AN OPTIMIZATION OF DIGITAL FILTER SYSTEM USING ADAPTIVE GENETIC ALGORITHM

Vijay Yadav[1], Dr. Manoj Shukla[2]
[1]Research Scholar (ECE), [2]Supervisor, Mewar University, Chhitorgarh

Abstract: Optimization of different methods of IIR filter and compare their results are presented. A Genetic Algorithm (GA) that operates on a complex, continuous search space is constructed and optimized by statistically determining the abilities of commonly utilized genetic operators. The specific new genetic operators are introduced & its desire crossover and adaptive mutation which enhance the convergence rate with solution quality of the GA. The customized application layer, called the Adaptive Genetic Algorithm (AGA), has been developed for optimize the magnitude response, pole-zero diagram, and fitness of the Adaptive filter which is utilize the system and handle the specific formative operation and specific properties in the filter design problem.

Keywords: Adaptive Genetic Algorithm (AGA), Continuous Genetic Algorithm (CGA), Filter Design Algorithm (FDA), Butterworth Bandpass Filter (BBF), Chebyshev filter, Discrete Genetic Algorithm (DGA), Infinite Impulse Response (IIR) filters, Finite Impulse Response Filter (FIR), Breeder Genetic Algorithm (BGA), Least Mean Square (LMS).

I. INTRODUCTION

As digital filters offer features that have no counterparts in other filter technologies, that's why, digital filters can be applied to an almost endless variety of applications, including signal manipulation, noise cancellation, and channel equalization. Therefore, efficient design Strategies are necessary to resolve the digital filter transfer function needed for useful implementation. Due to instability problem, magnitude and phase response for filters occurring during the design process of digital filters. The GA is a framework to underneath the actual algorithm required for optimization and it must be customized for a given application of the filter system [4]. The final algorithm presented for optimizing continuous and discrete problems is called the Adaptive Genetic Algorithm (AGA). Here we are identified different objects for low-pass, high-pass and band-pass filters such as coefficients of filter, order of filter, multi-objects, efficiency, effectiveness, comparing results obtained from different techniques, pole-zero plots and responses. This paper represents the designing of filters with different approaches and their comparison with other to find the better most technique.

II. PREVIOUS METHODOLOGY

There are various methods for digital IIR filter design have been implemented and design. Because of multi-modal mean square error function, IIR design is better than finite impulse response (FIR) design. To eliminate this, a global least mean square algorithm has been proposed to allow a global minimum search. This algorithm, similar to simulated annealing, is called stochastic approximation with convolution smoothing (SAS) and is realized by convolving the objective function with a noise probability density function. When we combined with the least mean square (LMS) algorithm, it has shown success in adaptive IIR filtering [6]. Another approach for adaptive IIR filter design is also an extension of the LMS algorithm. The LMS algorithm is applied to multiple filters of different initial conditions to help reduce the probability of convergence to a local minimum. Whenever the rate of convergence slows or a filter becomes unstable, an embedded evolutionary computation is utilized to move the previous filter coefficients. This approach benefits from the directed search of the LMS algorithm and the ability to recover from instability [7]. A GA is utilized because of its implicit parallelism and robustness. Simulated results show improvements in global optimization compared to the LMS algorithm, pure SA, and pure GA, but the proposed technique has yet be actually applied to the IIR design problem for investigation of abilities [8]. The main differences between them lie in the customization of the algorithm for use in the IIR design problem. The method of generating a sample population, performing crossover and mutation, and evaluating the results are all tailored to this goal [9]. The coefficient symmetry gives us a linear phase even in the case of an IIR filter [10].

III. DISCRETE GENETIC ALGORITHM

The Discrete Genetic Algorithm (DGA) operates on a discrete set of chromosomes where, for sake of simplicity, the chromosomes are generally considered to be binary encoded as single-bit genes. There are three operator of discrete genetic algorithm.

Selection

The main purpose of selection is to drive P (g) towards the most promising areas of S while still maintaining enough variation in P(g) to prevent premature solution convergence. To comply with the accepted notion of survival of the fittest, any selection strategy chosen for the DGA must have a higher probability ps of selecting the more fit elements of P(g) to form the selection subset Ps(g).The three most common methods of selection are proportionate, rank-based, and truncation selection.

Crossover

The crossover genetic operator redistributes genetic material within P(g). It is generally the first genetic operator applied after selection and is often considered the most important genetic operator in the DGA. Crossover is executed by combining or mixing two elements from Ps with probability pc to form one or more offspring that encapsulate the combined genetic material. The goal is to generate new elements that are fit than their parents, thereby contributing to the overall fitness and convergence of the population. Three most common methods of crossover are single point, double point and uniform crossover.

Mutation

During the optimization and convergence process it is sometimes necessary to remove desirable genetic material from the population to overcome local optima of the system. Furthermore, there is no guarantee that all necessary optimal genetic information appears in the population at a given generation. Therefore, the mutation operator has been introduced to create new or lost genetic material into the population. The new mutation operation is performed by probabilistically modifying vector values of offspring elements. This enables exploration into areas of the search space otherwise be reached through crossover of the current set of elements. The three most common methods of mutation are uniform, normal and BGA.

Figure 1, 2 and 3 are shows that, the normal crossover offspring vector distribution, Uniform mutation joint PDF and Normal mutation joint PDF.

IV. PROPOSED ALGORITHM

Adaptive Genetic Algorithm is designed for both Discrete and Continuous parameter of the IIR Filter System. While the Adaptive Genetic Algorithm is a generic optimization algorithm, it still needs to be modified to suit specific applications, such as digital IIR filter design. For this purpose, a customized application layer has been developed for handle the specific format and properties of the filter design problem. This algorithm include to mapping a filter to an element, evaluating the fitness function of a filter, creating an initial population of filters, and ensuring that all digital filters are optimized. Digital filters are design and optimized for magnitude, phase responses, Pole-Zero and fitness such as a Butterworth bandpass filter and Chebyshev filter.

Design algorithm steps

The complete flow chart of the AGA is shown in Figure 4. The new hybrid operator, normal crossover, is classified as a crossover method that can be combined with an additional mutation method.

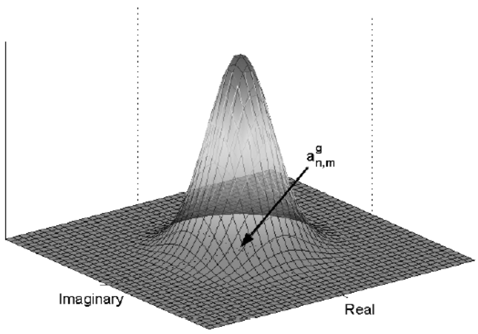


Figure 1: Normal crossover offspring vector distribution

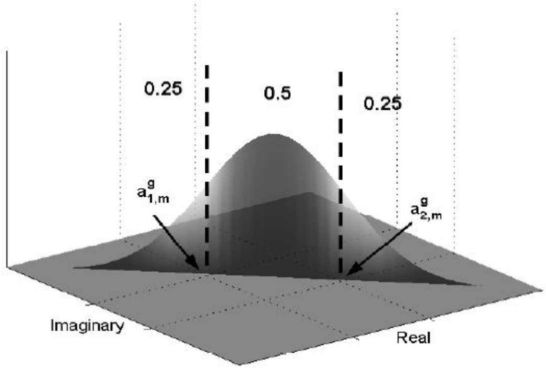


Figure 2: Uniform mutation joint PDF

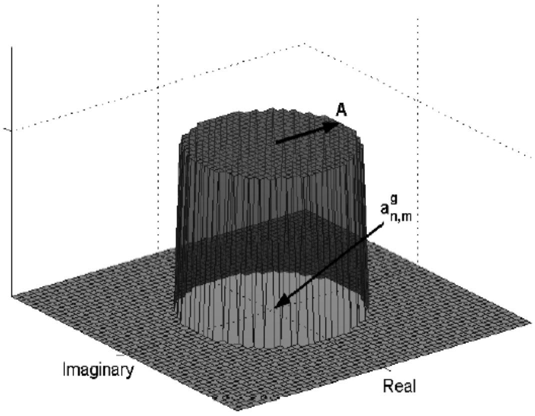


Figure 3: Normal mutation joint PDF

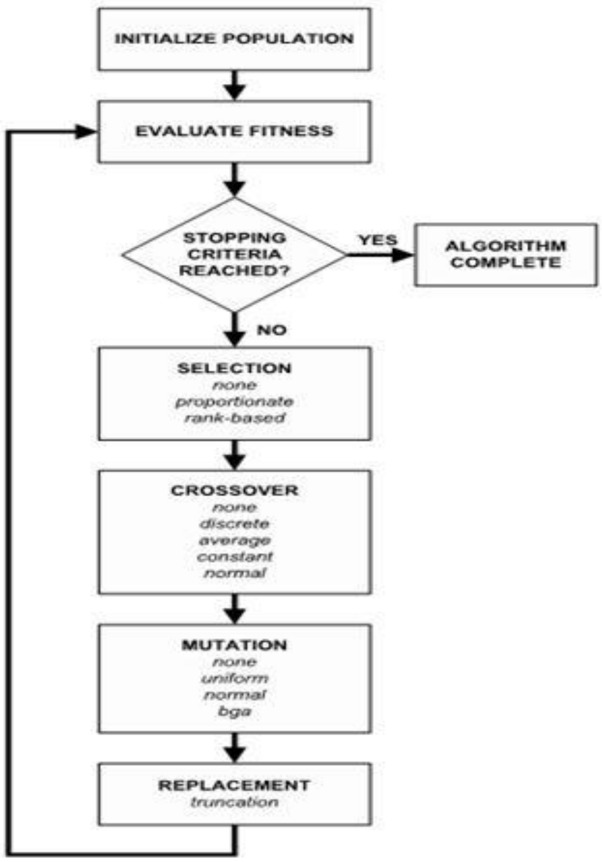


Figure 4: Flow chart of the AGA

AGA steps-

- The initial population is randomly generated by selecting double precision floating-point, complex numbers with uniform probability for all vectors from the chromosome space *X* that encodes the solution space *S*.
- Element fitness $f(x_n)$ is calculated with an objective function that is specified to the given application of the AGA with the requirement that a lower fitness value indicates a better solution.
- The AGA terminates when specified criteria are met. The two utilized in the AGA are called *gen_ max* and *fit_ min*. The algorithm stops when it has executed the number of generation specified by the value of *gen_ max* regardless of the quality of solutions. This is to prevent the AGA from running forever. The algorithm can also stop when an element in the current population has a fitness value equal or less than the value of *fit_min*. This saves time by stopping the AGA when a solution of adequate quality is achieved.
- A subset of elements is selected for recombination during selection. The size of the selection subset is the same as the population. Possible selection strategies are proportionate, rank-based, or none at all. When no selection is utilized, the selection subset is identical to the population set. Truncation selection is not considered here.
- Elements are chosen from the selection subset with probability p_c to become crossover parents. Possible crossover strategies are none, discrete, average, constant or normal. When no crossover is utilized, parents selected for crossover become the offspring. When average crossover is utilized, the number of generated offspring is one-half the number of parents. All other techniques produce the same number of offspring as parents.
- Vector values of the offspring elements are perturbed with probability p_m. Possible mutation methods are uniform, normal, breeder genetic algorithm (BGA) or none at all. Offspring elements from crossover are the only elements that can undergo mutation. The next generation $P(g+1)$ is created by adding the offspring elements into the current population of elements. This is done with a truncation technique that removes the worst elements from $P(g)$ based on fitness to make room for the offspring. The size of $P(g+1)$ remains the same as $P(g)$.

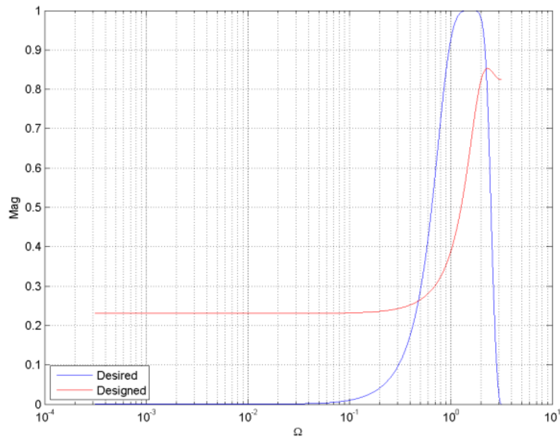
Fitness calculation-

Thus, the fitness of the system would be related on both the magnitude and phase responses of the filter undergoing evaluation and the desired magnitude and phase responses. A frequency weighted squared error technique is designed for this. The fitness is measured by first mapping the vectors of the pole and zero pairs of H_n(z). Next process, the magnitude response of the filter $H_n(e^{j\omega})$ of H_n(z) with a default gain of k=1 is evaluated for all frequency bins of the system. The simulated magnitude response of $H_d(e^{j\omega})$ must be identified at these same frequency bins of the system. To compensate for the use of unity gain in H_n(z), $H_n(e^{j\omega})$ is scaled b k_ny, where k_n is chosen to minimize the error between $k_n H_n(e^{j\omega})$ and $H_d(e^{j\omega})$. This is achieved by forcing the average magnitude value of $k_n H_n(e^{j\omega})$ to equal the average magnitude value of $H_d(e^{j\omega})$. The equation for calculating k_n is,

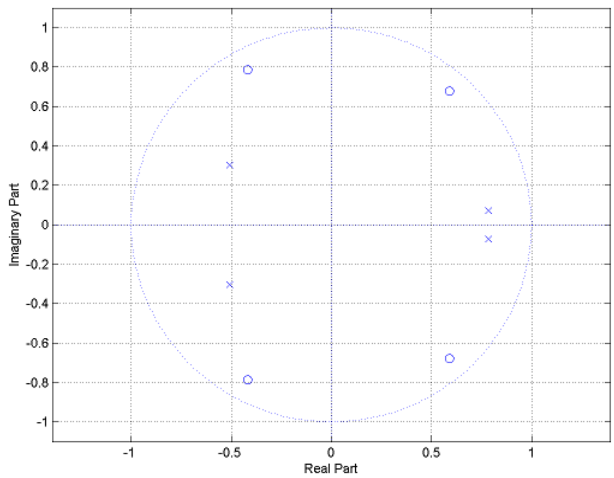
$$k_n = \frac{\sum_{y=1}^{Y} |H_d(e^{j\Omega y})|}{\sum_{y=1}^{Y} |H_n(e^{j\Omega y})|} \qquad (3)$$

Next, the squared error is determined by squaring the difference between $K_n H_n(e^{j\omega})$ and $H_d(e^{j\omega})$ for all. The squared error values are then weighted by multiplying them with a weighting vector Q that assigns a weighting factor to each frequency bin. This enables certain frequency bins of the magnitude response to contribute more or less to the overall fitness of x_n. Finally, the weighted squared error values are summed and scaled to produce the fitness value of x_n. If $k_n H_n(e^{j\omega})$ is identical to $H_d(e^{j\omega})$, then the fitness value will be zero. The complete fitness function utilized by the FDA is,

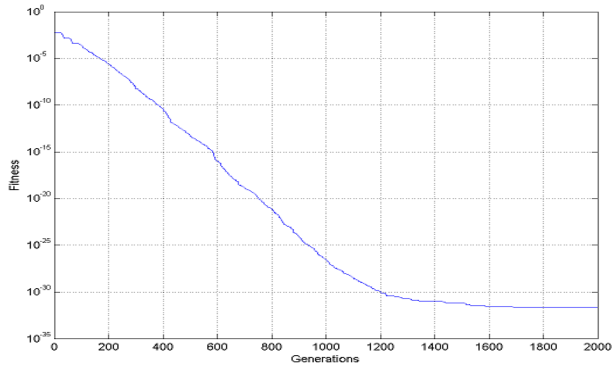
$$f(x_n) = \frac{1}{Y}\sum_{y=1}^{Y}\left[K_n \left|H_n(e^{j\Omega y})\right| - \left|H_d(e^{j\Omega y})\right|\right]^2 Q_y \qquad (4)$$



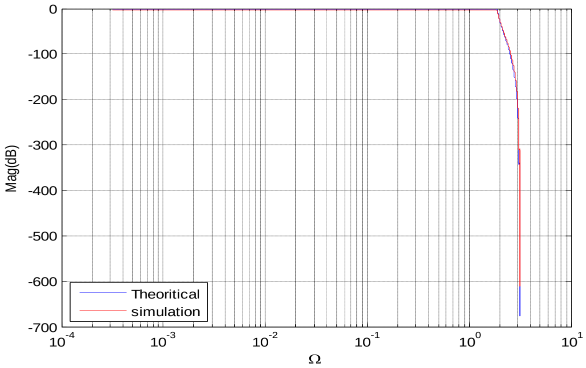
Magnitude Response



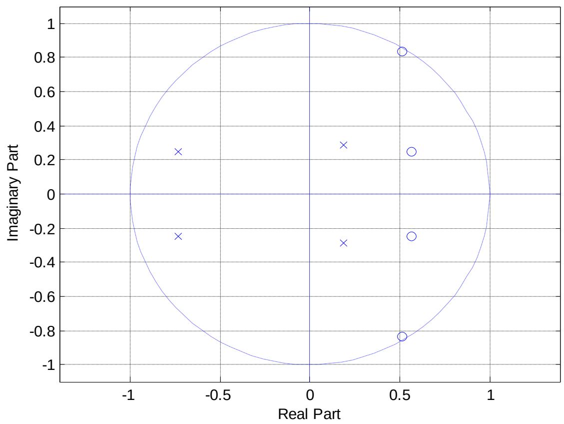
Pole-Zero Diagram



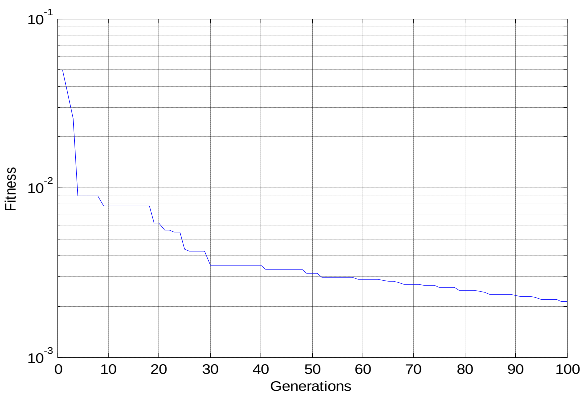
Fitness Curve
4th order Butterworth filter



Magnitude Response



Pole-Zero Diagram



Fitness Curve
4th order chebyshev filter

V. RESULT

The result between the 4th order Butterworth filter and chebyshev filter design are shown in figure. An algorithm result is arbitrary magnitude response, pole-zero diagram and fitness curve are presented. In this simulations, the population size $N = 50$, element size $M = \alpha = 4$ and probability of crossover $P_c = 0.7$. The exit criteria are set to $gen\_max = 1000$ and $fit\_min = 0$, so that the FDA will search for the optimal solution for 1,000 generations. In this algorithm when we increase the order of the filter the complexity is also increases. But when we select the less order than the result is better than the higher order filter.

VI. CONCLUSION

The design of a digital Butterworth IIR filter and chebyshev filter with an arbitrary magnitude response, pole-zero diagram and fitness curve is presented. An algorithm is utilized for continuous search parameter rather than discrete search parameter. This is due to the continuous valued coefficients or roots of a filter transfer function. The algorithm result shows that the best Adaptive filter is 4th order. These results analysed to select the best AGA configuration for complex, continuous parameter optimization of the digital IIR Butterworth and Chebyshev filter. A proposed algorithm provides the excellent stability, robustness, minimum error value, min fitness and arbitrary magnitude response of the filter as compared to different existing optimization algorithm.

REFERENCES

[1] S. Mitra, Digital Signal Processing: A Computer-Based Approach, 2nd ed. Boston: McGraw-Hill Irwin, 2001.
[2] R. Mersereau & M. Smith, Digital Filtering: A Computer Laboratory Textbook. John Wiley & Sons, Inc, 1994.
[3] D. Goldberg, Genetic Algorithm in Search, Optimization, and Machine Learning. Reading, MA: Addison Wesley Pub. Co., 1989.
[4] W. Edmonson et al., “A global least mean square algorithm for adaptive IIR filtering," IEEE Trans. on Circuits and Systems, vol. 45, no. 3, pp. 379-384, Mar 1998.
[5] D. Talla, S. Rao & L. John, “An evolutionary computation embedded IIR LMS algorithm," in Proceedings of International Conference on Signal Processing Applications and Technology, Orlando, FL, Nov 1-4, 1999.
[6] L. Wang, W. Li, & D. Zheng, “A class of hybrid strategy for adaptive IIR filters designs." Shanghai, China: 8th International Conference on Neural Information Processing, Nov 14-18, 2001.
[7] D. Dumitrescu et al., Evolutionary Computation. Boca Raton, FL: CRC Press, 2000.
[8] A. Kosir & J. Tasiċ, “Genetic algorithm and filtering." Sheffield, UK: First International Conference on Genetic Algorithms in Engineering Systems: Innovations and applications, Sep 14-18, 1995.

[9] N. Karaboga and B. Çetinkaya, "Performance Comparison of Genetic Algorithm based Design Methods of Digital Filters with Optimal Magnitude Response and Minimum Phase," The 46th IEEE Midwest Symposium on Circuits and Systems,Egypt.2003

[10] Sangeeta Mandal and S.P. Ghshal, " Swarm intelligence based optimal linear FIR High Pass Filter using Particle Swarm optimization with construction factor and Inertia Weight Approach.

[11] Merkle, D., Middendorf, M.: Swarm intelligence and signal processing. IEEE Signal Process. Mag. 25(6), 152–158 (2008).

[12] Su, K., Cai, H.P.: A modified SQP-filter method for nonlinear complementarily problem. Appl. Math. Model. 33(6), 2890–2896 (2009).

[13] Lin, J., Chen, C.: Parameter estimation of chaotic systems by an oppositional seeker optimization algorithm. Nonlinear Dyn. 76(1), 509–517 (2014).

[14] Soltanpour, M.R., Khooban, M.H.: A particle swarm optimization approach for fuzzy sliding mode control for tracking the robot manipulator. Nonlinear Dyn. 74(1–2), 467–478 (2013).

[15] Zhang, R.D., Lu, R.Q., Xue, A.K., Gao, F.R.: Predictive functional control for linear systems under partial actuator faults and application on an injection molding batch process. Ind. Eng. Chem. Res. 53(2), 723–731 (2014).

[16] Shynk, J.J.: Adaptive IIR filtering. IEEE ASSP Mag. 6(2), 4–21 (1989).

[17] Astrom, K.J., Wittenmark, B.: Adaptive Control. Addison-Wesley, Reading (1995).

[18] Hu, H., Ding, R.: Least squares based iterative identification algorithms for input nonlinear controlled autoregressive systems based on the auxiliary model. Nonlinear Dyn. 76(1), 777–784 (2014).

[19] Regalia, P.A.: Stable and efficient lattice algorithms for adaptive IIR filtering. IEEE Trans. Signal Process. 40(2), 375–388 (1992).

[20] Eberhart, R.C., Shi, Y.: Comparison Between Genetic Algorithms and Particle Swarm Optimization. Evolutionary Programming VII. Springer, Berlin (1998).

[21] Chen, S., Luk, B.L.: Digital IIR filter design using particle swarm optimization. Int. J. Model. Identif. Control 9(4), 327–335 (2010).

