© July 2015 | IJIRT | Volume 2 Issue 2 | ISSN: 2349-6002

# REALIZATION OF BUTERWORTH IIR FILTER USING GENETIC ALGORITHM

Pankaj Mohindru, Tarandeep Kaur, Pooja Mohindru
*Student, Punjabi University, Patiala, India*

***Abstract***— **Genetic algorithms are search methods based on principles of natural selection and genetics. These encode the decision variables of a search problem into finite-length strings. The strings are referred to as chromosomes and the alphabets are referred to as genes. This paper presents the design of IIR filter using GA. To formulate GA capable of designing an IIR filter, various constraints has been developed for the desired fitness function. The proposed algorithm has been tested for Butterworth filter. The magnitude response of the designed Butterworth IIR filter almost matches the desired response with an error 2e-6%. This shows the accuracy of the proposed algorithm. . Comparison of the pole-zero plot shows that there is very small error between the pole-zero placements in the desired and designed filters. Also the fitness function has converged with 1300 generations and achieve minimum value of is approximately 3e-32.**

***Index Terms***—**Infinite Impulse Response Filter, Genetic algorithm, Butterworth Filter.**

## I. INTRODUCTION

The basic idea of Genetic algorithm is that it works in iterations and there is an improvement in every step as a result of benefiting from the inherited treats from the previous step. It starts from the initial solution, in each iteration step, an operation is selected and it is applied to the current solution. If the altered result is acceptable e.g. it is better than the current one, it becomes the current solution for the next iteration, otherwise, it is refused. The iteration process stops when the requested solution or the maximum number of iterations is reached. In general, genetic algorithms are better than gradient search methods if the search space has several local minima or maxima. Since the genetic algorithm traverses the search space using the genotype rather than the phenotype, it is less likely to get stuck in a local high or low [3, 4]. GA's start with a set of initial random solutions called population and these solution are evaluated by test and are sorted according to their fitness, those solution or individuals having higher fitness values are given a chance to reproduce by an operation called cross-over and the less-fit individuals are discarded. After crossover some bits "gene" are flipped either from 1 to 0 or vice versa in an operation that mimics mutation in living organisms. A flowchart of a generic genetic algorithm is shown in figure 1.1.

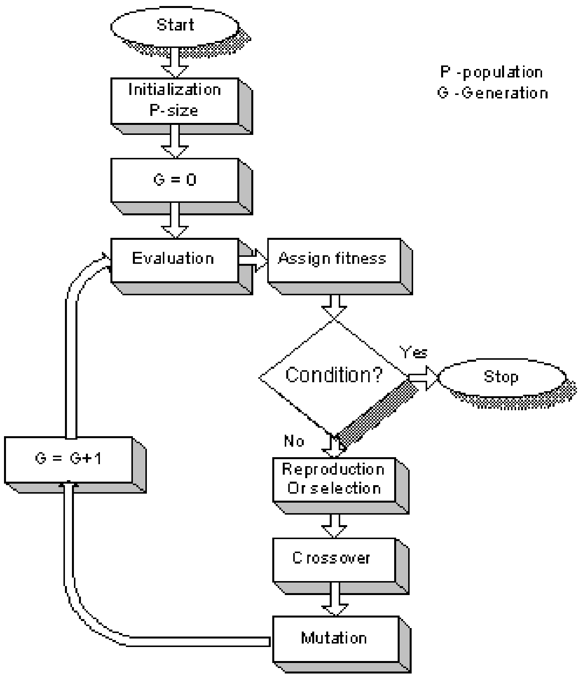


**Fig. 1.1 : Genetic algorithm flow chart**

Although the GA is a general optimization algorithm, but it needs to be modifed to design a digital IIR filter design. These modifications include a method for mapping a filter to an element, evaluation of the fitness function of the IIR filter, creation of an initial population of the IIR filter, and very importantly, the designed filter must be are realizable. But for development of fitness function, transfer function $H(z)$ for a digital IIR filter could be defined as [27]

$$H(z)=\frac{N(z)}{D(z)}=\frac{\sum_{i=0}^{\alpha}c_i z^{-i}}{1+\sum_{i=1}^{\alpha}b_i z^{-i}}=K\times\frac{\prod_{i=1}^{\alpha}(z-z_i)}{\prod_{i=1}^{\alpha}(z-p_i)}. \qquad (1.1)$$

Where bi and ci are the cofficients of the polynomial. zi and pi, represents zeroes and poles respectively. K is the gain factor. A determines the order of the filter. Not that for a filter to be realizable the following two conditions must meet [1]

- A causal, Linear time invariant 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. $\left(|p_i|<1\right)$
- 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.

## II. FILTER DESIGN USING GA

Generally GA must be modified to apply it to the FDA. We have mapped the filter transfer function *Hn*($z$) to an element *xn*. We have mapped r the coefficients of the polynomial form of *Hn*($z$) to the vectors of *xn*. As filter stability requires that all poles *pi* of *Hn*($z$) must be inside the unit circle. Thus we have put the constraint to the value of *pi*. To meet the minimum phase requirements, same constrain has also been applied to the zeros *zi*,. Although this can put a restriction on the required phase.

Ignoring *Kn* and putting $M = 2\alpha$ in equation 1.1, complex vectors will be required to map *Hn*($z$) to *xn*. For this complex vector requirement and Hn($z$) to be real all poles and zeroes must have a complex conjugate pair or they should lie on the real axis. To meet above requirement, we can say, for every complex vector *an,m* in *xn*, there must exist another complex vector *an,k* where

$$a_{n,k}=a_{n,m}^{*} \qquad (1.2)$$

This relationship between vectors changes the way crossover and mutation can perate. 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 satifies (1.2) is also generated. Moreover, if mutation modifies $a_{n,m}$ by , $\lambda$, then mutation must also modify $a_{n,k}$ by $\lambda^{*}$. The gain $K_n$ is not mapped into $x_n$. Thgis is due to the fact that as the pole and zeros locations are restricted inside the unit circle, the gain factor may lie in the range of $0<K<\infty$.

Population management for good performance is a major issue in design problems where GA is used. The random generation of the initial population $P(0)$ of filters and check on $P(g)$ for all $g$ to keep $P$ within the range of $S$ is very important After selecting the value of $\alpha$, zero and pole locations should be randomly selected for each $x_n$ in $P(0)$. For the proposed filter, we have selected a complex vector with the uniform joint PDF as

$$a_{n,m}^{0}=\mathrm{Re}^{j\theta} \qquad (1.3)$$

$$f_{R,\theta}(\mathrm{r},\theta)=\begin{cases}\dfrac{1}{2\pi r^2}, & 0\le r\le 1, 0\le\theta<\pi\\ 0, & otherwise\end{cases} \qquad (1.4)$$

For crossover and mutation process to manipulate vectors without regard to $S$, we have used strategy where any vector representing poles and zeros outside the unit circle is mapped back into the unit circle. Any zero $z_i$ that lies outside the unit circle is mapped to a zero $z_i'$ inside the unit circle with the equation

$$z_i'=\frac{1}{z_i^{*}} \qquad (1.5)$$

As the shape of the filter magnitude response will change when mapping poles from outside to inside the unit circle. Therefore, it is assumed that all poles are located within the unit circle before crossover and mutation are applied. The pole mapping strategy needed to maintain stability can be seen in (1.5). The reciprocal nature of the equation maps poles close to the unit circle to another location close to the unit circle and poles far from the unit circle to a location closer to the origin. equation 1.5 is equally valid for poles with $z_i$ replaced by $p_i$.

As our main aim is to design and optimize a IIR filter with an arbitrary magnitude response. Hence, the fitness function should include both the magnitude responses of the filter undergoing evaluation and the desired magnitude response. The fitness function is evaluated as under:

1. The fitness of $n$ is calculated by first mapping the vectors of $x_n$ to the pole and zero pairs of $H_n(z)$.
2. The magnitude response $\left|H_n\left(e^{j\Omega}\right)\right|$ of $H_n\left(z\right)$ with a default gain of $K$ = 1 is evaluated for all frequency bins Ω.
3. The desired magnitude response $\left|H_d\left(e^{j\Omega}\right)\right|$ is also identified at these same frequency bins.
4. To compensate for $H_n\left(z\right)$, $\left|H_n\left(e^{j\Omega}\right)\right|$ is scaled by $K_n$ , where $K_n$ I chosen to minimize the error between $K_n\left|H_n\left(e^{j\Omega}\right)\right|$ and $\left|H_d\left(e^{j\Omega}\right)\right|$. This is achieved by forcing the average magnitude value of $K_n\left|H_n\left(e^{j\Omega}\right)\right|$ to equal the average magnitude value of $\left|H_d\left(e^{j\Omega}\right)\right|$ . The equation for calculating $K_n$ is

$$K_n = \frac{\sum_{y=1}^{Y}\left|H_d\left(e^{j\Omega_y}\right)\right|}{\sum_{y=1}^{Y}\left|H_n\left(e^{j\Omega_y}\right)\right|} \qquad (1.6)$$

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

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

where $Y$ is the total number of frequency bins, $\Omega_y$ is an element of Ω, and $Q_y$ is an element of Q .

## III. DESIGN OF BUTTERWORTH FILTER

To design, Butterworth filter following parameter selected as follows:

- The desired magnitude response $\left|H_d(e^{j\Omega})\right|$ a fourth-order Butterworth bandpass filter with lower and upper 3-dB cutoff points of $\Omega_l = \frac{\pi}{4}$ and $\Omega_u = \frac{3\pi}{4}$ , and unity passband gain.
- The frequency vector Ω for specifying $\left|H_d(e^{j\Omega})\right|$ and evaluating $K_n\left|H_n(e^{j\Omega})\right|$ consists of 10,000 frequency bins equally spaced between 0 and $\pi$ .
- The weighting vector Q equals 1 for all 10,000 points. This force all frequency bins of the magnitude response to be equally important for optimization.
- To accommodate the exact Butterworth transfer function output α is selected as 4.Population size N = 200, element size M = α = 4, and probability of crossover pc = 0.7. The exit criteria are set to gen_max = 2,000 and fit_min = 0.

The magnitude response of the designed IIR filter is shown in Figure 1.2. Its pole-zero plot is shown in Figure 1.2. For comparison, the pole-zero plot for the theoratical fourth order Butterworth bandpass filter is shown in Figure 4.3. The errors between the pole-zero placements in the desired and designed filters are listed in Table 4.1. The fitness function convergence is shown in Figure 4.4. From Figure 4.4, it has been concluded that the ending fitness level of the designed filter is approximately 3e-32 and that major fitness improvements ceased after approximately 1,100 generations.

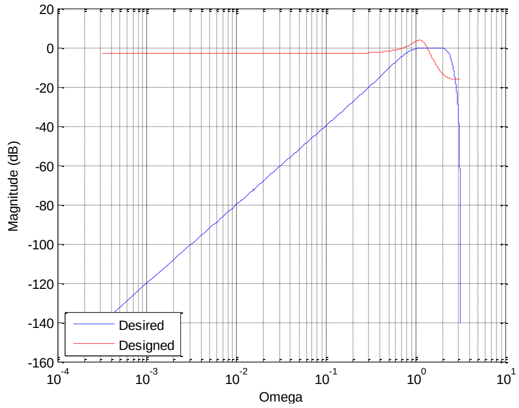


**Figure 1.2. Magnitude response of the designed Butterworth filter .**

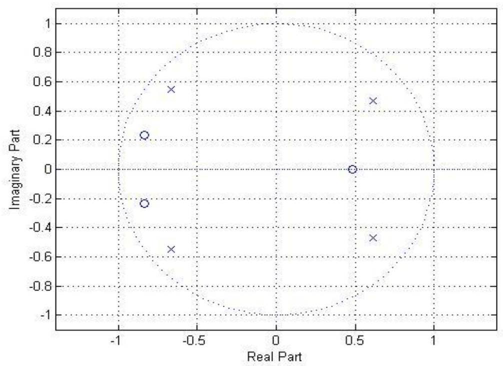


**Figure 1.3. Pole-zero plot for the designed butterworth filter.**

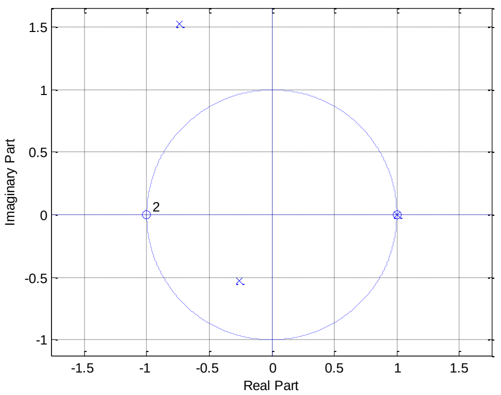


**Figure 1.4. Pole-zero plot for the desired Butterworth filter**

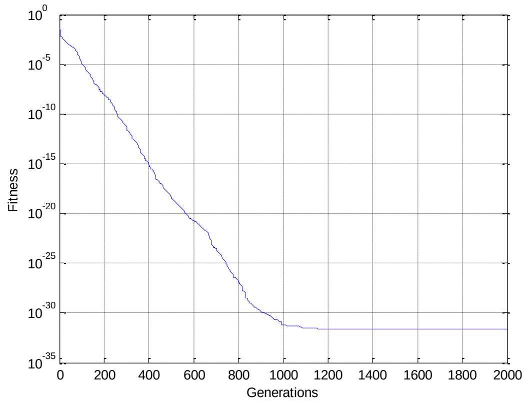


**Figure 1.5. Fitness curve of the Butterworth Filter**

## IV. CONCLUSION

This paper prposes the design of IIR filter using GA. Various constraints has been developed for the desired fitness function for designing an IIR filter. The proposed algorithm has been tested for Butterworth filter. The magnitude response of the designed Butterworth IIR filter almost matches the desired response with an error 2e-6%. The proposed work shows the accuracy of GA algorithm. . Comparison of the pole-zero plot shows that there is very small error between the pole-zero placements in the desired and designed filters. Also the fitness function has converged with 1300 generations and achieve minimum value of is approximately 3e-32.

## REFRENCES

[1] S. Mitra, "*Digital Signal Processing: A Computer-Based Approach*", 2nd ed. Boston: McGraw-Hill Irwin, 2001.

[2] Wu-Sheng Lu, *Fellow, IEEE,* and Takao Hinamoto, *Fellow, IEEE,"* Optimal Design of IIR Digital Filters With Robust Stability Using Conic-Quadratic-Programming Updates", IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 51, NO. 6, JUNE 2003

[3] Mathias C. Lang," Least-Squares Design of IIR Filters with Prescribed Magnitude and Phase Responses and a Pole Radius Constraint", IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 48, NO. 11, NOVEMBER 2000

[4] Wu-Sheng Lu, Dept. of Electrical and Computer Engineering University of Victoria

Victoria, BC, Canada V8W 3P6," An Argument-Principle Based Stability Criterion and Application to the Design of IR Digital Filters" 0-7803-9390-2/06/$20.00 ©C2006 IEEE

[5] bSuman Saha, bSonam Yangchen, bDurbadal Mandal, bRajib Kar, aSakti Prasad Ghoshal, bDepartment of Electronics and Communication Engineering a Department of Electrical Engineering National Institute of Technology Durgapur, West Bengal, INDIA," Digital Stable IIR High Pass Filter Optimization using PSO-CFIWA", 2012 IEEE Symposium on Humanities, Science and Engineering Research

[6] Rajeev C. Nongpiur, *Member, IEEE*, Dale J. Shpak, *Senior Member, IEEE*, and Andreas Antoniou, *Life Fellow, IEEE,"* Rajeev C. Nongpiur, *Member, IEEE*, Dale J. Shpak, *Senior Member, IEEE*, and Andreas Antoniou, *Life Fellow, IEEE",* IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 61, NO. 4, FEBRUARY 15, 2013