International Journal of Engineering and Technology, 1 (4) (2012) 290-297

www.sciencepubco.com/index.php/IJET

Digital Design and Optimization of Higher Order Adaptive System Using Continuous Genetic Algorithm

Vijay Yadav, Dr.S.S. Bhadauria

Department of Electronics ,MITS, Gwalior
E-mail: vijay_yadavmpct@yahoo.com, saritamits61@yahoo.co.in

Abstract

An algorithm for design and optimization of higher order adaptive system are presented in this paper. In this work, the algorithm applied on the continuous search space parameter rather than discrete search space parameter. A new continuous genetic operator such that Rank based selection, Normal crossover and Mutations are used to improve the rate of convergence and solution quality has been proposed.

Keywords: *Adaptive system, Butterworth bandpass filter, Continuous genetic algorithm (CGA), Filter design algorithm (FDA), IIR filters, Optimization.*

1 Introduction

In digital filter design instability problem, and magnitude and phase response for IIR filters occurring during the design process. In this work, proposed algorithm has been used to design of higher order digital Butterworth bandpass filters which are able to provide the desired characteristics and response of the filter. The Genetic Algorithm (GA) however, is a parallel optimization technique that relies on the basics of evolution for optimizing a group of solutions [3]. The concept of evolution was popularized in the early 1900s [5]. The GA is merely a framework underneath the actual algorithm required for optimization and it must be customized for a given application [4]. The final algorithm presented for optimizing continuous problems is called the continuous genetic algorithm (CGA).

2 Previous Methodology

In this paper digital IIR filter design has been proposed and implemented. Due to a multi-modal mean square error function, IIR design is more preferable than finite impulse response (FIR) design. To solve this, a global least mean square algorithm has been proposed to allow a global minimum search. This is similar to simulated annealing (SA) and is called stochastic approximation with convolution smoothing (SAS) and is realized by convolving the objective function with a noise probability density function. When 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 used 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]. Simulated results show improvements in global optimization compared to the LMS algorithm, pure SA, and pure GA, but the proposed technique are used to 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 [9]. The coefficient symmetry gives us a linear phase even in the case of an IIR filter [10].

3 Proposed Algorithm

A new algorithm called CGA is proposed for the global optimization of multi-minima functions. A new CGA selection, crossover and operator will be proposed and the performance abilities of the CGA will be characterized by applying it to the optimization of randomly generated multimodal objective functions. This will be done for several configurations of the CGA to determine the effect and interaction of the different CGA operators and parameters, such as selection, crossover and mutation strategies. These include strategies for mapping a filter transfer function into the CGA population set, evaluating the quality of filters in the population compared to the desired magnitude response, and preserving both required and desired digital IIR filter properties.

3.1 Encoding function

An encoding function C capable of representing a continuous search space S in the element space X is required by the CGA such that,

$$C: \mathcal{R} \rightarrow X \tag{1}$$

It is impossible to exactly encode R in a binary format to allow processing in a digital environment.

3.2 CGA Operator

Reproduction

1. None
2. Proportionate
3. Rank-based

Crossover

1. Discrete
2. Intermediate
3. Normal (Hybrid)

Mutation

1. Uniform
2. Normal
3. BGA

3.3 Digital IIR filter properties

Let us begin by defining the transfer function *H* (*z*) for a digital IIR filter as [1]

$$H(z) = \frac{N(z)}{D(z)} = \frac{\sum_{i=0}^{\alpha} c_i z^{-i}}{1+\sum_{i=1}^{\alpha} b_i z^{-i}} = k \, \frac{\prod_{i=1}^{\alpha}(z-z_{i})}{\prod_{i=1}^{\alpha}(z-p_{i})} \quad (2)$$

The coefficients of the polynomial form are b_i and c_i. The zeros and poles of the factored form are z_i and p_i respectively. The gain factor *K* is necessary for equivalence between the polynomial and factored forms. The order of *H* (*z*) is determined by α.

Two properties of *H* (*z*)

- A causal, linear, time invariant (LTI) system with system function *H* (*z*) is bounded input bounded output (BIBO) stable if and only if all the poles of *H* (*z*) lie inside the unit circle. ($|p_i|<1$)
- A causal, stable, LTI system with system function *H* (*z*) is real if and only if all complex poles and zeros of *H* (*z*) have complex conjugate pairs or exist singularly on the real axis.

3.4 Design algorithm steps

The complete flow chart of the CGA is shown in Fig.1. It is quite similar to the traditional DGA block diagram except for the use of continuous rather than discrete operations [11]. For sake of completeness, the details of each stage of the algorithm are provided. The new hybrid operator, normal crossover, is classified as a crossover method that can be combined with an additional mutation method.

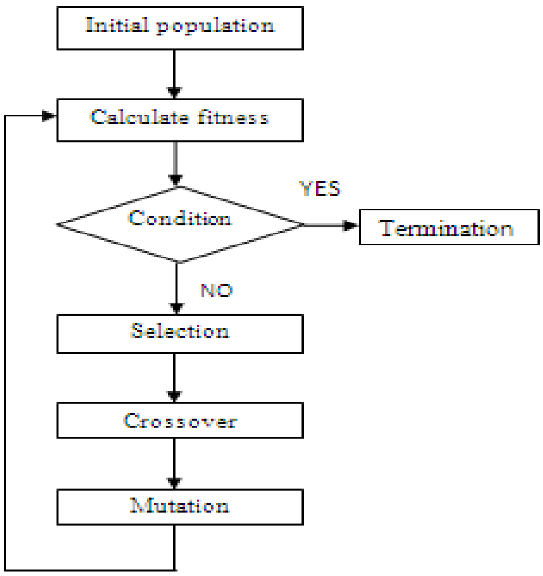


Fig.1: Flow chart of CGA

- The initial population *P* (0) is randomly generated by selecting double precision floating-point, complex numbers with uniform probability for all vectors $a^0_{n,\,m}$ 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 CGA with the requirement that a lower fitness value indicates a better solution.
- The CGA terminates when specified criteria are met. The two used in the CGA 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 CGA 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 CGA 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 used, 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 used, parents selected for crossover become the offspring. When average crossover is used, 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).

3.5 Filter mapping to CGA

For the CGA to evolve and optimize digital IIR filters, a method for mapping a filter transfer function H_n(z) to an element x_n is needed. Two straightforward methods for doing this include mapping either the coefficients of the polynomial form of H_n(z) or the roots and gain of the factored form of H_n(z) to the vectors of x_n. While both of these options are mathematically equivalent, polynomial coefficients b_i and c_i can have several orders of magnitude of dynamic range necessitating a very large search space S. Filter stability requires that all poles p_i of $H_n(z)$ are inside the unit circle, thus, limiting the search space S for p_i. Minimum-phase is accomplished by having all poles and zeros of $H_n(z)$ inside the unit circle. This relationship between vectors changes the way crossover and mutation can operate. For instance, if crossover generates an offspring with a complex vector $a_{n,m}$ the crossover operator must ensure that a complex vector $a_{n,k}$ that satisfies is also generated. Furthermore, if mutation modifies $a_{n,m}$ by then mutation must also modify $a_{n,k}$ *by* λ*. Removing these vector dependencies allows previously discussed crossover and mutation strategies for the CGA to be used. This is done by removing the vector $a_{n,k}$ from x_n and interpreting every complex vector $a_{n,m}$ as two vectors $a_{n,m}$ and $a^*_{n,m}$. This simplification act does not come without consequence. It requires the number of poles and the number of zeros of $H_n(z)$ to be even, and poles and zeros are no longer able to exist singularly on the real axis. However, the reduction in the size of M and the ability to use regular crossover and mutation operators serves as sufficient justification.

3.6 Finding the fitness function of the filter

Thus, the fitness function of the CGA should be based on both the magnitude responses of the filter undergoing evaluation and the desired magnitude response. A frequency weighted squared error technique is proposed for this. The fitness of x_n is calculated by first mapping the vectors of x_n to the pole and zero pairs of H_n(z). Next, the magnitude response $H_n(e^{j\omega})$ of H_n(z) with a default gain of k=1 is evaluated for all frequency bins. The desired magnitude response of $H_d(e^{j\omega})$ must also be identified at these same frequency bins. To compensate for the use of unity gain in H_n(z),H_n(e$^{j\omega}$) is scaled byk_n, 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,

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

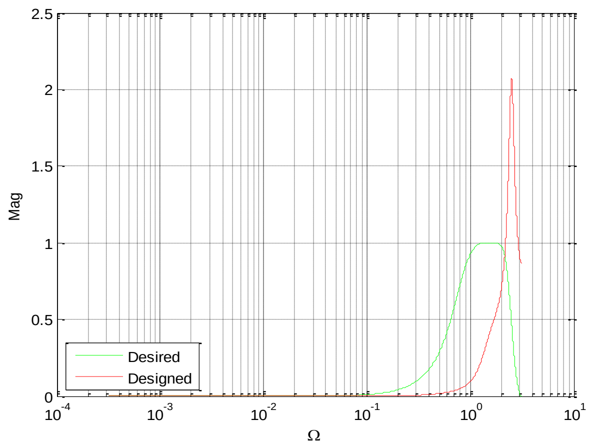
Next, the squared error is calculated by squaring the difference between $K_n H_n$(e$^{j\omega}$) and H_d(e$^{j\omega}$) for all. The squared error values are then weighted by multiplying them with a weighting vector Q that assigns a weighting factor to each frequency bin. This enables certain frequency 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 used by the FDA is,

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

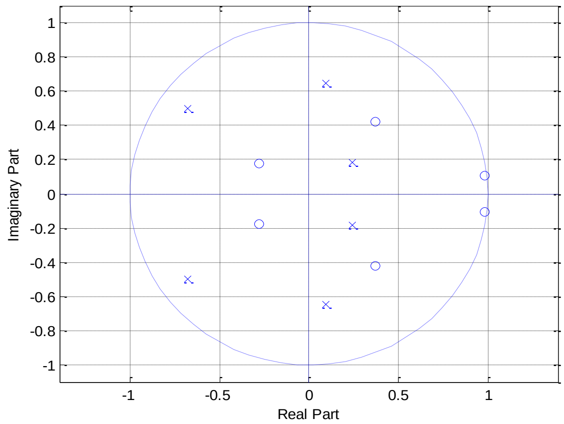
Where Y is the total number of frequency bins, Ω_y is an element of Ω, and Q_y is an element of Q. Since only real filters are considered for H_n(z), Ω is generally only specified in the range of 0 to π. Any distribution of frequency points, such as linear for logarithmic spacing, within Ω can be applied with varying results on the outcome of the algorithm.

4 Results

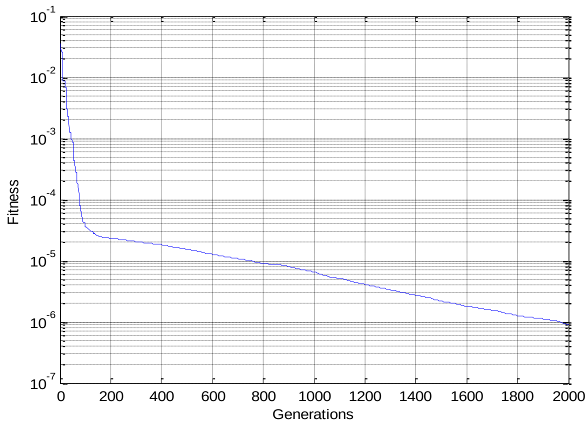
The algorithm was applied to the 6th order Butterworth bandpass filter design shown in Fig.2. In this simulation, the population size N = 100, element size M = α = 6 and probability of crossover P_c= 0.7. The exit criteria are set to *gen_ max* = 2000 and *fit_ min* = 0, so that the FDA will search for the optimal solution for 2,000 generations.



(a) Magnitude response plot



(b) Pole-zero diagram



(c) Fitness curve

Fig.2: 6th order Butterworth filter

5 Conclusion

In this paper, the design of a digital Butterworth IIR filter with an arbitrary magnitude response, pole-zero diagram and fitness curve is presented. An algorithm is used for continuous search parameter rather than discrete search parameter. Due to the continuous valued coefficients or roots of a filter transfer function. These results analyzed to select the best CGA configuration for complex, continuous parameter optimization of the digital IIR filter. A proposed algorithm provides the better stability, robustness, minimum error and arbitrary magnitude response of the filter as compared to different existing 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] Holland, Adaptation in Natural and Artificial Systems. Ann Arbor, MI: The University of Michigan Press, 1975.

[5] C. Darwin, the Origin of Species, ser. The Harvard Classics. New York: P F Collier & Son, 1909, vol. 11.

[6] 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.

[7] 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.

[8] 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.

[9] D. Dumitrescu et al., Evolutionary Computation. Boca Raton, FL: CRC Press, 2000.

[10] 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.

[11] 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