ISSN: 2688-8653

Design of FIR Digital Filter based on Particle Swarm Optimization Algorithm

Li Wang*, Kaifei Bai and Wei Wang

Department of Electronic Engineering, Xi'an Aeronautical University, Xi'an Shaanxi 710077, China

Abstract

This paper studies the application of particle swarm optimization algorithm in the design of digital filters. According to the design standards and guidelines of digital filters, combined with the design steps of particle swarm algorithm, a solution to design digital filters with particle swarm algorithm is given. Software simulation is performed based on MATLAB software, and the simulation results are presented.

Keywords

digital filter; particle swarm optimization algorithm; MATLAB.

1. Introduction

Digital filters play an important role in digital signal processing technology and have high value in engineering applications. In the actual engineering application process, in order to make the designed digital filter meet the design standards and guidelines, an approximation method is usually used to design the filter [1]. If large-scale mathematical deduction is carried out, the complexity of the design process is unbearable [2] [3]. Therefore, many scholars use intelligent optimization algorithms to make the various indicators of the digital filter meet the design requirements under certain approximation conditions.

This paper studies the application of particle swarm optimization algorithm in digital filter design, mainly for the design of finite impulse response digital filter. First introduce the design standards and main design methods of the digital filter, and describe the implementation steps of the particle swarm algorithm. Then the optimization model of the digital filter is given, and the realization process of the particle swarm optimization algorithm to solve the digital filter design problem is given. Finally, conclusion marks are presented.

2. Digital Filter Design Theory

Digital filter is a mathematical operation composed of digital multiplier, adder and delay unit. The digital filter changes the frequency spectrum of the input signal to obtain the required output signal. The common digital filter system is a linear time invariant system. The types of digital filters include low pass, high pass, band pass, all pass and band stop. According to the time domain characteristics of the impulse response function, digital filters are divided into finite impulse response (FIR) digital filters and infinite impulse response (IIR) digital filters. Digital filter design standards and guidelines include: mean square error minimization criterion and maximum error minimization criterion [4] [5].

2.1. Mean Square Error Minimization Criterion

This criterion makes the error value close to the minimum. The error range keeps shrinking, enhancing the stability of the system. If $H_d(e^{jw})$ is used to represent the filter frequency response actually obtained, and $E(e^{jw})$ is represented as the actual frequency response error, the error function is expressed as,

ISSN: 2688-8653

$$E(e^{jw}) = H_d(e^{jw}) - H(e^{jw}) \tag{1}$$

the mean square error is,

$$e^2 = \frac{1}{2\pi}\int_{-\pi}^{\pi} | E(e^{jw}) |^2 dw = \frac{1}{2\pi}\int_{-\pi}^{\pi} | H_d(e^{jw}) - H(e^{jw}) |^2 dw \tag{2}$$

Using the mean square error minimization criterion to design the filter is to select the appropriate filter frequency response h(n), so as to minimize the mean square error e^2. The $H_d(e^{jw})$ and $H(e^{jw})$ in the formula (2) are expressed by their impulse responses as,

$$H_d(e^{jw}) = \sum_{n=-\infty}^{\infty} h_d(n)e^{-jwn} \tag{3}$$

$$H(e^{jw}) = \sum_{n=-\infty}^{\infty} h(n)e^{-jwn} \tag{4}$$

This article chooses to design a finite-length digital filter, and the formula (1) can be written as,

$$E\left(e^{j\omega}\right) = H_d\left(e^{j\omega}\right) - E\left(e^{j\omega}\right) = \sum_{n=0}^{N-1}\left[h_d(n) - h(n)\right]e^{-j\omega n} + \sum_{else\ n} h_d(n)e^{-j\omega n} \tag{5}$$

According to the Paseva formula, $\sum_{n=-\infty}^{\infty} | x(n) |^2 = \frac{1}{2\pi}\int_{-\pi}^{\pi} | X\left(e^{jw}\right) |^2 dw$,

$$e^2 = \frac{1}{2\pi}\int_{-\pi}^{\pi} \left|E\left(e^{j\omega}\right)\right|^2 d\omega = \sum_{n=0}^{N-1}\left[h_d(n) - h(n)\right]^2 + \sum_{else\ n}\left|h_d(n)\right|^2 \tag{6}$$

From formula (6), it can be seen that in order to minimize the mean square error, the first term on the right side of the equation must be minimized, that is,

$$| h_d(n) - h(n) | = 0 \text{，} 0 \le n \le N-1 \tag{7}$$

At this time, formula (6) can be rewritten as,

$$e^2 = \min\left[e\right]^2 \tag{8}$$

2.2. Maximum Error Minimization Criterion

This criterion is also called the weighted Chebyshev equivalent ripple approximation criterion. The weighted Chebyshev equivalent ripple approximation is an approximation criterion, and its principle is to ensure that the design error and design accuracy reach an approximation value during the design process. When the frequency response h(n) has center symmetry, the linear phase digital filter can be described as,

$$H(e^{jw}) = e^{-j(\frac{N-1}{2})w} e^{j(\frac{\pi}{2})L} H(\mathrm{w}) \tag{9}$$

Among them, H(w) is the amplitude function.
According to the triangle identity, H(w) is described as,

$$H(w) = Q(w) \cdot P(w) \tag{10}$$

The four linear phase digital filters can be expressed as:

$$H(w) = \sum_{n=0}^{(N-1)} a(n)\cos(wn) = \sum_{n=0}^{(N-1)/2} a^{'}(n)\cos(wn) \tag{11}$$

Where, Q(w)=1.

$$H(w) = \sum_{n=0}^{N/2} b(n)\cos\left[w(n-\frac{1}{2})\right] = \cos(\frac{w}{2})\sum_{n=0}^{N/2} b^{'}(n)\cos(wn) \tag{12}$$

Where, Q(w)=cos(w/2).

$$H(w) = \sum_{n=0}^{(N-1)/2} c(n)\sin(wn) = \sin(w)\sum_{n=0}^{(N-1)/2} c'(n)\cos(wn) \tag{13}$$

$$H(w) = \sum_{n=0}^{N/2} d(n)\sin\left[w(n-\frac{1}{2})\right] = \sin(\frac{w}{2})\sum_{n=0}^{N/2} d^{'}(n)\cos(wn) \tag{14}$$

Where, Q(w)=sin(w/2).
Given that the amplitude function of the filter is H(w), the weighted error function is defined as:

$$E(w) = W(w)[H_d(w) - H(w)] \tag{15}$$

Substituting formula (10) into formula (15), we get,

$$E(w) = W(w)\left[H_d(w) - P(w)Q(w)\right] = W(w)Q(w)\left[\frac{H_d(w)}{Q(w)} - P(w)\right] \tag{16}$$

Make

$$\hat{H}_d(w) = \frac{H_d(w)}{Q(w)}, \quad \hat{W}(w) = W(w)Q(w) \tag{17}$$

The formula (16) can be rewritten as,

$$E(w) = \hat{W}(w)\left[\hat{H}_d(w) - P(w)\right] \tag{18}$$

According to the description of formula (18), the design requirement of linear phase digital filter is to solve different coefficients to minimize the maximum error on different frequency bands, namely,

$$\|E(w)\| = \min\left(\max|E(w)|\right)_{for\ each\ coefficient} \tag{19}$$

3. Particle Swarm Optimization Algorithm

In the particle swarm optimization algorithm, the solution of each optimization problem can be set as a random point on the D-dimensional space, called a "particle". Particles fly in the search space area at a certain speed, and their flying speed can be dynamically adjusted according to the flight experience of themselves and their companions [6] [7]. In each iteration, the particle will update itself through two extreme values, namely the individual extreme value and the global extreme value.

3.1. Mathematical Description

Suppose that in the D-dimensional search space, m particles form a population. The i-th particle is represented as $X_i = (x_{i1}, x_{i2}, ..., x_{iD})$, which represents a point in the D-dimensional solution space, and the flight speed is also a D-dimensional vector $V_i = (v_{i1}, v_{i2}, ..., v_{iD})$. Use fitness function to evaluate each particle, record the optimal position of the i-th particle as $P_i = (p_{i1}, p_{i2}, ..., p_{iD})$, and the optimal position of the entire population as $P_g = (p_{g1}, p_{g2}, ..., p_{gD})$. In the particle swarm algorithm, the position of each particle is updated in the following way,

$$v_{id} = w * v_{id} + c_1 * rand_1 * (p_{id} - x_{id}) + c_2 * rand_2 * (p_{gd} - x_{id}) \tag{20}$$

$$x_{id} = x_{id} + v_{id} \tag{21}$$

Among them, i=1,2,...,m, d=1,2,...,D. The inertia weight w is a scale factor related to speed. The larger the w, the stronger the global search ability of the algorithm, and the smaller the w, the stronger the local search ability of the algorithm. The acceleration constants c_1 and c_2 are non-negative numbers, called learning factors, which respectively determine the influence of individual particle experience and group experience on the particle trajectory. In order to prohibit the particles from moving away from the search space, the speed of the particles will be restricted within the interval $[-v_{d\max}, +v_{d\max}]$.

3.2. Main Parameters

The main parameters of the particle swarm optimization algorithm include: particle population size N, acceleration coefficients c1 and c2, inertia weight w, maximum speed $v_{d\max}$, and boundary condition processing strategies.

(1) population size N

Generally, the number of particles is 20-50. Studies have found that setting the population size to 10 can achieve better results for most research problems; for more complex research problems, the population size can be 100 or 200. The larger the number of particles, the larger the entire spatial range of the algorithm, and the easier to find the global optimal solution; similarly, the smaller the number of particles, the smaller the entire spatial range of the algorithm, and the optimal solution is not easy to find.

(2) inertia weight w

The inertial weight w controls the development ability and exploration ability of the algorithm. When the inertia weight is small, the global optimization ability is weaker, and its local optimization ability is stronger. There are two types of inertial weights, fixed weights and time-varying weights. The fixed inertial weight enables the particles to maintain the same exploration and development capabilities, while the time-varying inertial weight enables the particles to have different development and exploration capabilities.

(3) acceleration coefficients c1 and c2

The acceleration constants c1 and c2 adjust the maximum step length of flight in P_{best} and g_{best} directions respectively. If c1=c2=0, the particles are far away from the boundary or even fly away from the boundary, and it is difficult to find the optimal solution. When c1=0, it belongs to the "social" model, the particle cognitive ability is lacking, and there is group experience. Although the convergence speed is faster, it is easy to fall into the local optimum. When c2=0, it is a "cognitive" model, there is no shared information in society, and the probability of finding the optimal solution is small.

(4) Maximum speed $\text{v}_{\text{d max}}$

The maximum speed limit value of the particle is used to clamp the speed of the particle and control the speed within the range $\text{v}_{\text{d max}}$. The parameter $\text{v}_{\text{d max}}$ is very important. If the value is too large, the particles will fly out of the predetermined search space, making it difficult to find the optimal solution.

4. Design Process and Results

4.1. Optimization Model

This paper adopts the principle of minimizing mean square error to design the digital filter. According to the analysis in Section 2.1, it is necessary to design a set of filter coefficients to minimize the mean square error between the designed filter and the ideal filter. According to formula (7) and formula (8), the optimization model of a one-dimensional linear phase digital filter can be described as,

$$e^2 = \frac{1}{A}\sum_{p=0}^{p-1} a_p \left\{ \left[\sum_{i=0}^{N-1} h_R(i)\cos(iw_p) - H(w_p)\cos\varphi(w_p) \right]^2 + \left[\sum_{i=0}^{N-1} h_R(i)\sin(iw_p) - H(w_p)\sin\varphi(w_p) \right] \right\} \geq 0 \quad (22)$$

$$A = \sum_{P=0}^{P-1} a_p \quad (23)$$

It can be seen that the essence of the optimization problem of a one-dimensional digital filter is to use an optimization algorithm to find a set of coefficients $h_R(i)$ to make the mean square error described by formula (22) as small as possible.

4.2. Implementation Process

The particle swarm algorithm uses fitness to determine the pros and cons of the current position of the particles, so the fitness function F must be selected according to the needs of the actual problem. Here we choose formula (22) as the fitness function, namely,

$$F = e^2 \quad (24)$$

Obviously, the smaller the value of the F function, the smaller the mean square error of the filter coefficient represented by the particle, which can meet the design requirements.

The optimization design flowchart is shown in Figure 1, and the design steps are summarized as follows.

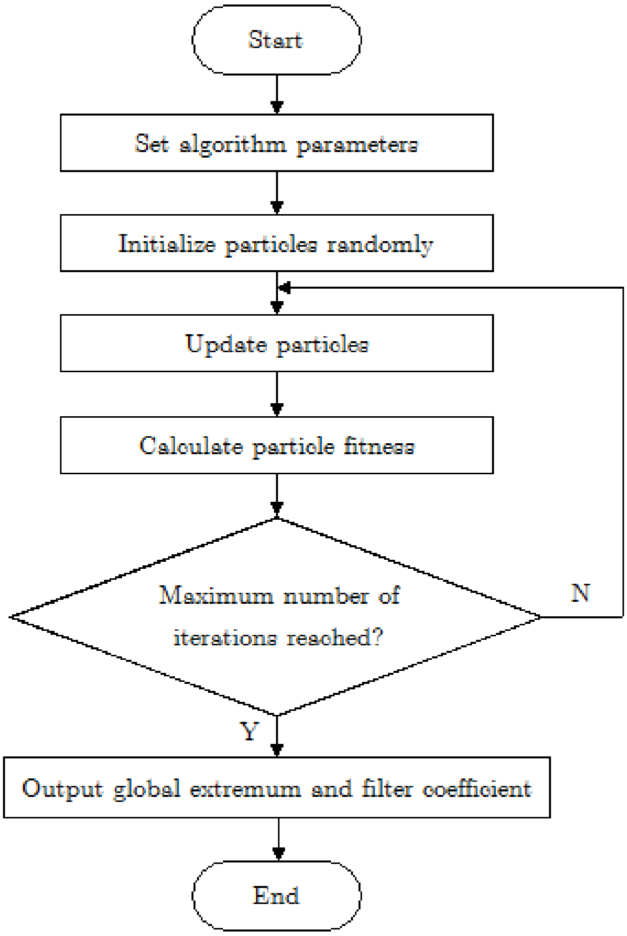

Figure 1: Flow chart of optimized design

(1) Set algorithm parameters, including population size, number of iterations.

(2) Initialize the particles randomly, and calculate the initial individual extreme value and population extreme value according to the fitness function.

(3) According to individual extreme value and population extreme value, update each particle.

(4) Calculate the fitness of the updated particles, record individual extreme values and population extremes.

(5) Judge whether it reaches the maximum number of iterations, if it reaches the maximum number of iterations, stop the iteration and output the population extremum; otherwise, return to step (3).

(6) According to the extremum of the population, obtain the filter coefficients of the current design.

4.3. Experimental Results

Research problem description: Design a one-dimensional low-pass FIR digital filter, the parameters are: N=38, w_p =0.10π, w_s =0.20π, $\varphi(w)$ =20w.

The frequency characteristic of this filter is,

ISSN: 2688-8653

$$H(e^{jw}) = \begin{cases} e^{-j18w}, 0 \le w \le 0.3\pi \\ 0, 0.4 \le w \le \pi \end{cases} \quad (25)$$

Use MATLAB software to carry on the simulation, the result is shown as in Fig. 2. As can be seen from the figure, the passband range of the designed digital filter is 0~0.3π. After the frequency is greater than 0.3π, the response of the filter is below -50dB, indicating that the designed filter can approach the ideal filter. The performance meets the design requirements.

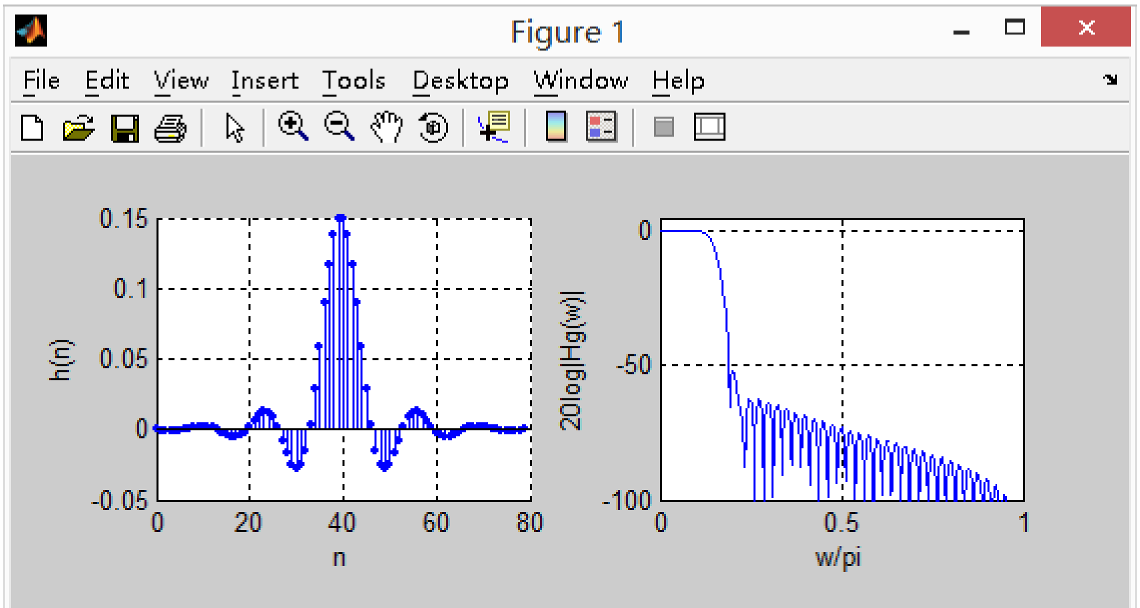

Figure 2: Design filter performance

5. Conclusion

This paper discusses the application of particle swarm optimization algorithm in the design of digital filters, describes the design theory of digital filters in detail, and introduces the mathematical description and main parameter selection of particle swarm optimization algorithm. The optimization model and realization process of the particle swarm optimization design digital filter are described in detail, and MATLAB software is used for simulation design. Experimental results show that the particle swarm algorithm can design a low-pass digital filter with better performance.

Acknowledgements

This paper was supported by the National Natural Science Foundation of China (grant number 61901350); and Science Research Fund of Xi'an Aeronautics University (grant number 2019KY0208).

References

[1] Cai Zhendong, Liu Aiqing, Wang Guanfeng. Application of FPGA-based FIR filter in digital rod control system [J]. Automation Instrumentation, 2020, 41(1): 87-90.

[2] Chen Shaorong, Xu Shun, Shen Jianguo, et al. Design of FIR digital high-pass filter based on Hanning Window [J]. Communication Technology, 2020, 53(5): 1077-1085.

[3] Qian Xiaofeng. Research on digital filter design based on DSP[J]. Electronic World, 2020, (6): 152-153.

[4] Yang Guixin, Yang Yue, Zhang Pengfei. Design of digital bandpass filter based on MATLAB[J]. Digital World, 2020, (4): 284.

[5] Yang Xu, Tian Xiaoying, Sun Hongkai. Digital filter design based on MATLAB[J]. China New Communications, 2020, 22(10): 71.

[6] Hu Xinnan. FIR high-pass digital filter design based on improved chaotic particle swarm optimization algorithm[J]. Computer Science, 2019, 46(z1): 601-604.

[7] Chen Xiaowen. Optimal design of FIR filter based on particle swarm optimization[J]. Journal of Ningde Normal University (Natural Science Edition), 2019, 31(3): 257-262.