VOL. 4, NO. 4, JUNE 2009 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

# THE ROBUST DESIGN OF LINEAR PHASE FIR FILTER USING MIX-MUTATION EVOLUTIONARY PROGRAMMING

S. Nagaraja Rao[1], M. N. Giri Prasad[2] and Manoj Kumar Singh[3]
[1]Department of ECE, G. Pullaiah College of Engineering and Technology, Kurnool, A. P., India
[2]Department of ECE, J. N. T. U. College of Engineering, Pulivendula, A. P., India
[3]Advance Technology Consultant, Manuro Technology Research Labs, Bangalore, India
E-Mail: suryakari2k1@yahoo.com

**ABSTRACT**

In the design of frequency-selective filters, the desired filter characteristics are specified in the frequency domain in terms of desired magnitude and phase response of the filter. In this paper we present a design approach by determining the closely approximated coefficients using powerful Evolutionary Programming to find the solution for the optimization problem in selecting the coefficients. In this paper the design of Causal FIR filter with desired frequency response and phase response is presented. In practice, FIR filters are employed in filtering problems where there is a requirement for linear phase characteristics within the passband of the filter. The Evolutionary Programming is the best search procedure and most powerful than Linear Programming in providing the optimal solution that is desired to minimize the ripple content in both passband and stopband. We presented here how the values of $\delta_1$ and $\delta_2$ are minimized with best optimized approach using Evolutionary Computation. The optimized filter bank structure is implemented in our research work for effective compression of images.

**Keywords**: Finite Impulse Response filter, Evolutionary programming, digital signal processing, image processing, image compression.

## 1. INTRODUCTION

### 1.1 Linear phase FIR filter

Most finite impulse responses (FIRs) are linear-phase filters; when a linear-phase filter is desired, a FIR is usually used. "Linear Phase" refers to the condition where the phase response of the filter is a linear (straight-line) function of frequency (excluding phase wraps at +/- 180 degrees). This results in the *delay* through the filter being the same at all frequencies. Therefore, the filter does not cause "phase distortion" or "delay distortion". The lack of phase/delay distortion can be a critical advantage of FIR filters over IIR and analog filters in certain systems, for example, in digital data modems. Finite impulse response (FIR) filters are usually designed to be linear-phase (but they don't have to be.) A FIR filter is linear-phase if (and only if) its coefficients are symmetrical around the center coefficient, that is, the first coefficient is the same as the last; the second is the same as the next-to-last, etc. (A linear-phase FIR filter having an odd number of coefficients will have a single coefficient in the center which has no mate.) The formula is simple: given a FIR filter which has N taps, the delay is: (N - 1) / (2 * Fs), where Fs is the sampling frequency. So, for example, a 21 tap linear-phase FIR filter operating at a 1 kHz rate has delay: (21 - 1) / (2 * 1 kHz)=10 milliseconds. Non-linear phase, of course.) Actually, the most popular alternative is "minimum phase". Minimum-phase filters (which might better be called "minimum delay" filters) have less delay than linear-phase filters with the same amplitude response, at the cost of a non-linear phase characteristic, which leads to. "Phase distortion." A low pass FIR filter has its largest-magnitude coefficients in the center of the impulse response. In comparison, the largest-magnitude coefficients of a minimum-phase filter are nearer to the beginning.

For an N-tap FIR filter with coefficients h (k), whose output is described by?

$$Y(n) = h(0)\,x(n) + h(1)\,x(n-1) + h(2)\,x(n-2) + \dots h(N-1)\,x(n-N-1) \quad \dots\dots (1)$$

The filter's Z transform is:

$$H(z) = h(0)\,z^{-0} + h(1)\,z^{-1} + h(2)\,z^{-2} + \dots h(N-1)\,z^{-(N-1)} \quad \dots\dots (2)$$

or

$$H(z) = \sum_{n=0}^{N-1} h(n) z^{-n}$$

The variable z in H (z) is a continuous complex variable, and we can describe it as: $z = r\cdot e^{j\omega}$, where r is a magnitude and ω is the angle of z. If we let r = 1, then H (z) around the unit circle becomes the filter's frequency response H (jω). This means that substituting $e^{j\omega}$ for z in H (z) gives us an expression for the filter's frequency response H (ω), which is:

$$H(j\omega) = h(0)\,e^{-j0\omega} + h(1)\,e^{-j1\omega} + h(2)\,e^{-j2\omega} + \dots h(N-1)\,e^{-j(N-1)\omega}$$, or

Using Euler's identity, $e^{-ja} = \cos(a) - j\sin(a)$, we can write H (ω) in rectangular form as:

$$H(j\omega) = h(0)[\cos(0\omega) - j\sin(0\omega)] + h(1)[\cos(1\omega) - j\sin(1\omega)] + \dots h(N-1)[\cos((N-1)\omega) - j\sin((N-1)\omega)]$$, or

$$H(j\omega) = \sum_{n=0}^{N-1} h(n)[\cos(n\omega) - j\sin(n\omega)] \quad \dots\dots (3)$$

VOL. 4, NO. 4, JUNE 2009 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

Again, the key is the lack of feedback. The numeric errors that occur when implementing FIR filters in computer arithmetic occur separately with each calculation; the FIR doesn't "remember" its past numeric errors. In contrast, the feedback aspect of IIR filters can cause numeric errors to compound with each calculation, as numeric errors are fed back.

The practical impact of this is that FIRs can generally be implemented using fewer bits of precision than IIRs. For example, FIRs can usually be implemented with 16 bits, but IIRs generally require 32 bits, or even more.

**1.2 Evolutionary programming**

Evolution is a two-step population based process of random variation and selection. In an algorithm, this can be captured by creating a collection of potential solutions to a problem and using random numbers from a chosen distribution to generate new solutions. A selection criterion is imposed to determine which solution should be kept and which should be discarded. This is a very general generate and test routine, however the effectiveness of the procedure depends in large part on the choices of parameters and operators such as the number of parent' solutions, the population size, the type and form of random variation and so forth. The typical flow chart of Evolutionary algorithm is shown in Figure-1.

**Typical flowchart of an evolutionary algorithm**

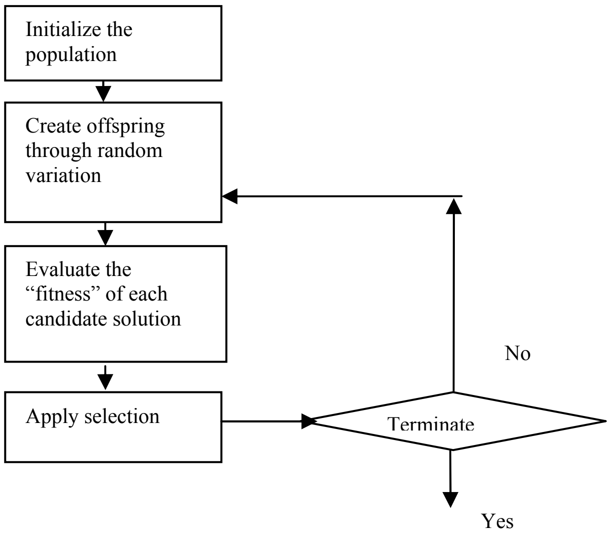


**Figure-1**. Flow chart of an evolutionary programming.

The process starts with a population of candidate solutions to the task at hand. These may be sampled randomly or provided as hints from previous experience or other algorithms. New solutions are created by applying random variation to the existing solutions. This variation can come in the form of single parent or multi-parent operators. Alternative choices offer different sampling distributions from the space of all possible solutions. Each of the individuals in the population are scared with respect to how well they accomplished the task at hand ('fitness') and selection is used to eliminate same subset of the population or to amplify the percentage of above average solution. The algorithm terminates when some extrinsic criterion has been satisfied, such as prescribed maximum number of generations or a suitable error tolerance. The procedure may be written as the difference equation:

$$X(t+1) = s(v(X[t])) \quad \ldots\ldots (4)$$

Methods of evolutionary computation such as evolutionary programming, evolution strategies and genetic algorithms are often used to address general forms of the problem: Find $x \in R^n$ such that $f(x): R^n \rightarrow R$ is minimized. Each of these techniques maintain a population of a candidate solutions to the task at hand, impose random variations on these solutions and applies a selection criterion to determine which solutions should propagate into future generations. The difference between the methods derived from the philosophical foundation of how each simulates evolution. Genetic algorithms abstract evolution as a process of adaptive genetics, traditionally operating on binary encoding of the parameters to be optimized, and using genetic operators based on observed genetic mechanism. Evolution strategies and evolution programming, abstract evolution as a process of adaptive behaviors, emphasizing carefully constructed mutation operations so that parents generate offspring with suitable functionality, regardless of any particular characteristic of their underlying coding structure.

The advantage with evolutionary programming and evolution strategies over genetic algorithm is these algorithms directly operate on the real values to be optimized in contrast with genetic algorithm which usually operate on a separately coded transform of the objective variables.

### 1.2.1 Evolutionary programming (EP) for real valued function optimization

The basic algorithm proceeds as follows:

a) A population of μ parent solutions $X_i$, i = 1, 2……. μ is initialized over a region M є $R^n$.
b) Each parent solution $X_i$, i = 1, 2… μ is scored in light of the objective function f (X).
c) Each parent creates offspring X'i, i = 1, 2… μ where the $j^{th}$ component of $X'_i$ is determined by $x'_{ij} = x_{ij}$ + N (0, σj), j = 1, 2… n
where N (0, σj) is a zero mean Gaussian random variable with standard deviation $\sigma_j$.
d) Each offspring solution $X'_i$, i = 1, 2… μ is scored in light of the objective function f (X).
e) Each solution $X_0$ and $X'_i$, i = 1, 2……. μ is evaluated against to other randomly chosen solution from the population. For each comparison a 'win' is assigned to the solution's score is less than or equal to that of its opponent.
f) The μ solutions with the greatest number of wins are retained to be parents of the next generation.
g) If the available computing time or defined terminating criteria is met, halt, otherwise proceed to step (c).

## 2. APPLICATION OF EP TO FIR

Apart from getting the optimal solution the other advantage to filter design through EP is that a variety of additional constraints can be utilized. For example, the response at a given frequency can be completely nullified or a constraint can be added to limit the amount of overshoot in the impulse response. Filter is also designed to minimize deviation from a given spectral prototype.

### 2.1 General constrained optimization

The general constrained optimization problem is defined as Minimize $f\left(\bar{x}\right)$ subject to constraints

$$g_1\left(\bar{x}\right) \le 0,......., \; g_r\left(\bar{x}\right) \le 0 \quad .... (5)$$

$$h_1\left(\bar{x}\right) = 0,......., \; h_m\left(\bar{x}\right) = 0 \quad .... (6)$$

Where $f$ and $g_i$'s are functions on $R^n$ and $h_j$'s are functions on $R^n$ for m ≤ n.

Let the coefficient of a linear phase FIR filter be Gn, N = $\frac{-(N-1)}{2}, \frac{-(N-1)}{2}+1, ..... \frac{(N-1)}{2}$ with N odd and $g_n = g_{-n}$ (with suitable modification, the result can be extended to filters with an even number of coefficients and defining $h_n = g_n - (N-1)/2$ results in a causal filter with the same magnitude response).

The frequency response of the filter is

$$G(e^{i\omega}) = g_0 + 2\sum_{n=1}^{(N-1)/2} g_n \cos \omega_n \cdot \quad .... (7)$$

We want to design a low pass filter with passband frequency $\omega_P$ and stopband frequency $\omega_S$ as shown in Figure-1. The ripple in the passband is $\pm\delta_1$ and the ripple in stopband is $\pm\delta_2$. There are various design possibilities and in all the cases in the design of the filter, we desire to choose the filter coefficients in association with $\delta_1$ and $\delta_2$ such that the defined constraints are to be satisfied.

$$0 \le \omega \le 2\pi F_p : \begin{cases} g_0 + 2\sum_{n=1}^{(N-1)/2} g_n \cos\omega_n \le 1+\delta_1 \\ -g_0 - 2\sum_{n=1}^{(N-1)/2} g_n \cos\omega_n \le -1+\delta_1 \end{cases} \quad .... (8)$$

$$2\pi F_s \le \omega \le \pi : \begin{cases} g_0 + 2\sum_{n=1}^{(N-1)/2} g_n \cos\omega_n - \delta_2 \le 0 \\ -g_0 - 2\sum_{n=1}^{(N-1)/2} g_n \cos\omega_n - \delta_2 \le 0 \end{cases}$$

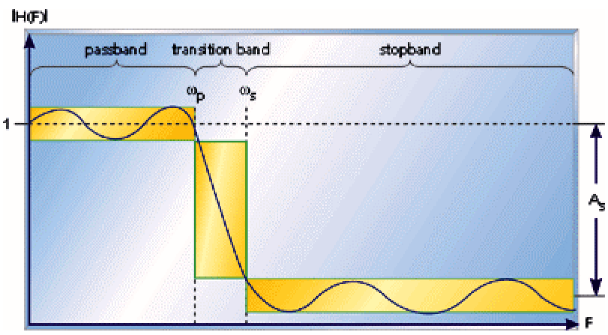


**Figure-2**. Magnitude response of a linear phase FIR filter.

If number of coefficients are known, the objective could be to find such a set of coefficients so that $\delta_1$ and $\delta_2$ can be minimal or if $\delta_1$ is specified then $\delta_2$ could be minimal.

### 2.2 Cauchy-Gaussian distribution

The one dimensional Cauchy density function centered at origin is defined by $f_t(x) = \frac{1}{\pi}\frac{t}{t^2 + x^2}$ $-\infty < x < \infty$ where $t > 0$ is a scalar parameter.

The corresponding distribution function is $F_1(x) = \frac{1}{2} + \frac{1}{\pi}\arctan\left(\frac{x}{t}\right)$ the shape of $f_t(x)$ resembles that of the Gaussian density function but approaches the axis so slowly that an expectation does not exist. As a result, the variance of Cauchy distribution is infinite.

It is known that large jumps in evolutionary search are beneficial. The comparison of Cauchy and Gaussian density function is shown in Figure-3.

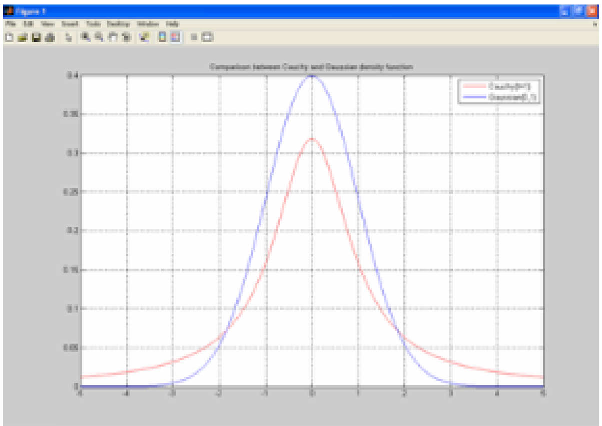

**Figure-3**. Cauchy- Gaussian density function.

### 2.3 Algorithmic approach

**Step 1**: A population of N trial solutions was initialized. Each solution was taken as a pair of real-valued vectors $(\bar{x}_i, \bar{\sigma}_i)$, $\forall i \in \{1,2,...,N\}$ with their dimensions corresponding to the number of variables, n to be optimized.

The initial components of each $\bar{x}_i, \forall i \in \{1,2,...,N\}$ were selected in accordance with a uniform distribution ranging over a presumed solution space.

The value of $\bar{\sigma}_i, \forall i \in \{1,2,...,N\}$, the so called strategy parameters were initialized with some value.

**Step 2**: The fitness score for each solution $\bar{x}_i$ was evaluated in light of an objective function

$$\phi(x_i),\ where\ \phi(x_i) = \rho f(\bar{x}_i) + s\left( \sum_{k=1}^{r} (g_k^{+}(\bar{x}_i))^2 \right)$$

where $\rho$ and $s$ are tunable parameters.

**Step 3**: Two offspring were generated from each parent by

$$\bar{x}_i'(j) = \bar{x}_i(j) + \bar{\sigma}_i(j).C_j$$

$$\bar{x}_i''(j) = \bar{x}_i(j) + \bar{\sigma}_i(j).N_j(0,1)$$

$$\bar{\sigma}_i'(j) = \bar{\sigma}_i(j).\exp(\tau' N(0,1) + \tau N_j(0,1))$$

$\forall j \in \{1,2,...,r\}$ Where $\bar{x}_i(j), \bar{x}_i'(j), \bar{x}_i''(j), \bar{\sigma}_i(j)$ and $\bar{\sigma}_i'(j)$ denote the j[th] component of vectors $\bar{x}_i, \bar{x}_i', \bar{x}_i'', \bar{\sigma}_i$ and $\bar{\sigma}_i'$ respectively. $C_j$ Is Cauchy random variable with scale parameter t = j.

$N(0,1)$ denotes a standard Guassian random variable. $N_j(0,1)$ Indicates that the random variable is sampled for each new value of the counter *j*. The scaling factors $\tau$ and $\tau'$ is robust exogenous parameters which have been set to $\left(\sqrt{2\sqrt{n}}\right)^{-1}$ and $\left(\sqrt{2n}\right)^{-1}$ following Back and Schwefel (1993).

**Step 4**: The fitness score for each offspring $\phi(\bar{x}_i)$ was determined.

**Step 5**: The pair-wise comparisons over all the 3N solutions were made. For each solution 10% of population is selected randomly as opponents from among all parents and offspring with equal probability. In each comparison, if the conditioned solution offers at least as good performance as the randomly selected opponent, it receives a "win".

**Step 6**: The N best solutions out of 3N based on number of wins received were selected to be the parents for the subsequent generation.

**Step 7**: The algorithm will be proceeded to step 3 unless the available time for execution was exhausted or acceptable solution had been discovered.

The Evolutionary Programming procedure was halted with the following condition satisfied. For each best solution at generation k, $\bar{x}_1^{(k)}$ and generation k-1, $\bar{x}_1^{(k-1)}$ if $\left| x_1^{(k)}(j) - \bar{x}_1^{(k-1)}(j) \right| \le \rho$ for all *j* and for successive $N_g$ generation then the procedure was halted.

VOL. 4, NO. 4, JUNE 2009 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

**Offspring generation**

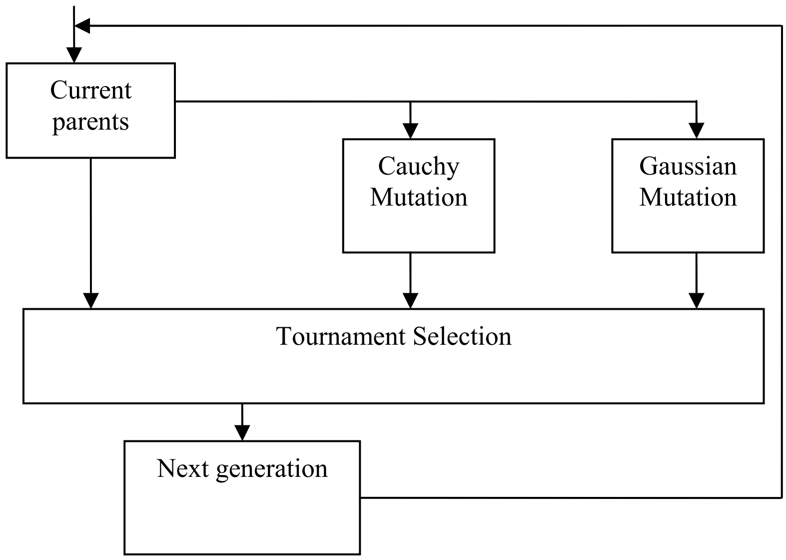


**Figure-4**. Block schematic represents the generation of offspring.

## 3. TEST RESULTS

The test results are being presented in various plots obtained to understand the effective utilization of powerful Evolutionary Programming Technique for the design of a robust linear phase FIR Filter structure.

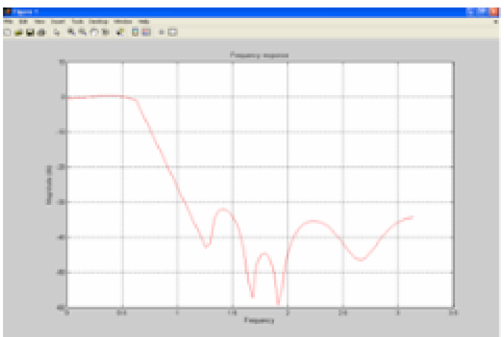

**Figure-5.** Frequency response of the filter.

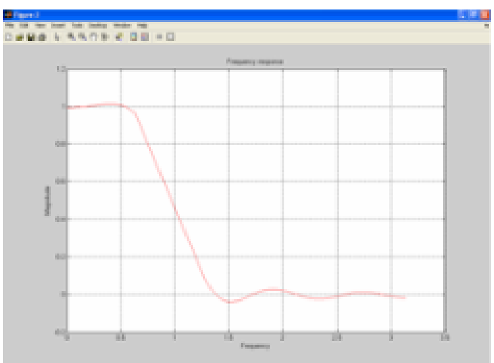

**Figure-6**. Frequency response after mutation.

The following shows the experimental evaluation for the frequency response shown in Figure-6 after mutation.

Fv = -0.0000 0.0000 0.3156 0.2633 0.1351 0.0140 -0.0445-0.0388-0.0041 0.0144

$h_{f=}$ 0.2633 0.1351 0.0140 -0.0445-0.0388-0.0041 0.0144

$n_c$ = 7

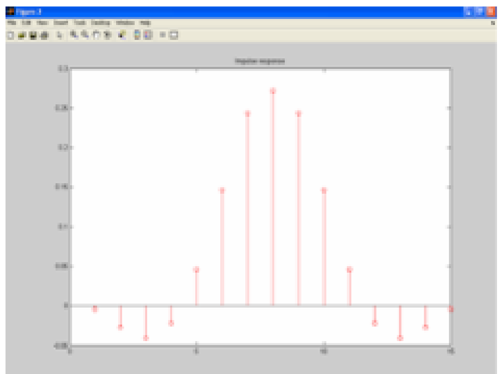

**Figure-7**. Impulse response of fir filters.

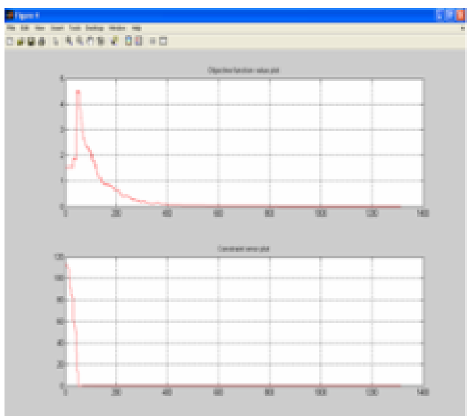

**Figure-8**. Objective function value and constraint error value.

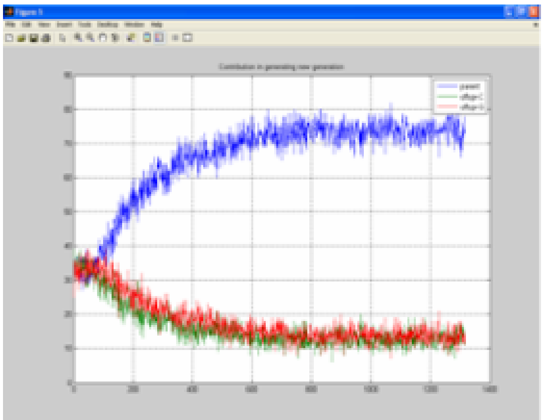

**Figure-9**. Contribution in generating the new generation by cauchy and gaussian distributions.

The general approach in the constraint optimization problem is to define the objective function as a combination of function to be optimized along with constraints. These constraints generally define the solution region. Once the solution region is defined and it is to find out the minimum value of function. In most of the cases variables involve in the function also participate in the definition of constraints. Hence, the penalty method is able to find the minimum value of function. In the problem under consideration for FIR Filter, the constraints are involving the coefficients along with the ripple band values $\delta_1$ and $\delta_2$ .

But the function which is to be minimized involves only $\delta_1$ and $\delta_2$ . Hence the combination of function and constraints is to be associated with certain tunable parameters. It would be a better if more weightage is given to the function parameter compared to constraint tunable parameter, s. It is well known that Cauchy distribution takes longer jumps compared to Gaussian distribution. Many researchers have given the idea of application of Cauchy distribution for mutation purpose to enhance the speed of convergence. But at the same time there is a precaution about the deviation from the global solution as the progress is moving towards that. The other approach many researchers have taken is to switch form Cauchy to Gaussian distribution with the program of iteration. The reason behind that is Gaussian distribution can take smaller steps and this can be useful to find the global solution.

## 4. CONCLUSIONS

With the empirical results we have observed that irrespective of generation there is always nearly same contribution from Cauchy as well as Gaussian distribution. This is very surprising and against all the theoretical descriptions established so far. Research must be started to find out the existing phenomenon. From the results it is also very clear that Cauchy and Gaussian distributions are not sufficient to generate the filter offspring using mutation when taking these distributions separately. Here, we have taken the new approach to develop population by combining the existing parent offspring of Cauchy distribution along with offspring of Gaussian distribution to find out the next generation.

## REFERENCES

[1] D.B.Fogel. 1998. Evolutionary computation. The Fossil record. IEEE Press, Piscataway, N.J.

[2] D.Whitley, S.Rana, J.Dzubora and E.Mathios. 1996. Evolutionary algorithms. Artificial Intelligence. 88: 245-276.

[3] I.De.Falco, Della eiappa and E.Tarantino. 2002. Mutation based genetic algorithm: performance evaluation applied soft computing. 1: 285 -299.

[4] J.He, X.H.Yu. 2001. Conditions for the convergence of evolutionary algorithms. Journal of system architecture. 47(7): 601-612.

[5] L.Daris. 1991. Handbook of Genetic algorithms. Van Nostrand Reinhold, NY.

[6] X.C.Jhao, X.S.Gao. 2004. A micro evolutionary programming for optimization of continuous space. PPSN VIII workshop on challenges in real world optimization using evolutionary computing. pp. 17-23.

[7] Ysuo Takeshisa, Hidenori Sakanashi, Tetsuya Higuchi. 2000. Adaptive wavelet transforms for lossless compression using genetic algorithms. Genetic and Evolutionary computation conference.

[8] S.Nagaraja Rao, Dr.M.N.GiriPrasad. 2008. Evolutionary computation, the high power computing solution to optimization problems. Conference on algorithms. 2: 83-91.

[9] B. I. Hounsell, T. Arslan, R. Thomson. 2004. Evolutionary design and adaptation of high performance digital filters within an embedded reconfigurable fault tolerant hardware platform. Soft Computing. 307-317, Springer-Verlag.

[10] A. Antoniou. 2005. Digital Signal Processing: Signals, Systems, and Filters. New Jersey: McGraw-Hill.

[11] O. Franzen, H. Blume, and H. Schroder. 1998. FIR Filter Design with Spatial and Frequency Design Constraints Using Evolution Strategies. Signal Processing. 68: 295-306, August.

[12] A. Zhigljavsky and A. ilinskas. 2008. Stochastic Global Optimization. New York: Springer.

[13] R. Nambiar, C. K. Tang, and P. Mars. 1981. Genetic and learning automata algorithms for adaptive Filter design. In: Proceedings of the IEEE International

Conference on Acoustics, Speech and Signal Processing. Vol. 50, Atlanta, GA, March. pp. 1253-1256.

[14] N. E. Mastorakis, I. F. Gonos, and M. S. Swamy. 2003. Design of two-dimensional recursive filters using genetic algorithms. IEEE Trans. on Circuits and Systems. 50(5): 634-639, May.

[15] P. Gentili, F. Piazza, and A. Uncini. 1995. Efficient genetic algorithm design for power-of-two FIR Filters. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing. Vol. 2, Detroit, MI, May. pp. 1268-1271.

[16] M. Yagyu, A. Nishihara, and N. Fujii. 1996. Fast FIR digital Filter structures using minimal number of adders and its application to Filter design. IEICE Trans on Fundamentals. Vol. E79-A. 8: 1120-1128, August.

[17] Y. C. Lim and S. R. Parker. 1993. Discrete coefficient FIR digital Filter design based upon LMS criteria. IEEE Trans on Circuits Systems. Vol. CAS-30. pp. 723-739, October.

[18] Y. Leung and Y. Wang. 2001. An orthogonal genetic algorithm with quantization for global numerical optimization. IEEE Trans on Evolutionary Computation. 5(1): 41-53, February.

[19] Y. C. Lim. 1990. Design of discrete-coefficient-value linear phase FIR Filters with optimum normalized peak ripple magnitude. IEEE Trans. on Circuits Systems. 37(12): 1480-1486, December.

[20] M. C. Lang. 1995. An iterative reweighted least-squares algorithm for constraint design on nonlinear phase FIR filters. In: Proceedings of the IEEE International Symposium Circuits. SystemsSeattle, WA, May. 5: 367-370.

[21] J. W. Adams. 1991. FIR digital Filters with least-squares stopbands subject to peak- gain constraints. IEEE Trans. on Circuits Systems. 39(4): 376-388, April.

[22] W. -S. Lu and A. 2000. Antoniou Design of digital Filters and Filter banks by optimization: a state of the art review in Proceedings of European Signal Processing Conference on Communications. Vol. 1, Tampere, Finland, September. pp. 351-354.

[23] M. Lang. 1999. Algorithms for the Constrained Design of Digital Filters with Arbitrary Magnitude and Phase Responses. Vienna University of Technology, Institute for Communications and Radio-Frequency Engineering, Bibliography 140 Ph.D. Thesis CSD-TR No. 030056. Available online: http: /www.nt.tuwien.ac.at/fileadmin/users/gerhard/diss_Lang.pdf.

[24] L. M. Smith. 1998. Decomposition of FIR digital Filters for realization via the cascade connections of subfilters. IEEE Trans. on Singal Processing. 46(6): 1681-1684, June.

[25] M. K. Sai and B.Nowrouzian. 2007. A diversity controlled genetic algorithm for optimization of FRM digital filters over DBNS multiplier coefficient space. In: Proceedings of the IEEE International Symposium on Circuits and Systems, New Orleans, Louisiana. May. pp. 2331-2334.

[26] L. Cen and Y. Lian. 2005. A hybrid GA for the design of multiplication-free frequency response masking Filters. In: Proceedings of the IEEE International Symposium on Circuits and Systems. Vol. 1, Salt Lake City, Utah. May. pp. 520-523.

[27] W. R. Lee, L. Caccetta, K. L. Teo. And V. Rehbock. 2006. A weighted least squares approach to the design of FIR filters synthesized using the modified frequency-response masking structure. IEEE Trans. on Circuits and Systems II. 53(5): 379-383, May.