. However, optimization problems for the design of digital filters are often complex, highly nonlinear, and multimodal in nature. The problems usually exhibit many local minima. Ideally, the optimization method should lead to the global optimum of the objective function with a minimum amount of computation. Classical optimization methods are generally fast and efficient, and have been found to work reasonably well for the design of digital filters. These methods are very good in locating local minima but unfortunately, they are not designed to discard inferior local solutions in favor of better ones. Therefore, they tend to locate minima in the locale of the initialization point [2, 3].

Genetic Algorithms (GAs) received considerable attention about their potentials as novel optimization technique for complex problems, especially for the problem with non-differentiable solution space. While these algorithms tend to require a large amount of computation, they also offer certain unique features with respect to classical gradient-based algorithms. For example, having located local suboptimal solutions, GAs can discard them in favor of more promising subsequent local solutions and, therefore, in the long run they are more likely to obtain better solutions for multimodal problems. GAs are also very flexible, non-problem specific, and robust. Furthermore, owing to their heuristic nature, arbitrary constraints can be imposed on the objective function without increasing the mathematical complexity of the problem. [4]

The particle swarm optimization (PSO) is a population-based stochastic optimization technique inspired by social behavior of bird flocking or fish schooling. The algorithm starts with a random initialization of a swarm of individuals, referred to as particles, within the problem search space. It then endeavors to find a global optimal solution by simply adjusting the trajectory of each individual toward its own best location visited so far and toward the best position of the entire swarm at each evolutionary optimization step. The attractions of the PSO method include its simplicity in implementation, ability to quickly converge to a reasonably good solution and its robustness against local minima. [5]

Therefore, some researchers have attempted to develop the designed methods based on modern global optimization algorithms such as particle swarm optimization(PSO), genetic algorithm(GA)[6,7,8,9],ant colony optimization(ASO) , simulated annealing[10], tabu search(TS)[11],etc. Revna et al. (2012) used evolutionary algorithms for optimum filter design and also evaluated its performance [12].

In this paper we have designed low-pass IIR filter using PSO and GA and the filter responses have been compared with the desired responses.

II. Proposed Algorithm

A. Genetic algorithm –

The working process of a basic GA is illustrated in the steps listed below:

Step 1: Represent the problem variable domain as a chromosome of fixed length.

Step 2: Let the size of the chromosome population N, the crossover probability Pc and the mutation probability Pm.

Step 3: Define a fitness function to measure the performance of an individual chromosome in the problem Domain .The fitness function establishes the basis for selecting chromosomes that will be mated during reproduction.

Step 4: Randomly generate an initial population of size N: $x_1, x_2, x_3, x_4, \ldots\ldots, x_n.$

Step 5: Calculate the fitness of each individual chromosome: $f(x_1), f(x_2), f(x_3), f(x_4), \ldots., f(x_n).$

Step 6: Select a pair of chromosomes for mating from the current population. Parent chromosomes are selected with a probability related to their fitness. High fit chromosomes have a higher probability of being selected for mating than less fit chromosomes.

Step 7: Create a pair of offspring chromosomes by applying the genetic operators.

Step 8: Place the created offspring chromosomes in the new population.

Step 9: Repeat Step 6 until the size of the new population equals that of initial population, N.

Step 10: Replace the initial (parent) chromosome population with the new (offspring) population.

Step 11: Go to Step 5, and repeat the process until the termination criterion is satisfied.

A GA is an iterative process. Each iteration is called a generation. A typical number of generations for a simple GA can range from 50 to over 500. A common practice is to terminate a GA after a specified number of generations and then examine the best chromosomes in the population. If no satisfactory solution is found, then the GA is restarted [4].

B. Particle Swarm Optimisation algorithm –

The working process of a basic PSO is illustrated in the steps listed below:

Step 1: Create a 'population' of agents (called particles) uniformly distributed over X.

Step 2: Evaluate each particle's position according to the objective function.

Step 3: If a particle's current position is better than its previous best position, update it.

Step 4: Determine the best particle (according to the particle's previous best positions).

Step 5: Update particles' velocities according to : $v_i^{t+1} = wv_i^t + c_1R_1(pbest_i^t - x_i^t) + c_2R_2(gbest_i^t - x_i^t).$

where x_i and v_i are position and velocity of particle i, respectively; $pbest_i$, $gbest_i$ is the position with the 'best' objective value found so far by particle *i* and the entire population respectively; *w* is a parameter controlling the flying dynamics; R1 and R2 are random variables in the range [0, 1]; c1 and c2 are factors controlling the related weighting of corresponding terms.

Step 6: Move particles to their new positions according to: $x_i^{t+1} = x_i^t + v_i^{t+1}$.After updating x_i should be checked and limited to the allowed range.

Step 7: Go to step 2 until stopping criteria are satisfied.

PSO is a metaheuristic as it makes few or no assumptions about the problem being optimized and can search very large spaces of candidate solutions. However, metaheuristics such as PSO do not guarantee an optimal solution is ever found. More specifically, PSO does not use the gradient of the problem being optimized, which means PSO does not require that the optimization problem be differentiable as is required by classic optimization methods such as gradient descent and quasi-Newton methods. PSO can therefore also be used on optimization problems that are partially irregular, noisy, change over time, etc.

III. DEFINITION OF THE PROBLEM

Consider the IIR filter with the input-output relationship governed by

$$y(k) + \sum_{i=1}^{M} b_i\, y(k-i) \sum_{i=0}^{L} a_i\, x(k-i) \ldots\ldots\ldots\ldots (1)$$

where $x(k)$ and $y(k)$ are the filter's input and output, respectively, and M ($\geq L$) is the filter order. The transfer

International Journal of Scientific Engineering and Technology (ISSN : 2277-1581)
Volume 2 Issue 5, pp : 398-403 1 May 2013

function of this IIR filter can be written in the following general form:

$$H(z) = \frac{A(z)}{B(z)} = \frac{\sum_{i=0}^{L} a_i z^{-i}}{1+\sum_{i=1}^{M} b_i z^{-i}} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (2)$$

The parameters for the optimal filter design that are considered are stopband and passband normalized frequencies the passband and stopband ripple the stopband attenuation and the transition width. These parameters are mainly decided by the filter coefficients. In any filter design problem, some of these parameters are fixed while others need to be determined. Hence, the design of these filters can be considered as an optimization problem of the cost function J (**w**) stated as follows:

$$\min_{w \epsilon W} J(w) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (4)$$

where $w = [a_0 a_1 \ldots. a_L\, b_1 \ldots. b_M]^T$ is the filter coefficient vector for the IIR filter.

The aim is to minimize the cost function J (**w**) for IIR by adjusting **w**. The cost function is usually expressed as the time averaged cost function defined by (6):

$$F = \frac{1}{N} \sum_{K-1}^{N} (ideal(k) - actual(k))^2 \ldots\ldots\ldots\ldots\ldots \quad (5)$$

where ideal (K) and actual (K) are the magnitude response of the ideal and the actual filter, where N is the order of the filter. The evolutionary approaches are hence applied in order to obtain the actual filter response as close as possible to the ideal response and filter coefficients are obtained.

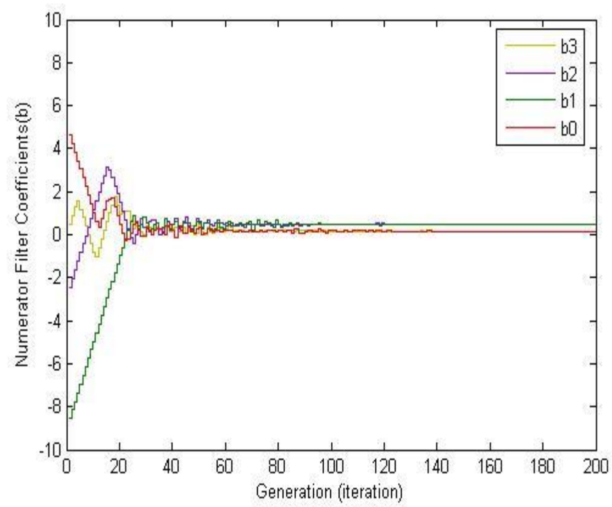
IV. SOLUTION METHODOLOGY

Various steps used for the designing of Digital Filters are as:
1. Check the response (Frequency response and Magnitude) with the Frequency difference equation of the IIR low pass filter.
2. Get the Coefficients (b, a) and assign it to a matrix that will be optimized. Secondly, optimization problem optimizes the magnitude using Particle Swarm Optimization and Genetic Algorithm by generating the objective function.
3. Design objective function based on the absolute value difference of frequency response between optimized coefficient and desired coefficient.
4. Fix the tolerance limits for objective and non linear constraint function to promote fast convergence.
5. Discretize and eliminate values that that are not free to vary.
6. Check the nearest integer values for better filter to be realized.
7. Plot the frequency response after optimization.

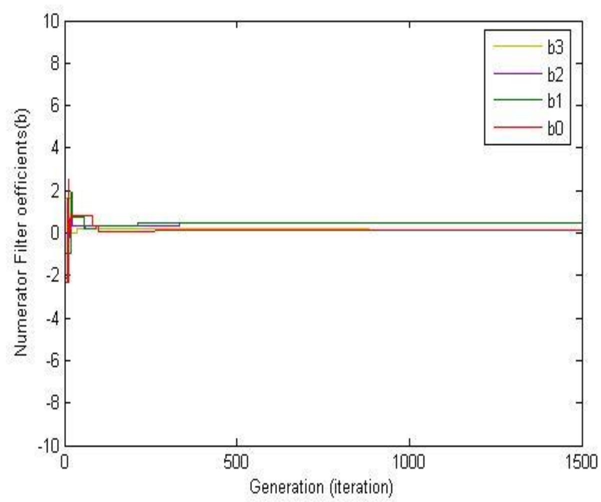
V. EXPERIMENT AND RESULTS

Simulations were conducted on a Intel(R) Atom(TM) CPU N270 @1.6 GHz(2 CPUs) computer, in the MATLAB 7 environment. Two problem statements described in APPENDIX were used for the identification task:

Simulation results of example 1:



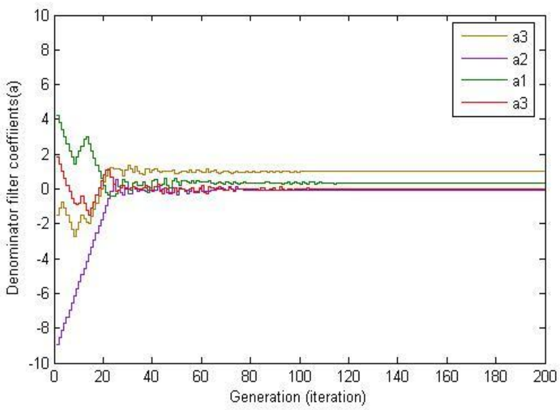
(a)



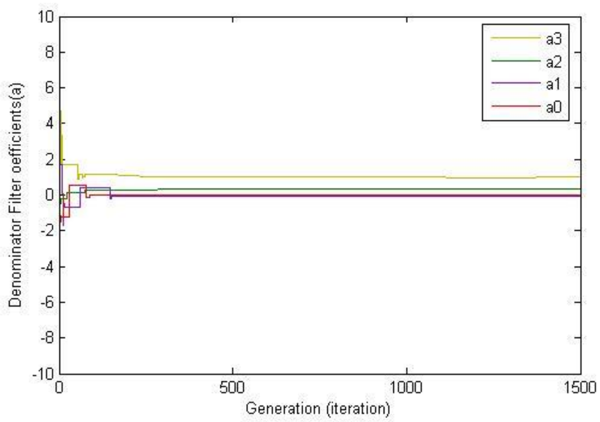
(b)

International Journal of Scientific Engineering and Technology (ISSN : 2277-1581)
Volume 2 Issue 5, pp : 398-403 1 May 2013

Figure 1: Evaluation of the numerator parameters of the LPF filter for both algorithms, (a) PSO and (b) GA

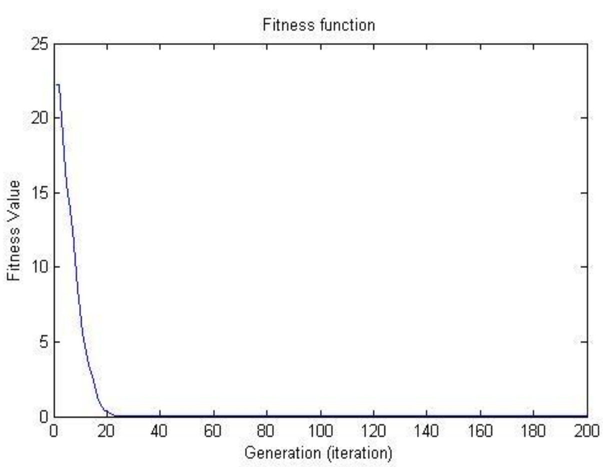


(a)

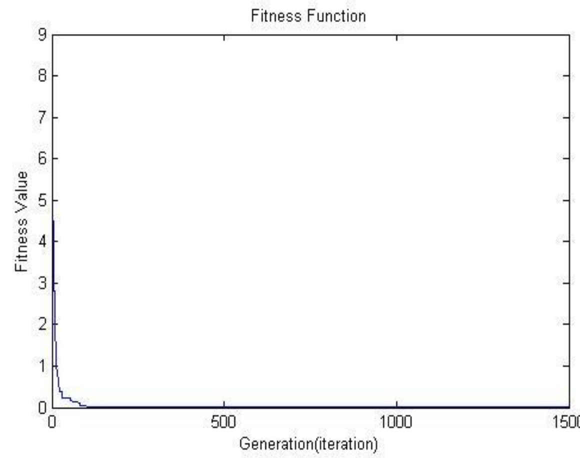


(b)

Figure 2: Evaluation of the numerator parameters of the LPF filter for algorithms, (a) PSO and (b) &(c) GA

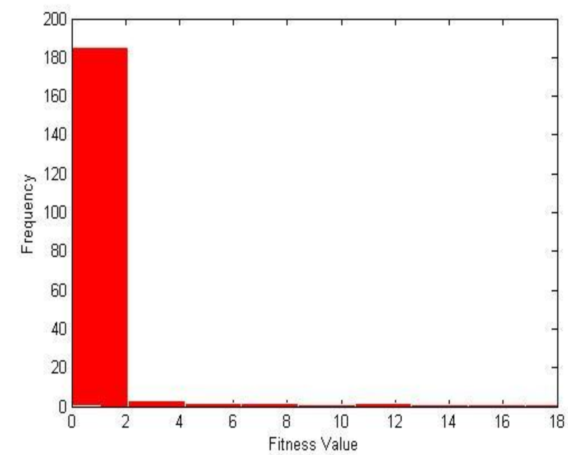


(a)

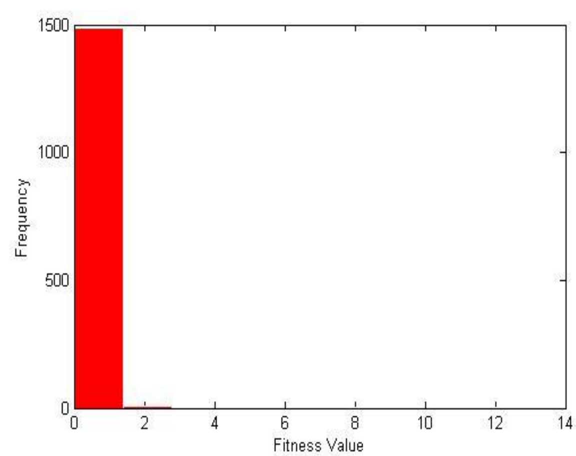


(b)

Figure 3: Improvement of average best solution by (a) PSO and (b) GA



(a)



(b)

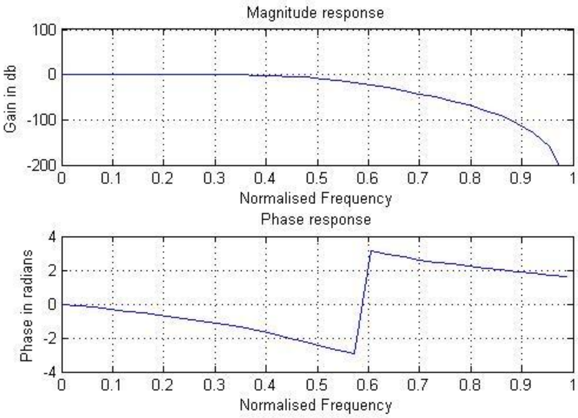
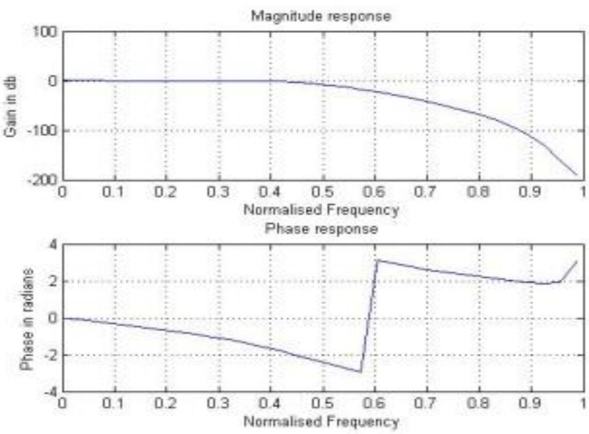
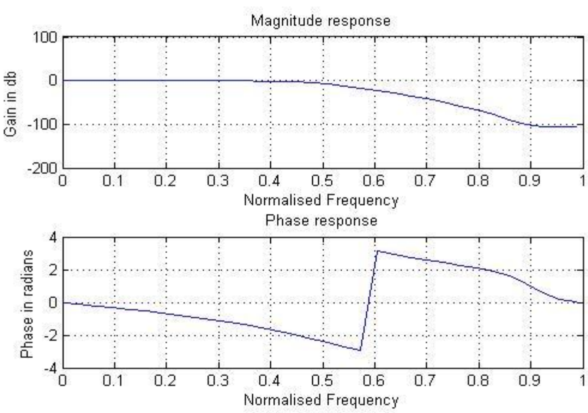


Figure 4: Histogram drawn for the results obtained for the fitness function by (a) PSO (b) GA

(a)

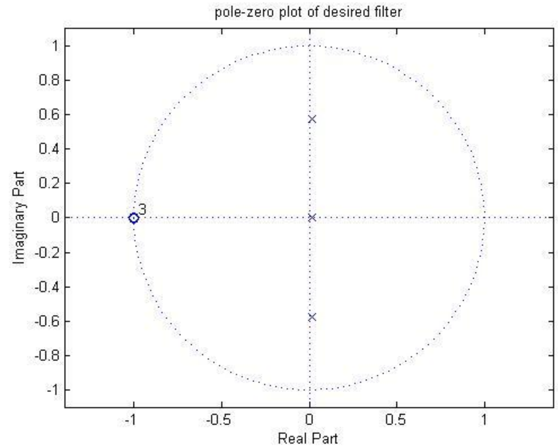


(b)

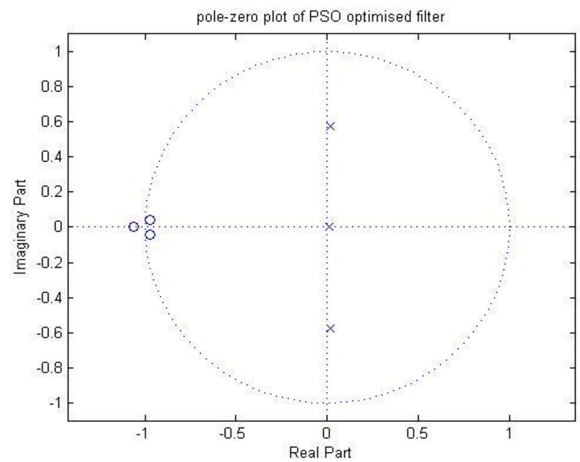


(c)

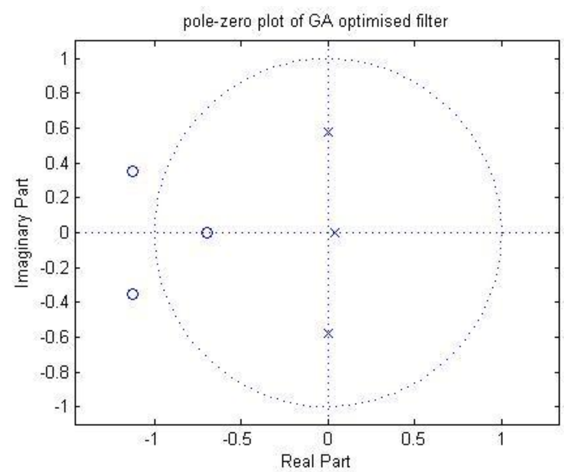
Figure 5: Magnitude and Phase Response of (a) desired filter, filter obtained using (b) PSO and (c) GA



(a)



(b)



(c)

Figure 6: Pole-Zero plot of (a) desired filter, filter obtained using (b) desired filter (b) PSO and (c) GA

Table-1 Optimised Coefficients of IIR Low-Pass Filter Designed with order=3

Coefficients	Desired Values Of Coefficients	Coefficients Obtained Using PSO	Coefficient s Obtained Using GA
a0	-0.0045	-0.0045	-0.014166
a1	0.3340	0.3340	0.3336
a2	-0.0494	-0.0494	-0.046885
a3	1.0000	1.0000	0.99403
b0	0.1600	0.1600	0.15907
b1	0.4800	0.4800	0.48562
b2	0.4800	0.4801	0.48383
b3	0.1600	0.1600	0.16425

From Table 1 it is clear that the results obtained by PSO are almost the same as of the desired coefficients as compared with GA.

VI. CONCLUSION

PSO algorithm is a new heuristic approach mainly having three advantages: finding true global minimum of a multimodal search, fast convergence, using a few control parameters. In this work, both PSO and GA algorithms were applied to digital filter design. From the simulations, it was observed that the performance of PSO algorithm in terms of convergence, speed and computation time required is better than that of GA.

VII. APPENDIX

Example 1: Design a digital low-pass IIR filter with following specifications: Pass/Stop band ripples 4dB/30dB and band edges 400Hz/800Hz and a sampling frequency of 2000Hz.

VIII. REFERENCES

i. *Andreas Antoniou, "Digital Filters analysis, design and application," Tata McGraw-hill edition, New Delhi, 2005.*
ii. *VinayK. Ingle and John G. Proakis," Digital Signal Processing using MATLAB," Thomson Books, New Delhi, 2004.*
iii. *A.P. Engelbrecht, Fundamentals of Computational Swarm Intelligence, John Wiley & Sons Ltd., New Delhi, 2005.*
iv. *Holland, J. "Adaptation in Natural and Artificial Systems". University of Michigan Press. 1975.*
v. *Kennedy, J. and R.C. Eberhart.." Particle Swarm Optimisation. Proceedings," IEEE International Conference on Neural Networks, IV,*
vi. *pp. 1942-1948, 1995.*
vii. *Joelle Skaf and P. Boyd Stephen, "Filter Design with Low Complexity Coefficients," IEEE Transactions on Signal processing, vol. 56, no. 7, p5. 3162-3170, July 2008.*
viii. *Richard J. Vaccaro and Brian F. Harrison, "Optimal Matrix-Filter Design," IEEE Transactions on Signal processing, vol. 44, no. 3, pp. 705-710, March 1996.*
ix. *Xi Zhang and Hiroshi Iwakura, "Design of IIR Digital Filters based on Eigen value Problem," IEEE Transactions on Signal processing, vol.44, no. 6, pp. 1325-1319, June 1996.*
x. *Fabrizio Argenti and Enrico Del Re, "Design of IIR Eigen filters in the Frequency domain," IEEE Transactions on Signal processing, vol. 46, no. 6, pp. 1694-1700, June 1998.*
xi. *Xin Yao, Yong Liu, and Guangming Lin, "Evolutionary Programming Made Faster," IEEE Transaction on Evolutionary Computation, vol. 3, no. 2, pp. 83-102, July 1999.*
xii. *T.William and W.C. Miller, "Genetic algorithms for the design of Digital filters," Proceedings of IEEE ICSP'O4, p. 9-12, 2004.*
xiii. *Revna Acar Varul, "Performance Evaluation of Evolutionary Algorithms for Optimal Filter Design,"IEEE Transactions On Evaluationary Computation, vol .16, No.1, February 2012.*