Exploring the use of Genetic Algorithms Toolbox in Engineering Education: Did it Provide an Interesting Learning Experience for Students?

Muhammad Anwar [1], Hendra Hidayat [1], Elsa Sabrina [1]

[1] *Universitas Negeri Padang, Jalan Prof Hamka Kampus Air Tawar Barat, Padang, Indonesia*

Abstract **– Teaching in engineering education using Genetic Algorithms (GAs) toolbox as a teaching tool in engineering education can be an effective approach, particularly in Signal Processing subjects, as it encourages students to learn how to find optimal values required in designing digital filters. This research investigates the use of GAs for teaching digital filter design, where the evaluation function of the problem is optimized using Gas. The GAs toolbox simulation is designed to yield a stable, lowest-order H[z] that meets the tolerance parameters and satisfies the design criteria. Teaching GAs involves representing the filter design problem in a way that can be accepted by genetic programming.**

Keywords **– Engineering education, genetic algorithm, digital filter design, signal processing.**

1. Introduction

Engineering education in the last few decades has become a serious concern, and teaching which indicates saturation and lack of experience has become a common thought.

DOI: 10.18421/TEM123-54
https://doi.org/10.18421/TEM123-54

Corresponding author: Muhammad Anwar,
Universitas Negeri Padang, Jalan Prof Hamka Kampus Air Tawar Barat, Padang, Indonesia
Email: muh_anwar@ft.unp.ac.id

Received: 01 April 2023.
Revised: 13 July 2023.
Accepted: 01 August 2023.
Published: 28 August 2023.

Digital filter design in signal processing subjects is one of the challenges in engineering pedagogy, so it requires interesting experiences in the teaching and learning process. The design filters work, including digital filters, require high precision and robustness due to their difficulty and complexity. In the increasingly developing signal processing tasks in the digital era, the problem of filter optimization as the main goal of digital signal processing (DSP) has been widely considered [1], [2], [3]. The traditional approach to filter design involves analog-to-digital conversion with certain criteria, usually using a bilinear transformation [4], [5]. Current design functions (Butterworth, Chebyshev type I and II and Elliptical) generally rely on the approximation method to implement frequency-selective analog filters such as Low, High, Band Pass and Band Stop. Until now, a filter design with more optimal performance and fulfilling the criteria, given in studies [1], [2] is needed. In general, there are 3 design criteria, namely: set the lowest filter, the filter must be stable (posts are inside the unit circle); and must meet tolerance requirements.

Mathematical and non-mathematical approaches have been extensively studied to meet all the criteria for optimal filter design [3]. An optimal design method based on the least-squares method was studied for the least-squares error criterion model [4]. Particle swarm optimization (PSO) has been applied to linear-phase FIR low and high digital filters optimization works [5]. The impulse response of each filter in this study gives a comprehensive characterization in accordance with the filter design parameters, and filtering systems like infinite impulse response (IIR) filters are also described by linear differential equations or linear transformations [6]. Another study proposed a new development design method structurally diverse digital filters [7].

Researchers currently focus on the application of evolutionary computing concepts and techniques as the greatest opportunity to overcome the weaknesses of traditional design techniques [8].

The use of GAs to solve problems, especially optimization techniques, continues to penetrate many disciplines, while GA theory itself is evolving [9], [10]. To create orthogonal filter banks for voice compression, a multi-objective genetic algorithm (GA) optimization model was put out [11]. A new FIR filter design technique using a genetic algorithm was also discussed. The filter coefficients are searched in a discrete space, so the architecture comprised of transitions and two adders [12]. For the construction of broadband infinite impulse response (IIR) digital differentiators (DDs), a method based on a weighted L1 norm optimization criterion was suggested and applied with the Salp-Swarming Algorithm (SSA) [13]. The accuracy, robustness, consistency, and efficiency of two popular methods—Particle Swarm Optimization (PSO) and Real Coded Genetic Algorithm (RCGA)—are compared in this paper. To solve the optimization problem, researchers used evolutionary algorithms to design the IIR filter [3]. Thus, it can be explained that the use of evolutionary algorithms is not a new thing in filter design works.

Genetic algorithm is a form of random search (stochastic), which imitates the principles of natural biological evolutionary processes to find an optimal solution of a complex problem [14], [15], [16]. A genetic algorithm is an event that has the inherent freedom and flexibility to select the desired solution according to design specifications. The obtained optimal value is the final product of the evolution of generations with the best individuals of the given population [17]. Thus, it is expected that learning how to design digital filters using GA will facilitate students' work on signal processing techniques and ensure optimal design results. In addition, the student will also be able to manage the use of GA as part of learning AI applications for wider use. Therefore, this paper discusses the teaching of optimal digital filter design using GAs and genetic programming in Signal Processing subject.

2. Methods

The scope of teaching begins with the study of input specifications and genetic algorithm processes as a tool to find optimal solutions in digital filter design. Filter design is seen as an objective function that represents an optimization problem. The objective function is then set as an evaluation function which will be processed by genetic programming. Therefore, the learning steps are stated as follows in Figure 1.

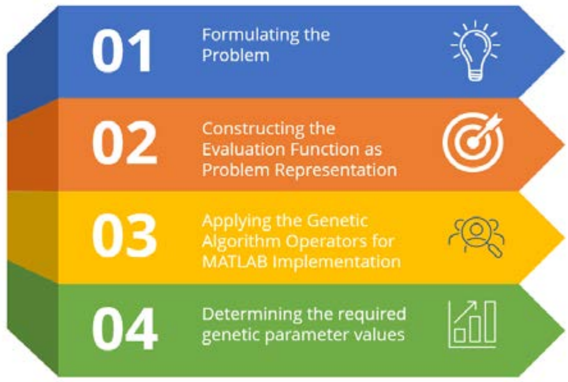

Figure 1. Four step flow scope GAs to find the solution

The evaluation function is an expression of the subject matter to be optimized. Therefore, it is necessary to first formulate the problem with the following steps:

1) Define the basic filter structure to be used as a transfer function for each filter type. For example, if we take the third-order cascade form for a lowpass filter, we get the transfer function as follows:

$$\mathrm{H}(z) = K\frac{(z+b_1)}{(z+a_1)}x\frac{(z^2+b_{11}z+b_{12})}{(z^2+a_{11}z+a_{12})} \quad (1)$$

where the value of K and each coefficient ($a_1, a_{11}, a_{12}, b_1, b_{11}, b_{12}$) is within the constraints that meet the requirements for the value of each pole, namely $-1 < z < 1$. This means that K and all coefficients must have values within the constraints such that an optimal solution is obtained.

2) Since the optimal condition of the filter is indicated by its magnitude response, the transfer function in the z domain must be represented in the frequency domain ω, with $z = e^{j\omega}$.

3) The evaluation function is obtained, namely:

$$f(z)_{evai} = f(e^{j\omega}, solusi) \quad (2)$$

$$\mathrm{H}\,(e^{j\omega}) = \mathrm{sol}\,(7)\,\frac{(e^{j\omega}+sol\,(1))((e^{j\omega})^2+sol(2)e^{j\omega}+sol(3))}{(e^{j\omega}+sol\,(3))((e^{j\omega})^2+sol(5)e^{j\omega}+sol(6))} \quad (3)$$

4) Where value ω about $0 \le \omega \le \pi$. Each solution (sol) must be meet the constraints pole, as follows:
   - $1 \le sol(1),(2),(3),(4),(5),(7) \le 1$
   - $-2 \le sol(5) \le 2$

The ideal magnitude response value in practical specifications for a lowpass filter in the range ω is:

$$|H(e^{j\omega})| = \begin{cases} (1-\delta_1) \le |H(e^{j\omega})| \le (1+\delta_1), & \text{for } 0 \le \omega \le \omega_p \\ 0 \le |H(e^{j\omega})| \le \delta_2, & \text{for } \omega_s \le \omega \le \pi \end{cases} \quad (4)$$

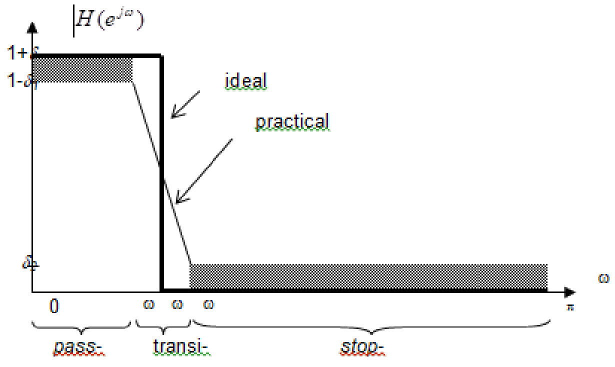

Figure 2. Ideal and practical magnitude response

To apply genetic algorithm operators to genetic programming, the structure must be represented in a form that is understood by the genetic programming language. The genetic programming in this research experiment was implemented using the MATLAB toolbox. MATLAB is a technical computing area for visualization and high-performance numerical computation [18]. MATLAB combines numerical analysis, matrix computing, signal processing, and graphics in an easy-to-use setting.

Every problem and solution are simply expressed in ordinary mathematical formulas, unlike traditional programming. As in genetic programming, all genetic parameters including the number of generations, population size, probability of selection, crossover, and mutation rate must be given in a form that is suitable for the genetic programming language. The value of the given parameter will greatly verify the results obtained. Therefore, optimal results can be obtained in several trials. However, GA still guarantees that the parameter obtained is the optimal value of the transfer function being tested.

3. Methods

The expected result of this experiment is that the method of teaching DSP with genetic algorithms is obtained. The learning begins with the discovery of transfer functions for each type of digital filter (LP, HP, BP, and BS) through the integrated execution of genetic programming. The evaluation function with solution constraints for each type of filter is modeled and presented in following table.

Table 1. Filter type evaluation function

Type of filter	Evaluation Function	Criteria
Lowpass (LP)	$H(e^{j\omega}) = sol(7)\left(\frac{(e^{j\omega}+sol(1))((e^{j\omega})^2+sol(2)e^{j\omega}+sol(3))}{(e^{j\omega}+sol(3))((e^{j\omega})^2+sol(5)e^{j\omega}+sol(6))}\right)$ where: $z = e^{j\omega}$, $0 \le \omega \le \pi$ solution constraint (1)=(2)=(3)=(4)=(6)=(7)= [-1 1] solution constraint (5) = [-2 2]	Order = 3 Tolerance setting: *pass*=[1 0.95] *stop*=[0.1 0] $\omega_p = 0.2\pi$ $\omega_s = 0.3\pi$
Highpass (HP)	$H(e^{j\omega}) = sol(7)\left(\frac{(e^{j\omega}+sol(1))((e^{j\omega})^2+sol(2)e^{j\omega}+sol(3))}{(e^{j\omega}+sol(3))((e^{j\omega})^2+sol(5)e^{j\omega}+sol(6))}\right)$ where: $z = e^{j\omega}$, $0 \le \omega \le \pi$ solution constraint (1)=(2)=(3)=(4)=(6)=(7)= [-1 1] solution constraint (5) = [-2 2]	Order = 3 Tolerance setting: *pass*=[1 0.95] *stop*=[0.1 0] $\omega_p = 0.8\pi$ $\omega_s = 0.7\pi$
Bandpass (BP)	$\lvert H(e^{j\omega})\rvert = sol(13)\frac{(z+sol(1))(z+sol(2))}{(z+sol(7))(z+sol(8))}$ $x\ \frac{(z^2+sol(5)z+sol(6))}{(z^2+sol(11)z+sol(12))}$ where: $z = e^{j\omega}$, $0 \le \omega \le \pi$ solution constraint (1)=(2)=(3)=(4)=(5)=(6)=(7)=(8)= (10)=(12)=(13) = [-1 1] solution constraint (9) = (11) = [-2 2]	Order = 6 Tolerance setting: *pass*=[1 0.95] *stop*=[0.1 0] $\omega_{s1} = 0.24\pi$ $\omega_{p1} = 0.4\pi$ $\omega_{p2} = 0.6\pi$ $\omega_{s2} = 0.76\pi$
Bandstop (BS)	$\lvert H(e^{j\omega})\rvert = sol(9)\frac{(z+sol(1))(z+sol(2))}{(z+sol(5))(z+sol(6))}$ $x\ \frac{(z^2+sol(2)z+sol(2))}{(z^2+sol(3)z+sol(8))}$	Order = 4 Tolerance setting: *pass*=[1 0.95] *stop*=[0.1 0] $\omega_{p1} = 0.24\pi$ $\omega_{s1} = 0.4\pi$ $\omega_{s2} = 0.6\pi$ $\omega_{p2} = 0.76\pi$

Table 1 displays the results of modeling the evaluation function for each type of filter using the cascade form filter structure, while also adhering to solution constraints. The filter order is set at the lowest value, without neglecting other design criteria, which is to remain stable and meet the tolerance limit.

In teaching genetic algorithms, following the evaluation function formulation, input parameters of genetic operators are also given to run the program in an integrated, as shown in the table below.

Table 2. Genetic operator input parameters

Parameter of Generic Operator	LP filter	HP filter	BP filter	BS filter
Population	121	101	112	99
Generation	3226	2790	10055	8242
Roulette selection coefficients	0.04	0.04	0.04	0.04
Crossover operators: Arithmetic Heuristic Simple	 [2 0] [2 3] [2 0]	 [2 0] [2 3] [2 0]	 [2 0] [2 3] [2 0]	 [2 0] [2 3] [2 0]
Mutation operators: Boundary Uniform	 [2 0 0] [2 0 0]	 [2 0 0] [2 0 0]	 [2 0 0] [2 0 0]	 [2 0 0] [2 0 0]

Following program execution, the initialization function will generate the starting population at random. The number of individuals created is represented by a matrix that includes the population's dimensions and a solution variable. For example, for an LP filter with a population of 121 and the solution variable being sought is 7, the number of individuals forms a 121x7 dimension matrix. Individuals who produce the greatest fitness value of the evaluation function (as in the 54th population), also have a high fitness value. Furthermore, these individuals (the best population) have the greatest chance of being selected as parents in order to give birth to a new, more resilient population in the next generation.

After the program is executed, the optimal solution is obtained for each type of filter that meets the design criteria as shown in the table above. Furthermore, the evaluation function can be expressed in the z domain in the form of the following transfer function.

- $\mathrm{H}\,(z)_{LP} = \frac{0.1152\,z^3-0.0557\,z^2+0.0661z+0.0381}{z^3-1.9210\,z^2+1.5149z-0.4287}$
- $\mathrm{H}\,(z)_{HP} = \frac{0.0831\,z^3-0.0085\,z^2+0.0016z-0.0761}{z^3+1.9110\,z^2+1.5063z+0.4248}$
- $\mathrm{H}\,(z)_{BP} = \frac{0.1315\,z^6+0.0024\,z^5-0.2069z^4-0.0022z^3+0.1083z^2+0.0005z-0.0189}{z^6-0.0116\,z^5+0.9165z^4-0.0167z^3+0.4083z^2-0.0092z-0.0002}$
- $\mathrm{H}\,(z)_{BS} = \frac{-0.4295z^4-0.0056\,z^3-0.6921z^2-0.0006z-0.3484}{z^4 0.0078z^3+0.2501z^2+0.0001z+0.2826}$

The magnitude response of each filter type defined by the criteria and its evaluation function in Table 1 is in Figure 3 below.

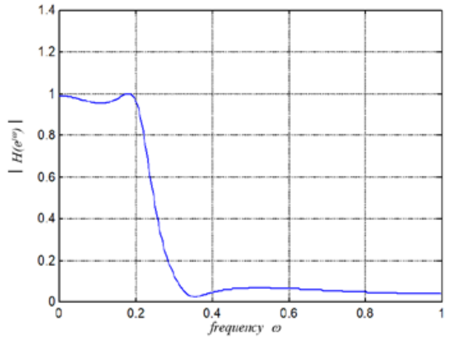

Figure 3. Lowpass filter

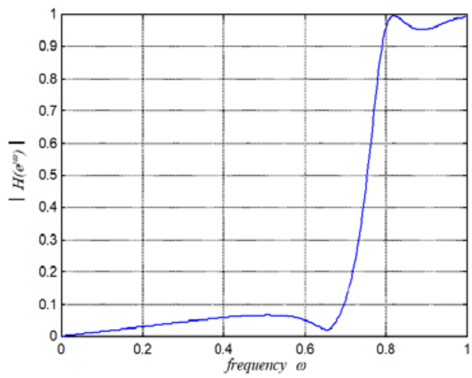

Figure 4. Highpass filter

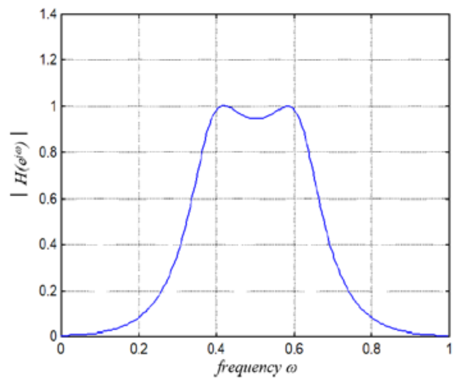

Figure 5. Bandstop filter

The findings of the experiments suggest that the genetic algorithm produces the optimal performance when constructing digital filters. GAs will ensure that the design criteria are always met, namely stability and compliance with tolerance limits obtained at the lowest filter order.

4. Conclusion

This research reveals an interesting experience in teaching genetic algorithms in designing optimal digital filters for all types, can be implemented. The important aspect of learning that greatly affects the results of filter design with genetic algorithms is the formulation of the evaluation function and determination of solution constraints. The more solution variables in the evaluation function, the GAs is rather difficult to find the most optimal value, although in general, the results obtained are quite adequate and inappropriate for the design criteria. Therefore, as far as possible the variables are minimized and the search space for solutions is tight, at least this will reduce the computational load and the length of execution time. In order to maintain the possibility that the selected population size is not the one that gives the most optimum results, the frequency of trials should be higher. The number of population and generation is directly proportional to the computational load and execution time. However, this study has proven that the performance of GAs to find the optimal solution is maintained. Meanwhile, students can not only design digital filters more easily but also apply genetic algorithms at once. Students will automatically acquire a number of hands-on skills that are important in their future.

Acknowledgements

Thanks to all parties who have facilitated and contributed to the implementation of this research, so that this article can be completed. Research activities are also supported by various educational programs that provide feedback and information about learning electronics engineering. Thanks also to lab assistant and operators who contributed with the work in the field.

References:

[1]. Abbasi, N. M. (2010). IIR digital filter design for low pass filter based on impulse invariance and bilinear transformation methods using butterworth analog filter. Retrieved from: https://12000.org/my_notes/IIR_digital_filter_design/index.htm [accessed: 12 March 2023]

[2]. Slowik, A. (2017). Efficient creation of population of stable Biquad sections with predefined stability margin for evolutionary digital filter design methods. In *International Conference on Artificial Intelligence and Soft Computing*, 451-460. Cham: Springer International Publishing.

[3]. Agrawal, N., Kumar, A., Bajaj, V., & Singh, G. K. (2021). Design of digital IIR filter: A research survey. *Applied Acoustics*, *172*, 107669.

[4]. Chen, T., Wang, Q., & Liu, H. L. (2018). FIR digital filter design based on evolutionary multi-objective algorithm. In *2018 14th International Conference on Computational Intelligence and Security (CIS)*, 349-352. IEEE.

[5]. Shao, P., Wu, Z., Zhou, X., & Tran, D. C. (2017). FIR digital filter design using improved particle swarm optimization based on refraction principle. *Soft Computing*, *21*, 2631-2642.

[6]. Penedo, S. R., Netto, M. L., & Justo, J. F. (2019). Designing digital filter banks using wavelets. *EURASIP Journal on Advances in Signal Processing*, *2019*, 1-11.

[7]. Liu, M., Chen, L., He, J., & Zhang, P. (2018). A novel evolutionary method of structure-diversified digital filter design and its experimental study. *Soft Computing*, *22*, 2381-2401.

[8]. Zebulum, R. S., Pacheco, M. A., & Vellasco, M. M. B. (2018). Evolutionary electronics: Automatic design of electronic circuits and systems by genetic algorithms. CRC press.

[9]. Liang, Y., & Wang, L. (2020). Applying genetic algorithm and ant colony optimization algorithm into marine investigation path planning model. *Soft Computing*, *24*(11), 8199–8210.

[10]. Elhoseny, M., Oliva, D., Osuna-Enciso, V., Hassanien, A. E., & Gunasekaran, M. (2020). Parameter identification of two dimensional digital filters using electro-magnetism optimization. *Multimedia Tools and Applications*, *79*(7), 5005–5022.

[11]. Boukhobza, A., Taleb, N., Taleb-Ahmed, A., & Bounoua, A. (2022). Design of orthogonal filter banks using a multi-objective genetic algorithm for a speech coding scheme. *Alexandria Engineering Journal*, *61*(10), 7649-7657.

[12]. Chandra, A., Kumar, A., & Roy, S. (2021). Design of FIR filter ISOTA with the aid of genetic algorithm. *Integration*, *79*, 107–115.

[13]. Ali, T. A. A., Xiao, Z., Sun, J., Mirjalili, S., Havyarimana, V., & Jiang, H. (2019). Optimal design of IIR wideband digital differentiators and integrators using salp swarm algorithm. *Knowledge-Based Systems*, *182*, 104834.

[14]. Meena, D. C., & Devanshu, A. (2017, January). Genetic algorithm tuned PID controller for process control. In *2017 International Conference on Inventive Systems and Control (ICISC)*, 1-6. IEEE.

[15]. Kumar, S., Jain, S., & Sharma, H. (2018). Genetic algorithms. In Nayyar, A., Le, D.N. & Gia, N. (Eds.) *Advances in swarm intelligence for optimizing problems in computer science*, 27–52. Chapman and Hall/CRC.

[16]. Hanne, T., Dornberger, R., Hanne, T., & Dornberger, R. (2017). *Computational intelligence*, 13-41. Springer International Publishing.

[17]. Nayyar, A., Garg, S., Gupta, D., & Khanna, A. (2018). Evolutionary computation: Theory and algorithms. In Nayyar, A., Le, D.N. & Gia, N. (Eds.) *Advances in swarm intelligence for optimizing problems in computer science*, 1–26. Chapman and Hall/CRC.

[18]. Waqas, H., Farooq, U., Khan, S. A., Alshehri, H. M., & Goodarzi, M. (2021). Numerical analysis of dual variable of conductivity in bioconvection flow of Carreau–Yasuda nanofluid containing gyrotactic motile microorganisms over a porous medium. *Journal of Thermal Analysis and Calorimetry*, *145*(4), 2033–2044.