Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985

Enhancing Efficiency of Digital Filters via Artificial Intelligence Algorithms

Hayder Kadhim Shareef Al-Safi[1*]

*Corresponding author E-mail: shhayder26@gmail.com

https://doi.org/10.46649/fjiece.v4.1.14a.25.3.2025

Abstract. *The "infinite impulse response" (IIR) digital filter was developed using the hybrid optimization artificial intelligence (AI) algorithm. At AI algorithm D-PSO, which stands for the "particle swarm optimization" dynamic topology, we make use of an approach that is both efficient and simple to memorize. The researchers investigated strategies to maximize the permissible ripples in a passband, the stopband, and the transition band while minimizing the nonlinear mean square error (MSE) between an initial filter response and the required filter response in those bands. Using the approach resulted in a considerable reduction in the square error rates of both the passband and the stopband, according to the findings of an experiment. The IIR digital filter square error values with the D-PSO are superior to those of digital filters using evolutionary AI algorithms such as the dynamic inertia PSO (DI-PSO) or the modified dynamic inertia PSO (MDI-PSO); this allows the IIR digital filter to achieve better results because the square error values depend on the sequence in which the filters are applied. Therefore, compared to digital filters and coefficients created by evolutionary ways, the ones produced by the suggested method are more precise and stable than those produced by evolutionary AI approaches.*

Keywords: *Artificial Intelligence Algorithms, IIR digital filter, Particle swarm optimization (PSO) Algorithm.*

1. INTRODUCTION

The phrase "signal processing" denotes manipulating and altering various signals. These signals may be sent via many mediums, including written text, music, and biometric data. Digital filters are a fundamental aspect of signal processing for the data under consideration. Both the "finite impulse response" FIR and the "infinite impulse response" IIR are terms used to describe digital filters based on "impulse responses.". Every single one of these classifications has a few positive qualities. In addition to the development of the first systematic approach that used linear programming, the IIR digital filter was simultaneously developed [1]. A filter was developed using a weighted least squares (WLS) method [2], implemented after the completion of operations to ensure an acceptable ripple response. An IIR filter used a method grounded on peak-constrained least squares (PCLS) criteria to attain its goals. The aim was to concurrently improve the group delay (τ) and the frequency response [3]. This task was executed to attain the intended results. Researchers have used the quadratic error function in developing IIR digital filters, as noted in [2]. The scholars above use various optimization techniques, including linear, nonlinear methods and Artificial Intelligence (AI) hybrid optimization technologies. The following examples exemplify techniques included under this category: Conic quadratic programming, Gauss-Newton method, sequential restricted least squares, frequency and phase response, and second-order cone programming.

Many evolutionary AI mechanisms have been used to tackle the issue of multimodal defect surfaces. The particle swarm optimization (PSOs), the artificial bee colonies (ABCs), the differential evolution algorithms (DEs), and the genetic algorithms (GAs) exemplify approaches within this area. However, this category is not limited to these techniques alone [4].

Two distinct outcomes are achieved due to the superior efficacy of PSO relative to other AI optimization methods [5]. Two benefits realized are decreased processing costs and expedited convergence. The results derived from solving nonlinear optimization problems across many applications with PSO substantiate this claim. One should choose a well-established optimization approach, such as the PSO technique. Since then, several proposals have been put out to enhance the PSO, all aimed at augmenting exploration opportunities. In [6] posited that including inertial weight (w) in IIR filter designs might enhance the scanning capabilities of these filters; this was supplementary to the methods that were previously used.

The quantization effect remains undesirable, even though the filter has been enhanced for greater sensitivity after the update. Moreover, PSO cannot address challenges involving more significant dimensions, necessitates individual tests, and often gets ensnared in suboptimal solutions. While simultaneous use of PSO is feasible, accurately determining the exact location and the accompanying velocity (v) data is not achievable. Particle Swarm Optimizations (PSOs) encounter many challenges, one being the potential for particles to converge on the local optimal solution at an overly quick rate. That is only one of the many concerns. Employing diverse neighbourhood topologies may somewhat mitigate this issue [7]. Particles in close contact may swiftly converge to provide the ideal solution. The study employs the D-PSO technique to achieve this aim, a version of the PSO methodology that accounts for dynamic topologies. D-PSO was created to improve exploration while optimizing the use of existing data in the search area. Another facet of D-PSO that enhances static performance is noted [8]. This feature involves assigning a substantial quantity of neighbours to each particle. Any of the two dynamic topologies may be used to attain this objective.

This work aims to improve the digital filter with defined stopband attenuation and passband ripple using the hybrid algorithm AI outlined in [7]. Furthermore, the digital filter, constructed with significant square error values, will be examined in detail.

The digital low-pass IIR filter inspires the design of D-PSO, one of the technologies applicable to this task. Another examination method calculates the digital filter's squared error (SE) values.

Simulation models are shown throughout the development of digital filters using optimization techniques to illustrate the strategy's effectiveness and compare the SE results. The D-PSO approach, DI-PSO, and MDI-PSO algorithms, together with the D-PSO strategy used in this study, may be accessed by referring to [9]. Currently, these algorithms are part of the collection and are accessible to users.

The following section will address the study conducted on IIR digital filters. Section 3 introduces the D-PSO algorithm and comprehensively describes the methodology. The fourth portion of the article explains the formulation of the stability constraints. Additionally, Section 5 encompasses a comprehensive analysis of the results, incorporating the simulation. In section 6, conclusions and future work are presented.

2. THE ANALYSIS OF DIGITAL FILTER

It is possible to explain an IIR-type digital filter using the Equation that is a linear and constant difference, as is seen in a sentence that follows [10], basic recipe Eq. (1)

$$\sum_{n=0}^{N} b_n, y[k_0 - n] = \sum_{n=0}^{M} a_n, x[k_0 - n] \qquad (1)$$

The output that matches the $x[k_0]$ input sequence is represented as $y[k_0]$, and the digital filter coefficients are a_n and b_n. In order to require frequency responses, the $y[k_0]$ and $x[k_0]$ coefficients serve as regulatory parameters in the z-domain digital filter transfer function, as shown in [10], and Eq. (2) and Eq. (3) are shown below:

$$Y(z)[1 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_K z^{-K} = X(z)[a_0 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_L z^{-L} \qquad (2)$$

$$\frac{Y(z)}{X(z)} = H(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_L z^{-M}}{1 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_N z^{-N}} \qquad (3)$$

The b_0 is set to 1, and z is equal to e^{jw}. Since this is the case, the study is all about finding the right values for these coefficients. This specific integer provides a decent filter response for the provided filter taps and is also resistant to quantization. Using the following procedures and the formula [7], one can calculate the digital filter phase response; this formula is quadratic Eq. (4).

$$\theta_{phase} = \arg\{H(e^{j\omega})\} \qquad (4)$$

3. PROPOSED ALGORITHM: THE D-PSO ALGORITHM

The dynamic mode of the "particle swarm optimization" PSO method is one example of a link between the PSO methodology and hybrid optimization strategies. This AI algorithm is a mixture of many optimization strategies. To do this, we use the standard PSO procedure, which creates a swarm of particles inside the search area. Each particle's neighbourhood, speed, and location are assigned at random. There is no fixed order for them. A subset of the swarm of particles may be described by a term called a neighbourhood. At every given moment in time, every single particle is accountable for evaluating a function in every process cycle. The present position of a particle will be taken as the new best if the fitness of an existing solution is greater than that of an existing personal best. Equations (5) and (7) are then used to change the particle's velocity and position, as seen in instances. The procedure is carried out again and again until the intended outcomes are realized [8].

Many features are shared between the D-PSO technique and the PSO standard method. Problems may be solved using any of these algorithms. Better results may be achieved with the D-PSO method since it modifies the design of dynamic variations used by the traditional PSO algorithm. The D-PSO method starts by seeing topology as interacting neighbours for each particle. Before everything else, this must be considered. Here, we can see the main difference between the two approaches. However, the point of dynamic extra designs is not to hasten convergence; it is to seek out space exploration because their

Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985

the intended function is to be put to use. The unique appearance of this design is one way in which it differs from its predecessors.

Furthermore, the technology is being used to generate the dynamic topology haphazardly throughout the method. While other dynamic implementations of the technique lose some of the exploitative qualities of the original PSO algorithm, the static topology preserves them all [8].

In the D-PSO, the particles are affected by utilizing the neighbourhood's best topology (*ni*). Also, the most fit is because, according to Equation (6), the velocities of all swarm members are continuously changing (*vi*). and its position (*pi*) according to Equation (7). The reason is that this is the case because the best fitness determines the topology of all neighbouring bests.

$$\vec{v}_i(\mathrm{t}) = \chi[\vec{v}_i(\mathrm{t}-1) + \emptyset_1, \mathrm{rand}[0.1]\big(\vec{p}_i(\mathrm{t}-1) - \vec{x}_i(\mathrm{t}-1)\big) + \emptyset_2, \mathrm{rand}[0.1]\big(\vec{n}_i(\mathrm{t}-1) - \vec{x}_i(\mathrm{t}-1)\big)\,] \qquad (5)$$

$$\vec{v}_i(\mathrm{t}) = \chi[\vec{v}_i(\mathrm{t}-1) + \emptyset_1, \mathrm{rand}[0.1]\big(\vec{p}_i(\mathrm{t}-1) - \vec{x}_i(\mathrm{t}-1)\big) + \emptyset_2, \mathrm{rand}[0.1]\left(\vec{d}_i(\mathrm{t}-1) - \vec{x}_i(\mathrm{t}-1)\right) \qquad (6)$$

$$\vec{x}_i(\mathrm{t}) = \vec{x}_i(\mathrm{t}-1) + \vec{v}_i(\mathrm{t}) \qquad (7)$$

Two variables are used within the equations above: *vi (t),* which represents velocity, and xi *(t),* which indicates the location of particle I during iteration t. Furthermore, the best solutions found thus far, which include static solutions, are represented by the variables pi (t) and di (t), respectively. In iteration *t*, an exhibit of *"i"* is a constriction coefficient χ of around 0.7298438. We are taking this action to ensure that the speeds do not increase. The symbols Ø1 and Ø2 indicate the acceleration coefficients, as stated in Equation (5). Particles of "i " scaling to *pi* and *ni*, respectively, are calculated using these coefficients. In standard PSO implementations, the acceleration coefficients are all the same and add up to 4.1. The values of Ø1 and Ø2 are found to be 2.05; This is about classical PSO. The acceleration coefficients Ø1 and Ø2 are used in equations (6) to ascertain the strength of the attraction between particle i's and pi (t) and di (t), respectively. The D-PSO method uses three acceleration factors because every particle is drawn to three different bests: an individual best, a best in motion, and a static best.

In contrast, the conventional PSO method often uses four acceleration factors. The starting values of these acceleration coefficients are 4.1/2, as shown by the symbols Ø1 and Ø2. The symbols stand for these coefficients. Based on the information provided by the D-PSO, we may conclude that 1.366666667 is the same as Ø1 and Ø2. Ordinary PSO implementations typically have acceleration factors totalling 4.1. A vector consisting of these arbitrary values ranging from zero to one is multiplied by each positional variable in the velocity equation. To spice things up a little, this is done. It may describe a range that guarantees all values in *vi* stay inside that range using the notation *[Vmin; Vmax]*. If we look within the search area, we can see that Vmin has the lowest value and Vmax has the highest value. The particle p in D-PSO is optimized by considering both *di (t)* instead of only one neighbourhood, which is best, *ni*. Doing so will increase the likelihood of it happening; this means that the dynamic topologies of particle p make possible the most successful solutions found so far. The optimization strategy utilized in this paper's pseudocode is detailed below [8], [11], [12].

Algorithm: D-PSO

1- The D-PSO algorithm.
2- The size of the swarm.
3- f, the target function for optimization
4- The max Iterations variable sets the maximum number of iterations.
5- Probabilities of reconstructing neighbourhoods that are still dynamic.
6- The Results:
7- The x_i, a spot where the least function value was found;
8- $f(x_i)$, the function's value at that spot
9- with "i = 1" through n,
10- Particle I should begin at a random place and speed. For each integer "*I*" from 1 to n, perform the following:
11- Let the number of iterations be less than the maximum number of iterations.
12- The optimum solution for particle " *i*" at the location
13- The p_i has not been found yet.
14- The d_i (t) is the optimal solution location determined by the current state of the particle "*i*'s" dynamic neighbourhood.
15- the modified speed of particle *"I"* as determined by equation (6)
16- the x_i = particle "*i's"* revised location from Equation (7)
17- As required, update the p_i (t), d_i (t), and x_i, and compute the function $f(x_i)$.
18- If the "random Double" is less than a probability of D,
19- restructure thriving communities
20- It should get back x_i and $f(x_i)$.

4. LIMITATIONS ON STABILITY

Extra precautions are required during the IIR digital filter development due to its increased instability risk. According to conventional wisdom, once the poles are located within the circle that encircles the z-unit plane, stability is guaranteed. Evolutionary AI algorithms carry out This repeated regression computation, which chooses each solution from the population matrix. Verifying the general consistency of all replies is the next stage. The first step of the study was categorizing higher-order transfer functions using the concept of first-order and second-order functions. The range of possible values for the coefficients contained in the denominator is limited using this strategy. A lower-order IIR filter may be designed [10] Using this method.

Because their lattice structure values were identical, the author of this paper used that feature instead of the polynomial-shortened version of the denominator [2].

There was yet another use of this method in that specific context. If one wants to solve the population matrix—also called the search space—one possible way to do so is by utilizing equation (8) [13].

The previously given Equation may be reformulated as Equation (8), where r is an element of the s et R, k is an integer between 1 and K and is elements set in the solution vector.

$$pop_{r.k}^{[n]} = [a_{r.0}\ a_{r.1}\ a_{r.2}\ \dots\ a_{r.M}: hr.M+1\ hr.M+2 \dots hr.M+N]^{[n]} \qquad (8)$$

The author of this research employed this information instead of the denominator that a polynomial had reduced; this occurred because their lattice structures had similar values. That particular setting also saw an approach quite similar to that one. Simplifying Equation (8) [2] yields one solution that may be used to solve the population matrix, sometimes called the search space. Another way to write the previously mentioned Equation (8) is where M is the set of elements that compose the solution vector, r is an element set R, and k is an integer between 1 and K. The updated version of the Equation that was earlier accessible is Equation (9) and Equation (10).

$$P_r(z) = P_{r-1}(z) + hr, z^{-1}, Q_{r-1}(z). r = 1.2. \dots . K-1 \qquad (9)$$

$$Q_r(z) = z^{-r}. P_r(z^{-1}) \qquad (10)$$

For any value of z for which Pr and Qr are integers equal to one, the conversion to the direct form of the denominator filter coefficients is only performed to calculate the frequency response using equations (1) and (2). The lattice-formed denominator filter coefficients are preserved during the exploration period. The solution vector's numerator filter coefficients are used to extract the lattice filter coefficients, the first step in the conversion process. We utilize the vector h to record these coefficients [14].

The fourth index, which is the initial element of h, is used to calculate Pr (z), and Pr-1 (z) and Qr (z) are both set to 1. The next step in obtaining the values of P2 (z) and Q2 (z) is to repeat the technique on the second component of the h, which is determined by the same method as the first element. The final *Q2 (z)* and *P2 (z)* calculations are completed.

Another step is to repeat the operation until all the components of h have been utilized. Following this, from [15], is a statement that shows that all the coefficients have been obtained by using the direct form of the polynomials of U(z), Equation (11):

$$[hr, M+N] \rightarrow [1\ b_r, 1 \dots\ b_r, n] \qquad (11)$$

The research and application of the CF methodology in recommender systems have been extensively explored and implemented [4]. There are primarily two categories of collaborative filtering: memory-based and model-based techniques [19][20]. Neighbourhood-based CF refers to the approach that leverages the system's rating matrix to estimate missing ratings for specific items. Conversely, model-based CF constructs a model using matrix values, which are subsequently employed to assess the relevance of new items to the target audience [19].

5. RESULTS OF EXPERIMENTS

One possible approach to creating an IIR filter is detailed in this section. Several design aspects are used to achieve the approach. Depending on the situation, the evolutionary method may provide a desired or realistic result. So, to find out how the concept works in practice, we have decided to run thirty separate simulations. The further analysis involves comparing the results to those of other previously studied situations. Parametric analysis, in accordance with the method described in [16], honed the most beneficial outcomes.

This paper presents an assessment of the strategy's success that takes into consideration a range of metrics, one of which is a square error (SE), stop error (e_s), and pass error (e_p) [17], [18] as in Equation (12):

$$SE = e_p + e_s \qquad (12)$$

An intensive investigation is required to acquire the control parameters for the methods utilized in this research. From what we have discussed so far, AI algorithmic systems may do better with a speed limit between -2 and 2. Moreover, the expected outcome was calculating the values controlling the local and global search parameters, denoted by the symbols Ø1 and Ø2, respectively.

Regardless, it was found that keeping v between 1 and -1 significantly improved performance throughout the experiment; this persisted throughout the whole simulation; this is why it has also been put to use in the pursuit of other research duties. The experiment takes into consideration both the size and cycle of the population. It is also possible to find the optimal values for the control parameters by distributing them among a large population. The D-PSO method may be fine-tuned with the following parameters: a maximum velocity of 2, a minimum velocity of -2, 60 swarm members, and 500 iterations. It may specify the proposed AI algorithm's control settings using these parameters.

Table 1 comprehensively explains the requirements for the various simulations that may be used to build filters. The first model considers a particle enhancement D-PSO and looks at the optimal inertial weights. Running the simulations 30 times for each weight and inertia sequence is required to find the optimal solution for low-pass filters. People may pick the best option by thinking about the operating range W and the minimal SE. We aim to provide the SE value that proved to be the most optimum for each sequence.

Table 1. Comprehensive Details on the Filter's Design.

Parameter	Value of Parameter
Filter order	4, 8, and 16
Type of filter	LPF
Stopband Sb	[0.5π, π]
Passband pb	[0, 0.4π]
Stopband of Ripple a_s	0.001
Passband of Ripple a_p	0.01

Furthermore, this analytical procedure was integrated into the subsequent simulation and executed continuously. This study investigated the effects of the modulation index D-PSO, which led to the discovery of the SEs of low-pass filters. Based on these results, it is clear that D-PSO is the superior computer software.

We need the characteristics listed in Table 1 to build a low-pass IIR digital filter prototype that might be utilized for D-PSO performance evaluations. The abbreviation LPF is often used to describe a low-pass filter. However, the words low-pass and LPF are equally accurate.

The dynamic DI-PSO algorithm in [9] and the MDI-PSO algorithm in [9], the SE values obtained while building a digital low-pass IIR filter are shown in Table 2.

Table 2. SE-Based Evaluation of LPF Achieved.

Order	DI-PSO	MDI-PSO	D-PSO (Proposed method)
4	0.389011	0.287613	0.211039
8	0.465778	0.21111	0.168395
16	1.927891	0.158887	0.156311

Figure 1 displays the steps involved in creating an IIR digital filter utilizing the MDI-PSO, DI-PSO, and D-PSO algorithms. Conventional regression on SE values achieves this aim. The filter order shows that compared to MDI-PSO and DI-PSO, the squared error (SE) values for IIR digital filters generated by the D-PSO method are much higher.

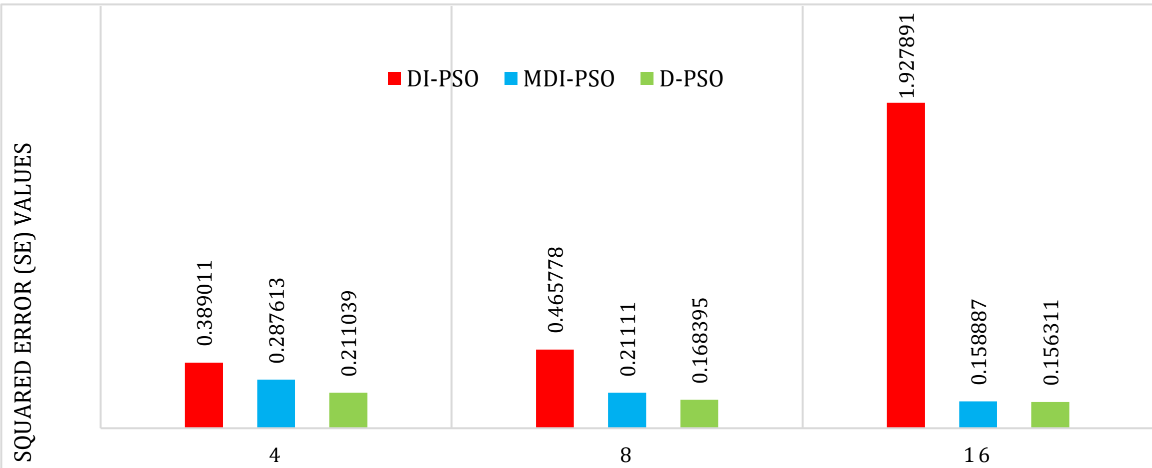

Fig. 1. Values For The Squared Error (SE) In The Results Scheme—Derived From Table 2.

Several stages of the study assessed the complexity of the methodology employed in this inquiry. Figure 1 compares the utilized version's performance to other versions, and Table 2 provides a numerical comparison between the two. Presented here are the SE values obtained with the maximum degree of accuracy for each of the thirty orders that were tested; this is because, compared to its forerunners, the DI-PSO and MDI-PSO algorithms, the D-PSO method produces far better low-pass IIR digital filters. Moreover, the D-PSO algorithm outperforms the original PSO method.

Figure 2 depicts the zeros and poles of the low-pass IIR digital filter based on the proposed D-PSO algorithm. Figure 2 shows the zeros and poles of the fourth-order low-pass IIR digital filter. This finding proves that the D-PSO method used in this research may provide a stable low-pass IIR digital filter, unchanged by changes to the filter order.

Al-Furat Journal of Innovations in Electronics and Computer Engineering (FJIECE)
ISSN -2708-3985

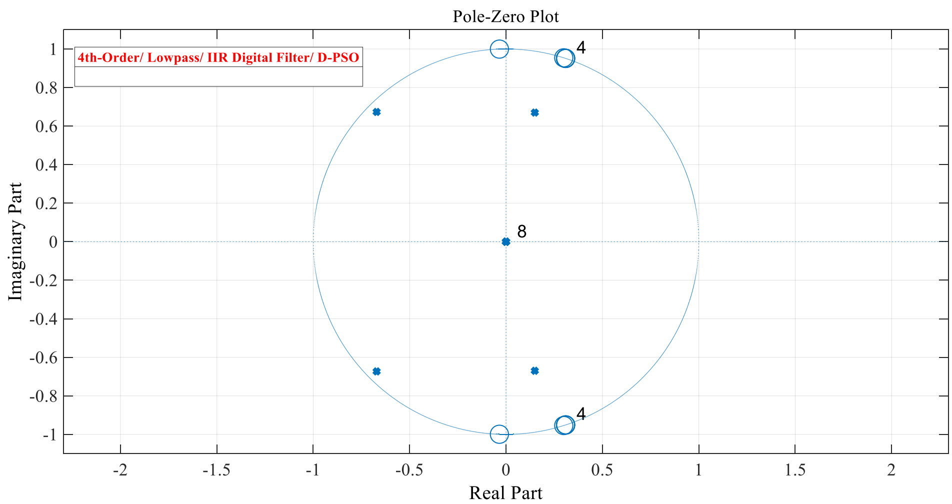

Fig. 2. The Digital Filter Used Here Is D-PSO's 4-Order Low-Pass IIR.

6. CONCLUSION

The creation of an IIR filter design is made possible via the AI hybrid technique. More specifically, this AI technique is founded on the D-PSO algorithm, an acronym that stands for dynamic-topology particle swarm optimization. The present approach has several notable benefits, including its resistance to dynamic topology displacements and limit cycle effect, as well as its adaptability for bigger filter taps. These are in addition to the many other advantages that the method has. The findings of the experiments indicate that the method that was used had an impact on the processes that were described before. The performance of the fidelity parameters has received a significant boost even though the complexity of the computations has increased somewhat. An additional possibility is that the same method may be used in the production of comb filters and fractional-order filters. Both signal filtering and preprocessing are possible applications for the proposed LPF. For example, the feature categorization and noise reduction procedures are examples of these applications.

REFERENCES

[1] N. T. Gadawe, R. W. Hamad, and S. L. Qaddoori, "Realization of IIR Digital Filter Structures for ECG Denoising.," *J. Eur. des Systèmes Autom.*, vol. 57, no. 2, pp. 599–608, 2024.

[2] B. Penumutchi and G. Tatikonda, "Systolic Digital IIR Filter Architectures for Real Time Signal Processing Applications," in *2024 5th International Conference for Emerging Technology (INCET)*, 2024, pp. 1–7.

[3] D. M. PERIŠIĆ, "NEW KIND OF IIR DIGITAL FILTERS INTENDED FOR PULSE PERIOD FILTERING," *Rev. Roum. DES Sci. Tech. ÉLECTROTECHNIQUE ÉNERGÉTIQUE*, vol. 69, no. 1, pp. 61–66, 2024.

[4] A. K. Roonizi, "Digital IIR filters: Effective in edge preservation?," *Signal Processing*, vol. 221, p. 109492, 2024.
[5] D. G. Lapitan, D. A. Rogatkin, E. A. Molchanova, and A. P. Tarasov, "Estimation of phase distortions of the photoplethysmographic signal in digital IIR filtering," *Sci. Rep.*, vol. 14, no. 1, p. 6546, 2024.
[6] S. Kocoń and J. Piskorowski, "Determination of Non-Zero Initial Conditions for IIR Notch Filters Using the Vector Projection Method with Minimum Delay," *Energies*, vol. 16, no. 4, p. 1702, 2023.
[7] S. Chauhan, M. Singh, and A. K. Aggarwal, "Designing of optimal digital IIR filter in the multi-objective framework using an evolutionary algorithm," *Eng. Appl. Artif. Intell.*, vol. 119, p. 105803, 2023.
[8] D. Sanchez, "DS-PSO: Particle Swarm Optimization with Dynamic and Static Topologies," 2017.
[9] N. Agrawal, A. Kumar, and V. Bajaj, "A New Method for Designing of Stable Digital IIR Filter Using Hybrid Method," *Circuits, Syst. Signal Process.*, vol. 38, no. 5, pp. 2187–2226, 2019.
[10] S. Singh, A. Ashok, M. Kumar, Garima, and T. K. Rawat, *Optimal design of IIR filter using dragonfly algorithm*, vol. 698. Springer Singapore, 2019.
[11] A. T. Alsahlanee, "Convergence Rate For Low-Pass Infinite Impulse Response Digital Filter," *J. Phys. Conf. Ser.*, vol. 1963, no. 1, p. 012103, 2021.
[12] A. T. Alsahlanee, "Digital Filter Performance Based on Squared Error," in *2021 International Conference on Advanced Computer Applications (ACA)*, 2021, pp. 74–79.
[13] Y. Dabas and N. Agrawal, "Improved Design and Optimization of IIR Filters by Cascading of ABC, PSO, and CA," in *Applications of Artificial Intelligence in Electrical Engineering*, IGI Global, 2020, pp. 332–343.
[14] N. Agrawal, A. Kumar, and V. Bajaj, "A new design approach for nearly linear phase stable IIR filter using fractional derivative," *IEEE/CAA J. Autom. Sin.*, vol. 7, no. 2, pp. 527–538, 2020.
[15] X. Tong, L. He, H. Du, and S. Dong, "A cascaded IIR filter with optimized group delay," *Proc. 14th IEEE Conf. Ind. Electron. Appl. ICIEA 2019*, no. 61674122, pp. 1990–1993, 2019.
[16] N. Agrawal, A. Kumar, and V. Bajaj, "Design of Infinite Impulse Response Filter Using Fractional Derivative Constraints and Hybrid Particle Swarm Optimization," *Circuits, Syst. Signal Process.*, vol. 39, no. 12, pp. 6162–6190, 2020.
[17] N. Agrawal, A. Kumar, and V. Bajaj, "Design of digital IIR filter with low quantization error using hybrid optimization technique," *Soft Comput.*, vol. 22, no. 9, pp. 2953–2971, 2018.
[18] E. S. Gopi, *Multi-Disciplinary Digital Signal Processing*. 2018.