Classification And Optimization Analysis Method of Digital Filter

Ruian Liu [1, a, †], Yiming Sun [2, *, †]

[1] School of Traffic and Electrical Dalian University of Science and Technology 116052, Dalian, China

[2] School of Optical-Electrical and Computer Engineering University of Shanghai for Science and Technology 200093, Shanghai, China

* Corresponding Author Email: 2035060725@st.usst.edu.cn, [a]liuruian-01@outlook.com

[†]These authors contributed equally

Abstract. The significance of digital filters in the field of digital signal processing has been increasingly emphasized due to the swift advancements in science, technology, and computing in recent years. At the same time, the optimization of digital filters has attracted the attention of many researchers. This paper describes in detail the classification and optimization methods of digital filters, and analyzes the advantages and disadvantages of several optimization algorithms. Among them, the parameters of the Genetic Algorithm are relatively complex. How to reduce the difficulty of calculation and ensure the performance of the filter is the direction of current research. Particle Swarm Algorithm works very well in the design process and parameter configuration, but it is still difficult to apply. This is also the direction of future research. Whether from the perspective of scientific research or practical application, the optimization of digital filters has a lot of room for development.

Keywords: digital filter, Genetic Algorithm, Particle Swarm Algorithm.

1. Introduction

The significance of digital filters in our everyday lives is on the rise due to the advancements in science and technology. Filtering is a fundamental and essential technique in signal processing which is used to extract desired signals from a variety of sources while filtering out unwanted interference signals. Filters are crucial components in the frequency domain analysis of a signal [1]. Digital filters can be found in many fields, such as communication industry, semiconductor industry, petrochemical industry, speech processing, image processing, etc. At present, in signal processing, the conventional algorithms of digital filters find extensive usage due to their widespread applicability, but they have limitations due to the high complexity of the system, the difficulty of adjusting parameters, the limited learning and adaptive functions, and the difficulty of grasping the noise and filter order [2].

Therefore, in this paper, based on the analysis of the two common digital filters Infinite Impulse Response (IIR)and Finite Impulse Response (FIR), the optimization methods such as genetic algorithm and particle swarm algorithm are further studied. The genetic algorithm is a method to continuously find the global optimal solution by avoiding the local optimal solution, and this paper uses simulation to obtain its spectral characteristics. The particle swarm algorithm searches for the optimal solution by continuously changing the particle position and velocity in the solution space, and this paper also uses simulation to make its spectrum more intuitive. Finally, the application of two optimization algorithms in digital filters is also analyzed.

2. Digital filter classification and principal analysis

2.1. Principle and formula of digital filter

The difference equation for the digital filter is expressed as formula 1.

$$y(n)=\sum_{k=1}^{N}a_k y(n-k)+\sum_{k=0}^{M}b_k x(n-k) \tag{1}$$

The system of a digital filter is discrete in nature, and it operates on the input discrete signal using a set of operations to extract the necessary information from the input signal. Typically, the system function of a digital filter is represented as in formula 2.

$$H(z)=\frac{Y(z)}{X(z)}=\frac{\sum_{k=0}^{M}b_k z^{-k}}{1-\sum_{k=1}^{N}a_k z^{-k}} \tag{2}$$

a_k and b_k are the filter coefficients. When a_k =0, the output of a non-recursive filter (FIR) only relies on the input and not the previous output, with M representing the filter's order. On the other hand, a recursive filter (IIR) relies on both the input and the previous output, with N representing the order of the filter.

FIR digital filters and IIR digital filters are the two categories into which digital filters are classified, based on the nature of their impulse response. From the point of view of the equation, the coefficients a_k of an FIR digital filter are always zero beyond a certain point, whereas the coefficients a_k of an IIR digital filter can have non-zero values at all points.

2.2. IIR Digital filter

An IIR filter is a filter used to process a signal that produces a continuous time response. This filter can be used to achieve frequency specific signal processing.

IIR digital filters can be broadly classified into two types - low-pass filters and high-pass filters. The implementation of a low-pass filter can effectively reduce the impact of noise, whereas a high-pass filter can accentuate the high-frequency components of a signal.

2.2.1. Direct Type IIR Filters

Equation 2 can be viewed as the result of multiplying two system function like formula 3.

$$H(Z)=H_1(z)H_2(z) \tag{3}$$

where H_1 contains all the zeros and H_2 contains all the poles. They can be represented as equation 4 and 5.

$$H_1(Z)=\sum_{k=0}^{M}b_k z^{-k} \tag{4}$$

$$H_2(z)=\frac{1}{1-\sum_{k=1}^{N}a_k z^{-k}} \tag{5}$$

The system shown in Figure 1 is obtained by cascading two systems, the all-zero system H_1 and the all-pole system H_2 [3].

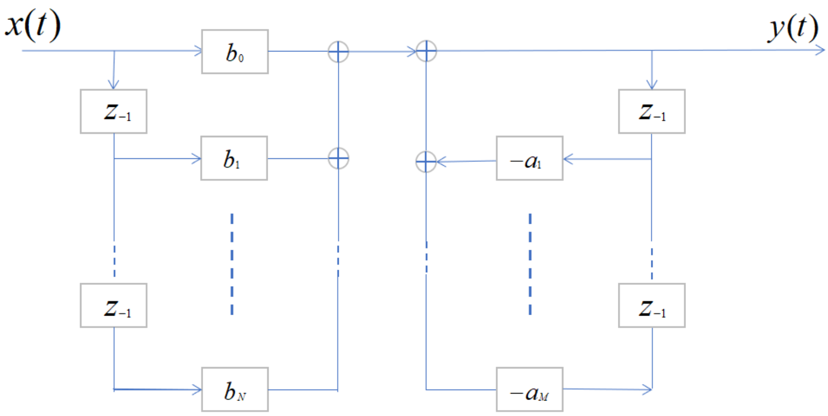


Figure 1. Structure diagram of IIR direct type filter

To implement this, *M+N+1* multiplications, *M+N* additions, and *M+N+1* storage spaces are necessary.

2.2.2. Cascaded and parallel IIR filters

In the digital filter system function given in Equation. 2, it is assumed that $N \geqq M$, which can be decomposed into multiple second-order subsystem cascades of the form as equation 6.

$$H(z) = \prod_{k=1}^{K} H_k(z) \tag{6}$$

K is the integer part of *(N+1)/2* and the general form of H_k can be expressed as formula 7.

$$H_k(z) = \frac{b_{k0} + b_{k1} z^{-1} + b_{k2} z^{-2}}{1 + a_{k1} z^{-1} + a_{k2} z^{-2}} \tag{7}$$

The general form structure of the cascade type is shown in Figure 2.

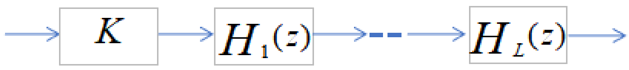


Figure 2. Cascade type structure of H(z)

Each of these second-order subsystems can be implemented in direct type. When $N \geqq M$ and the poles are independent of each other, The system function can be expressed as a product of multiple factors in the form of formula 8.

$$H(z) = C + \sum_{k=1}^{N} \frac{A_k}{1 - p_k z^{-1}} \tag{8}$$

p_k is the pole, A_k is the coefficient of the partial fractional expansion, and *C* is a constant term, but in the general case, the partial pole of *H(z)* may be complex, when A_k is likewise complex, and the second-order subsystem has the form like equation 9.

$$H_k = \frac{b_{k0} + b_{k1} z^{-1}}{1 + a_{k1} z^{-1} + a_{k2} z^{-2}} \tag{9}$$

At this point, the entire system function can be expressed as a product of two factors, {a_{ki}} and {b_{ki}}, as shown in the equation 10.

$$H(z) = C + \sum_{k=1}^{K} H_k(z) \tag{10}$$

K is the integer part of $(N+1)/2$, and the parallel-type structure is shown in Figure 3 [4].

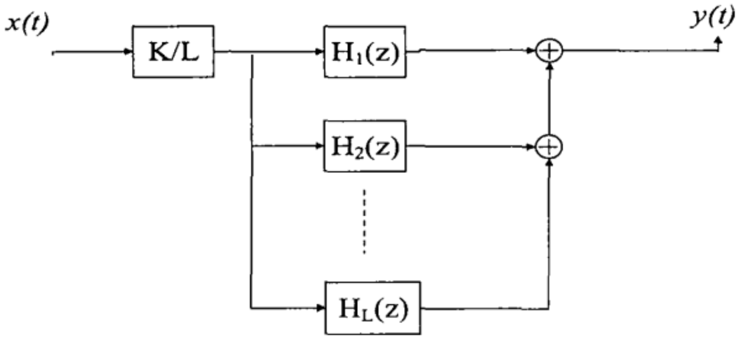


Figure 3. Parallel type structure

Because the parallel structure is fast, less sensitive to the coefficients of the filter, the zero-pole quantization error of each second-order system does not affect each other, and there is no combination in the cascade system and the connection order, so it is more widely used in practical applications.

IIR digital filters are characterized by the following features:

(1) The unit impulse response, $h(n)$, of an IIR digital filter is infinite in length, meaning that it extends indefinitely in both the positive and negative directions. This is due to the recursive nature of the filter, where the output of the filter depends on its previous outputs as well as the input signal.

(2) The system function $H(z)$ has poles in the finite z-plane.

(3) They have a recursive structure, which means they have output-to-input feedback. The design of the IIR filter is to determine the order N and the coefficients $\{a,b\}$ of the filter under the given technical specifications. The lower the order of the filter, the lower the cost of implementation.

When designing IIR filters, the most common approach is to use analog filters to design digital filters. The reasons for this are:

(1) The design techniques of analog filters are relatively mature and can be widely utilized;

(2) A large number of reference procedures and tables are available for analog filters;

(3) Its solution can be in closed form.

2.3. FIR digital filter

The FIR digital filter, also known as the finite-length impulse response filter, is characterized by the following features:

(1) The unit impulse response $h(n)$ of the system is non-zero for finite values of n.

(2) The system function H(z) of an FIR digital filter converges at $|z|>0$ and has zeros only at $|z|<=0$. This means that there are no poles in the finite z-plane, and all zeros are located at $z=0$ (indicating a causal system).

(3) An FIR digital filter is a type of digital filter where the impulse response of the filter is finite and has non-zero values only within a finite range. Unlike IIR digital filters, FIR filters do not have output-to-input feedback, meaning that the output of the filter depends only on the current and past input samples. However, some FIR filter structures (such as frequency sampling structures) may contain a recursive feedback component. The system function $H(z)$ of an M-order FIR filter is shown in formula 11.

$$H(z) = \sum_{k=0}^{M} h[k] z^{-k} \tag{11}$$

H(z) is an M-order polynomial in *z*, where *M* is the number of filter coefficients. *H(z)* has *M* zeros in the finite z-plane, which correspond to the frequencies at which the filter attenuates or completely blocks the input signal. Since FIR filters do not have output-to-input feedback, there are no poles in the finite z-plane. The z-plane. FIR filter has the following advantages compared to IIR:

(1) FIR filters are particularly well-suited for digital signal processing tasks due to their regular internal logic array and abundant connection resources. In contrast to general-purpose DSP chips that prioritize serial operation, FIR filters offer superior parallelism and scalability. By utilizing the FPGA's fast multiplicative accumulation algorithm, it is possible to design high-speed FIR digital filters.

(2) FIR filters offer the advantage of being able to increase accuracy indefinitely (provided there is sufficient computing power), and they do not suffer from the phase accuracy issues that can plague IIR filters.

As a result, FIR filters are currently considered a high-end solution. Nonetheless, there are still several disadvantages associated with FIR filters, such as:

(1) Due to the high level of precision utilized, there is an increase in computing resources and memory required, resulting in higher power consumption.

(2) FIR filters are commonly employed in various fields to address high-frequency issues. However, in audio applications, signals with frequencies below 1KHz are often encountered, and in order to achieve satisfactory results for such signals, a FIR filter with a minimum order of 512 is typically required.

(3) When attempting to address low-frequency issues using FIR filters, there may be an excessive amount of arithmetic required due to the inability to adjust the width of each processing unit, resulting in high-frequency components being affected as well.

2.3.1. Cross-cut FIR filter

The output of an FIR digital filter is obtained by convolving the input signal with the filter's impulse response, which is a sequence of filter coefficients. The filter coefficients are also known as weights and determine the filter's frequency response. As shown in figure 4.

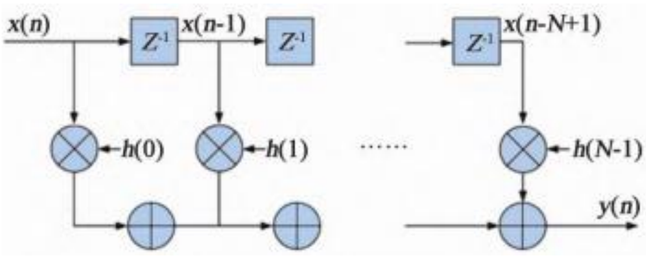
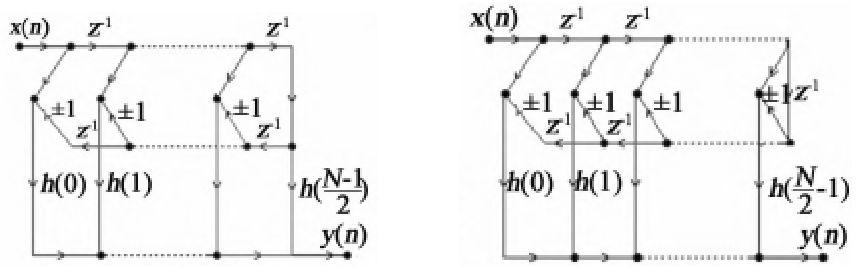


Figure 4. Cross-sectional FIR structure diagram

Each input sample is multiplied by a corresponding weight, and the products are then summed up to produce the output sample. *h(n)* has symmetry, one advantage of FIR filters is that they exhibit a desirable linear phase response, and the simplified transverse structure is shown in Figure 5(a) and (b) for odd and even numbers of *N*, respectively [5].



(a) (b)

Figure 5. Linear phase when *N* is odd or even: (a) N is an odd number, (b) N is an even number

2.3.2. Cascaded FIR Filters

By decomposing *H(z)* into second-order factorial products of real coefficients, the expression is given as equation 12.

$$H(z)=\sum_{N=0}^{N-1} h(n)\, z^{-N}=\prod_{k=1}^{\frac{N}{2}} \beta_{0k}+\beta_{1k} z^{-1}+\beta_{2k} z^{-2} \tag{12}$$

N is an even number when one of the $\beta_{2k}=0$ (*N-1* zeros).

The FIR filter is implemented using a second-order cascade structure, with each second-order section using a transverse truncation type structure, as shown in Figure 6 [6].

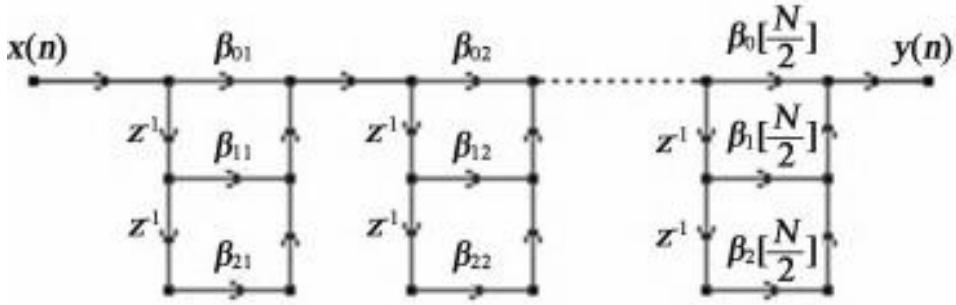


Figure 6. Cascade structure of FIR filter

3. Optimization Algorithm

Although the traditional digital filter algorithm is widely used in signal processing, it also has some limitations and shortcomings. For example, the complexity of the system is high: because the algorithm of the traditional digital filter is usually based on a mathematical model, it takes a lot of time and energy to perform steps such as model selection and parameter optimization in practical applications, which makes the system complex. It is difficult to adjust parameters: In traditional digital filters, the filter algorithm usually requires manual adjustment of parameters to achieve the optimal filtering effect, but this requires professional skills and experience and is not easy to master. Limited learning and adaptive functions: Traditional digital filter algorithms usually have fixed parameters and structures, which cannot be learned and adaptively adjusted for different signal scenarios. It is difficult to grasp the noise and filter order: the order of traditional digital filters usually needs to be designed according to the system requirements, and the filter coefficients often need to be adjusted through experience, and the suppression of noise may be limited by the filter Orders and Coefficients. These shortcomings make the application of traditional digital filters have certain limitations, and it is necessary to develop more efficient and flexible digital filter algorithms.

3.1. Genetic algorithm

3.1.1. The principle of genetic algorithm

The Genetic Algorithm (GA) is a computational approach that simulates biological processes, serving as a powerful tool for studying various phenomena. The algorithm's fundamental principle closely aligns with the evolutionary mechanisms observed in nature, facilitating a randomized global search optimization technique. This method draws inspiration from the theories of evolution proposed by Darwin and genetics pioneered by Mendel, integrating their conceptual frameworks into its design. This algorithm is quick and parallel, and while determining the answer, it adaptively seeks out the best option.

Genetic algorithm is to treat each chromosome as a solution of genetic algorithm. Finding the most suitable solution from many genomes is the core of genetic algorithm. The advantages and disadvantages of this solution are determined by the fitness function. measure. We can regard all solutions as a multi-dimensional surface, as shown in Figure 7 [7].

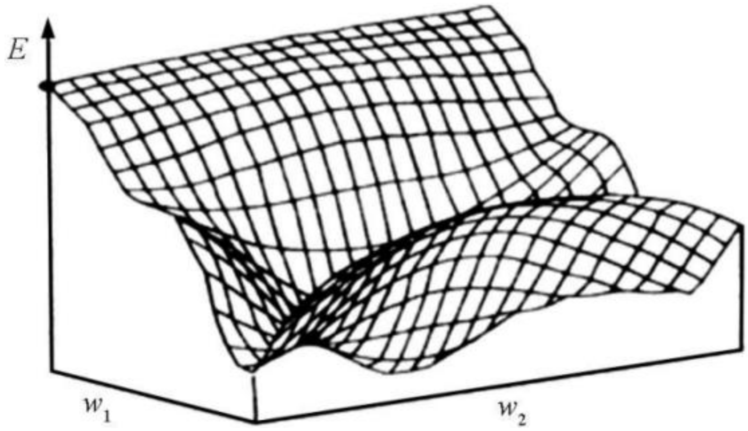


Figure 7. Multidimensional surface

There are many peaks and valleys in this surface. The peak corresponds to the local optimal solution, and the highest peak in the entire multidimensional surface is the global optimal solution. The tallest peak on the whole multidimensional surface is the global optimum solution, while the peak corresponds to the local optimal solution. In the same way, if you are looking for a problem with a small fitness evaluation, you must look for the deepest trough in the multidimensional surface, not the local trough.

Genetic algorithm is similar to the evolution in nature. First, it is necessary to encode the population and initialize a population randomly. Then use the fitness function to evaluate each individual, use the selection function to select according to the regulations, simulate the natural law, and make the individual gene mutate. Observe the resulting next-generation features, and finally draw conclusions. One potential benefit of utilizing the genetic algorithm lies in its inherent capacity for generating solutions that are deemed optimal, obviating the need for explicit search by means of continuous iteration. Such an attribute may prove advantageous in certain contexts within the realm of optimization. The steps of the genetic algorithm are as shown in Figure 8 [8].

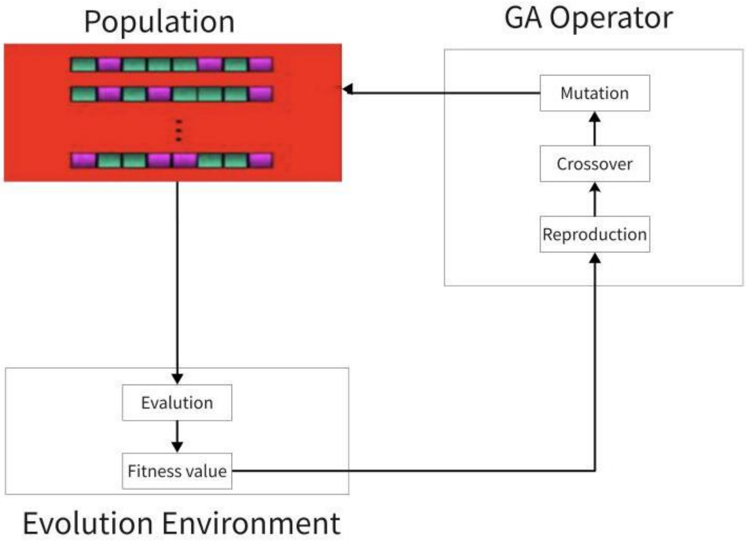


Figure 8. Steps of Genetic Algorithm

Start the cycle. 1.Assess the fitness of each chromosome within the population.2. The selection probability of an individual is directly proportional to its fitness. Consequently, two individuals are chosen as parents from the population. 3.Perform crossover, wherein the parental chromosomes are paired and combined to produce the next generation of chromosomes. 4.Introduce mutation to the chromosomes of the next generation, further enhancing genetic diversity. 5.Iterate through steps 2, 3, and 4 until a new population is generated, signaling the completion of one cycle.

3.1.2. Application of genetic algorithm in digital filter

As one of the classic algorithms of heuristic algorithms, the core of Genetic Algorithms (GA) is to gradually improve the quality of a set of candidate solutions through variation and selection. Since the digital filter design was first proposed to use the genetic algorithm to design the digital filter, there have been more than one hundred kinds of schemes. To reduce the complexity of the digital filter, the weighted approximation error is used to make the weighted ripple value of the passband and stopband extremely small. Compared with the simulated annealing method, the results show that the genetic algorithm has a small amount of calculation. In literature [9], a set of candidate solutions is obtained from the research problem at the time of design, the adaptability of the solutions is judged according to the favorable conditions, and finally the candidate solutions are calculated to obtain the results. First pass for formula (13)

$$E=\sum_{i=1}^{M}\left[\left|H\left(e^{jw}\right)\right|-\left|H_d\left(e^{jw}\right)\right|\right] \tag{13}$$

Where E means square error, where $H(\mathrm{e}^{\mathrm{j}w})$ means the amplitude-frequency response of the filter, where $H_{\mathrm{d}}(\mathrm{e}^{\mathrm{j}w})$ means a given amplitude-frequency response

Determine the minimum mean square error. then pass for formula (14)

$$\left|A_0\right|=\frac{\sum_{i=1}^{M}\left|G\right|\left(e^{jw}\right)\cdot\left|H_d\left(e^{jw}\right)\right|}{\sum_{i=1}^{M}\left|G\left(e^{jw}\right)\right|^2} \tag{14}$$

Where A_0 means the best gain, and where G means the optimization variable get the best gain A_0. Figure 9 shows the spectrum characteristics obtained after simulation [9].

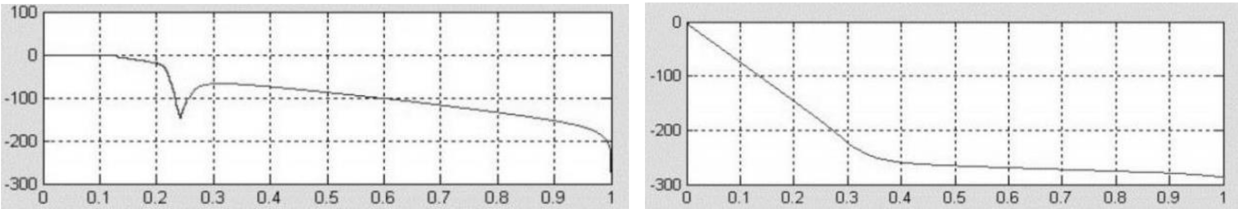


Figure 9. Spectrum characteristics

It can be seen that the algorithm can use relatively simple codes to express complex structures, and can adjust the direction adaptively by simulating genetic operations on one of the codes, simplifying the difficulty of calculation.

The literature [9], the author summarizes the characteristics of the genetic algorithm:

(1) Self-organization, adaptation, and self-learning are essential components of the genetic algorithm. Upon determining the coding scheme, fitness function, and genetic operator, the algorithm employs the information gathered in the evolutionary process to self-regulate the search. This is achieved by adhering to the natural selection strategy, where individuals with higher fitness levels have a greater likelihood of survival, and less fit individuals are eliminated.

(2) In the genetic algorithm, there is no requirement for prior knowledge of the search space or other conditions, but the fitness function is utilized to assess and select individuals for genetic research. Consequently, the domain of definition for this algorithm can be set arbitrarily, allowing for a wide range of potential applications.

(3) Intrinsic parallelism is a prominent feature of genetic algorithms, which takes two forms. Firstly, it is parallel within the algorithm itself. Genetic algorithms demonstrate seamless compatibility with contemporary parallel devices and distributed systems, while exhibiting minimal impact on parallel efficiency. Secondly, the parallelism is implicit, as population-based search organization makes it possible to process multiple areas in the space concurrently and exchange information between them. This results in greater benefits realized through reduced calculation effort. The algorithm employs probabilistic transition rules, rather than deterministic ones, to orient the search process towards a more optimal part of the search space. As a result, the search direction is unambiguous and clear.

3.2. Particle Swarm Algorithm

3.2.1. Principle of particle swarm algorithm

Particle swarm optimization (PSO) is an optimization algorithm grounded in swarm intelligence, drawing inspiration from clustering phenomena observed in natural systems such as avian flocks and fish schools. PSO aims to discover the optimal solution by iteratively exploring the search space. Its fundamental principle involves continuously adjusting particle positions and velocities within the solution space to facilitate objective function optimization and other parameter tuning challenges. The implementation process of the PSO algorithm commences with the random initialization of a set of particles, with each particle representing a group of potential solutions. Subsequently, particle fitness is evaluated by assessing the quality of each particle's performance through the objective function. Particle velocities and positions are updated based on historical, group, and global optimal positions. Furthermore, the algorithm maintains and updates historical and swarm optimal positions for each particle. Ultimately, a termination condition is assessed to determine whether to cease the search and return the optimal solution. By iteratively refining its parameters, the PSO algorithm effectively converges towards the global optimum [10]. In comparison to alternative optimization algorithms, PSO exhibits noteworthy advantages in terms of rapid convergence and straightforward implementation. The flow of the particle swarm optimization algorithm is shown in Figure 10.

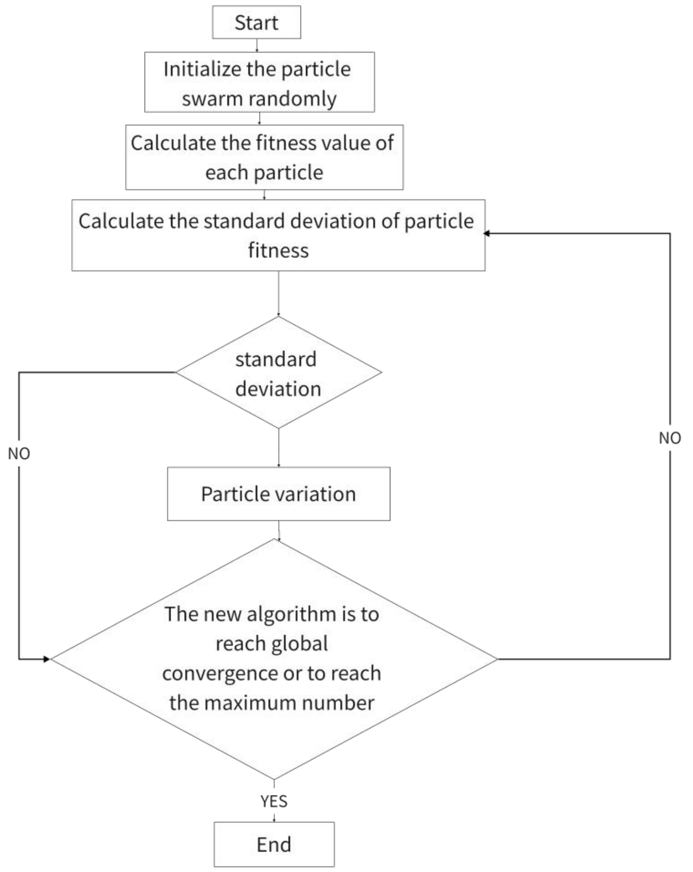


Figure 10. The flow of particle swarm optimization algorithm

3.2.2. Application of Particle Swarm Optimization Algorithm in Digital Filter

The literature [11], the author described that the particle swarm optimization algorithm is a parallel global random search algorithm. This algorithm is rooted in the simulation of a simplified social model and represents a prominent optimization method within the domain of swarm intelligence. It draws inspiration from the foraging behavior exhibited by birds, which serves as a fundamental principle in its design. Conceptually, one can envision a scenario where a group of birds is engaged in a random search for food within a defined area, with a sole piece of food available. In this setting, the birds possess no knowledge of the exact location of the food, but are aware of the distance between their current positions and the food source. The task at hand is to determine the optimal strategy for food acquisition [11]. The most straightforward and efficient approach involves searching the vicinity of the bird that is currently closest to the food. Notably, researchers have observed that bird flocks exhibit collective behaviors characterized by periods of aggregation and dispersion during their movements. Although the behaviors of individual birds are unpredictable, the flock maintains a remarkable level of coherence, facilitated by cooperative interactions and information sharing. These characteristics underpin the evolutionary potential within the flock. Drawing inspiration from the biological behaviors of birds, the Particle Swarm Optimization (PSO) algorithm has emerged as a widely employed method for solving diverse optimization problems. In PSO, the particle swarm originates from a group.

Every individual particle within the system possesses a fitness value that is determined by the optimization function. Moreover, each particle is accompanied by a velocity parameter that regulates both the direction and magnitude of its flight trajectory. Additionally, particles store information regarding the current optimal solution and navigate the solution space accordingly. It is important to note that each iteration of the algorithm is not entirely random; upon discovering a superior solution, it becomes the basis for seeking the subsequent optimal solution. PSO algorithm commences by initializing a collection of particles with random positions. Subsequently, it proceeds with multiple iterations in order to discover the optimal solution. During each iteration, the particles undergo selfupdates by monitoring two crucial "extreme values." Firstly, a particle tracks its own optimal solution, represented as an individual extreme point denoted by p best. This solution reflects the particle's best-found position within the search space. Furthermore, the particles consider the optimal solution that has been discovered by the entire population, referred to as the global extreme point, represented by g best. This value captures the best-found position achieved collectively by all particles in the population. Finally, the designed filter is simulated and tested. The sinusoidal signals containing 20HZ, 30HZ, and 40HZ are superimposed as the signal to be processed. The superimposed signal is shown in Figure 11, and the frequency spectrum is shown in Figure 12. The filtered waveform is shown in Figure 13 [11].

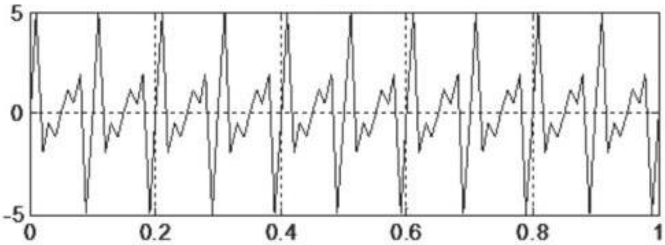


Figure 11. Superimposed signal

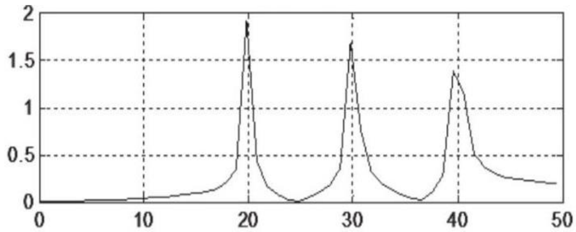


Figure 12. Spectrum after superimposition

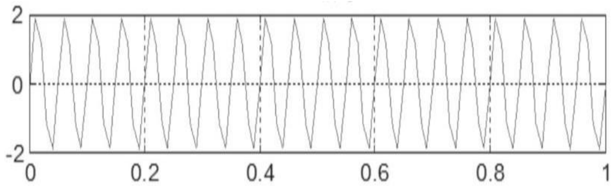


Figure 13. Filtered 20HZ signal

According to the simulation results, the filter designed by this method has the advantages of small algorithm calculation, fast speed, short running time, small passband fluctuation, and large stopband attenuation.

In the literature [12], the author described that the Particle Swarm Optimization algorithm (PSO) is an optimization method that draws inspiration from the collective foraging behavior exhibited by birds. The feasible solution space of an optimization problem can be envisioned as a point residing within a multi-dimensional search space. In the context of particle swarm optimization, particles correspond to candidate solutions, each possessing an adaptation value determined by the objective function. Additionally, particles are characterized by a speed value, dictating both their flying direction and distance. Consequently, particles follow the optimal particle that has been identified thus far within the search space, in order to efficiently navigate towards an optimal solution as shown in Figure 14 [12].

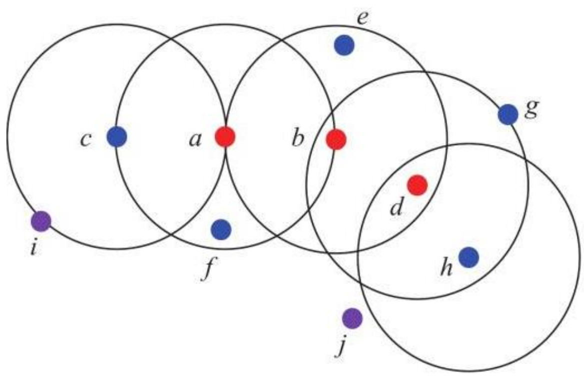


Figure 14. Schematic diagram of particle swarm

Figure 15 illustrates the effect of a low-pass filter on particle trajectory, with the input represented by the dotted line and the output by the solid black line. The filter is observed to exert a smoothing influence on particle movement, as demonstrated by the altered trajectory. It can be seen from Figure 16 that the larger the value of mc is set, the better the smoothing effect will be. In the early stage of the PSO algorithm, in order to prevent the particle from oscillating and retain the diversity of the particle in the later stage, the particle can perform a fine local search. A larger mc value can be set in the early stage, and a smaller mc value can be set in the later stage of the search. The value of mc in Figure 15 is 0.98. The value of mc in Figure 16 is 0.9 [13].

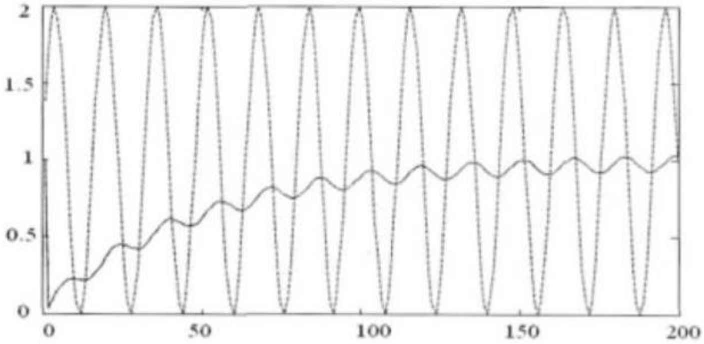


Figure 15. mc=0.98

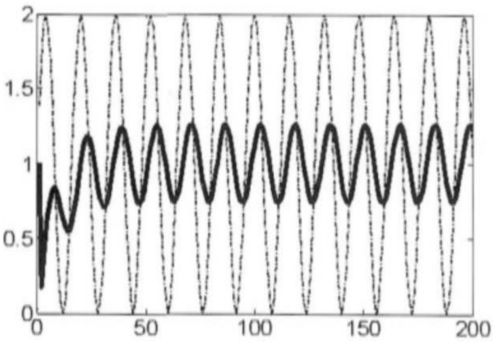


Figure 16. mc=0.9

Due to the absence of mechanisms such as crossover and mutation in the particle swarm algorithm, a phenomenon known as "premature convergence" occurs when a particle discovers a local optimal solution. In this scenario, other particles are subsequently attracted towards the identified optimal solution and rapidly converge in its vicinity. This is also the research direction that this algorithm needs to improve in the future.

4. Conclusion

Through the study of existing articles, the core of genetic algorithm is to gradually improve the quality of a set of candidate solutions through variation and selection. The complexity of the digital filter can be reduced by using the genetic algorithm, and the weighted approximation error is used to make the weighted average value of the passband and stopband extremely small. The disadvantage is that the convergence rate is full and premature convergence when there are algorithms that sometimes find local optimal solutions. Although these problems can be overcome by mixing other algorithms, it increases the difficulty in calculation and the iterative process is more complicated.

Particle swarm optimization is a population stochastic optimization algorithm based on social psychology principles, which can deal with multi-objective function optimization problems. It has fewer setting parameters, faster convergence speed, less calculation amount, strong search ability and simple structure and easy implementation. The disadvantage is that due to the fewer parameters set by the algorithm, the digital filter can be accurately optimized only by improving the inertia weight and learning factor.

In the future development, the genetic algorithm has been deeply explored in the direction of encoding method, determination of control parameters, selection method and crossover mechanism. Dynamic strategy and adaptive strategy are introduced to improve the performance of genetic algorithm, and various variants are proposed. genetic algorithm. For instance, one approach to address this issue involves modifying the structure of the genetic algorithm or employing specific techniques tailored to the problem's unique characteristics. To mitigate the premature convergence phenomenon, a hybrid genetic algorithm is employed. Dynamic adaptive technology is leveraged to dynamically adjust the algorithm's control parameters and coding granularity throughout the evolutionary process. Or use parallel genetic algorithm. In terms of particle swarm optimization, researchers have proposed various improved learning factor adjustment techniques from different angles, including constant method, compression factor method, trigonometric function method, and learning that changes synchronously or asynchronously at any time based on the dynamic weight method. factor method, etc.

Through the above research, the parameters of the genetic algorithm are relatively more complicated than other algorithms when designing the filter, so how to reduce the complexity of the calculation while ensuring the performance of the filter is the direction of continued efforts. Particle swarm optimization algorithm shows good results in design, but there is still a big gap from practical application, which is also the direction of future research.

References

[1] Y. Hao, “Research on Optimal Design of Digital Filter,” Changsha University of Science and Technology, 2012.

[2] L. Lian and Z. Tian, “FIR Digital Filter Design Based on Improved Artificial Bee Colony Algorithm,” Soft Computing, vol. 26, no. 24, 2022.

[3] V. E. Debrunner and A. A. Louis) Beex, “Sensitivity Analysis of Digital Filter Structures,” SIAM Journal on Matrix Analysis and Applications, vol. 9, no. 1, 2006.

[4] M. Wang, “Design of IIR Digital Filter Based on FPGA,” Southeast University, 2016.

[5] X. Shi and D. Yan, “Design and Simulation of Digital Filter Based on FPGA,” Railway Locomotive and EMU, vol. 06, pp. 21 - 24+39+5, 2022.

[6] O. C. Coşkun and K. AVCI, “FPGA Schematic Implementations and Comparison of FIR Digital Filter Structures,” Balkan Journal of Electrical and Computer Engineering, vol. 1, 2018.

[7] Q. Gu and L. Ma, “Design and Simulation of IIR in the Optimization of Digital Filter Performance Indicators,” Jiangsu Science and Technology Information, vol. 29, pp. 41 – 43, Oct. 2017.

[8] B. Chen, “Realization of Digital Filter Based on Genetic Algorithm,” Nanjing Forestry University, 2009.

[9] F. Zhou and Z. Liao, “Design of grouping parallel particle swarm optimization algorithm for FIR filter,” Electronic Technology, vol. 46, no. 02, pp. 62 – 65, 2017.

[10] D. Song, “Optimization a Simulation of Digital Filter Based on Particle Swarm Optimization,” Computer Simulation, vol. 30, no. 08, pp. 356 - 359+375, 2013.

[11] R. Kar, D. Mandal, S. Mondal, and S. P. Ghoshal, “Craziness based Particle Swarm Optimization algorithm for FIR band stop filter design,” Swarm and Evolutionary Computation, vol. 7, pp. 58 – 64, Dec. 2012.

[12] D. Wang, D. Tan, and L. Liu, “Particle swarm optimization algorithm: an overview,” Soft Computing, vol. 22, no. 2, pp. 387 – 408, Jan. 2017.

[13] Y. Zhang, S. Wang, and G. Ji, “A Comprehensive Survey on Particle Swarm Optimization Algorithm and Its Applications,” Mathematical Problems in Engineering, vol. 2015, pp. 1 – 38, 2015.