Acta Technica 62 (2017), No. 6B, 691–706

FIR digital filter design based on improved hybrid particle swarm optimization algorithm

Chang Cui[1], Qiang Zhao[1]

Abstract. In this paper, the author provides a new inertia weight calculation method and an improved hybrid particle swarm optimization algorithm integrating with simulated annealing, chaos and hybridization aimed at several problems, such as premature convergence existed in particle swarm optimization algorithm. Such method is used in optimization design of finite impulse response (FIR) digital filter, solving the filter coefficient based on MATLAB simulation experiment. According to the experiment results, this paper has analyzed and summarized the effectiveness and superiority of improved method. The results show that such algorithm improves the capacity of getting rid of local extreme point and its convergence rate and precision are superior to other algorithms.

Key words. Digital filter Particle swarm optimization algorithm Chaos Simulated annealing Hybridization.

1. Introduction

The digital filter is an important technology branch of digital signal processing. As several distinctive advantages, such as system stability, strict linear phase and easy hardware realization etc., the FIR digital filter is widely used in biomedicine and geological prospecting, etc. The design of FIR digital filter essentially belongs to multidimensional variable optimization problem. The traditional design methods of FIR digital filter mainly includes the window function method, frequency sampling method and best uniform approximation method. The window function method and frequency sampling method are easily implemented but have difficulty in determining the boundary frequency of its pass band and stop band. The best uniform approximation method based on Parks-McClellan algorithm can obtain better stop

[1]School of Information and Control Engineering, Liaoning Shihua University, Fushun 113001, China

band and pass band capacity, but such algorithm is relatively complex. In the recent years, some scholars are devoted to the design and research of FIR digital filter and propose to design the FIR digital filter by using the optimization algorithm, for example, the intelligent optimization methods, such as particle swarm optimization algorithm and genetic algorithm etc., are integrated with design of FIR digital filter [1, 2].

In 1995, Eberhart and Kennedy put forward a global optimization evolutionary algorithm [3] based on the bird's foraging behavior process simulation, namely, the particle swarm optimization algorithm (called basic PSO, BPSO). Such algorithm is simple in structure and runs fast. In addition, the few required parameter adjustment can be realized by real number encoding so that such algorithm can perform well in solving the non-linear optimization problem, discrete combinatorial optimization problem and engineering application etc. [4] and such algorithm has become an important optimization tool. As many disadvantages exist in BPSO, such as easily premature convergence, easily caught in locally optimal solution and lower searching precision etc. Therefore, many improved algorithms have been proposed subsequently, for example, the particle swarm optimization algorithm with inertia weight and fuzzy self-adaptation particle swarm optimization algorithm respectively proposed in 1998 and 2001 by Shi Y[5, 6], the particle swarm optimization algorithm with Gaussian mutation proposed in 2003 by Natsuki. These algorithms have improved the algorithm performance in different extent. However, the effects are unsatisfactory in solving the multi-dimensional complex problems. Then some people try to integrate the particle swarm optimization algorithm with other thoughts to further improve the algorithm performance, including integrating the particle swarm optimization algorithm with chaos and simulated annealing etc. to form the hybrid particle swarm optimization algorithm [9], for example, chaos PSO, chaos hybrid PSO and simulated annealing hybrid PSP etc. [7, 9] The hybrid particle swarm optimization algorithm has improved the performance of PSO algorithm and performed well in solving the complex problems. According to these optimization thoughts, this paper has integrated PSO with the chaos optimization, simulated annealing and hybridization included in genetic algorithm, proposed a new hybrid particle swarm optimization algorithm by improving the inertia weight and optimization thoughts and verified the effectiveness of new algorithm by comparative analysis of simulation experiment.

2. Basic particle swarm optimization algorithm (BPSO)

The particle swarm optimization algorithm is a random search algorithm based on population and fitness concepts. The position of particle represents for the possible solution of problem. The advantages and disadvantages of particle position can be measured based on the fitness. The algorithm firstly initializes a group of random particles and then finds the optimal solution by multiple iterations. In each process of iteration, the particle can be updated by tracking two extreme values, including the optimal solution found by particle (namely, the individual extreme point (its position is represented by pbest)) and the optimal solution presently found by the whole population (namely, the global extreme point (its position is represented by gbest)). After finding the two extreme values above, the particle will update its speed and location by using the following two formulas:

$$v_{id}^{k+1} = v_{id}^{k} + c_1 rand_1^k (pbest_{id}^k - x_{id}^k) + c_2 rand_2^k (gbest_d^k - x_{id}^k)\,, \tag{1}$$

$$x_{id}^{k+1} = x_{id}^{k} + v_{id}^{k+1}\,. \tag{2}$$

Where the information of particle i can be represented by D dimension vector, its position can be represented as ***Xi***=(xi1,xi2,..., xiD)T, its speed can be represented as ***V***i=(vi1,vi2,..., viD)T. c1 and c2 are learning factors, respectively adjusting the individual optimal particle and global optimal particle. The suitable c1 and c2 values are beneficial to accelerate the convergence rate and not easily trapping into the local optimum. $rand_1^k$ and $rand_2^k$ are random numbers in the range of [0,1].

3. Hybrid particle swarm optimization algorithm integrating with chaos, simulated annealing and hybridization

3.1. Chaos optimization

Chaos is a non-linear phenomenon universally existed in the nature. The non-deterministic random motion state obtained by determining the equation is called chaos and the variable in chaos state in equation is called chaotic variable. Although the chaos is seemingly random, it has an exquisite internal structure and different characteristics, such as randomness, ergodicity and sensitivity to the initial value and is widely used in local optimization. In 2004, Gao Ying etc. [10] integrated the particle swarm optimization algorithm with chaos and conducted search by using chaos local search mechanism, which avoids the premature convergence problem caused due to particle stagnation and improves the algorithm convergence rate and global optimization capacity.

At present, there are lot of chaos models and Logistic mapping is used in most of literatures. However, the sequences generated by Logistic mapping are not uniform and will waste the calculation time. Therefore, another model-logistic self-mapping function is used in this paper. Such mapping is relatively simple and easy for computer calculation. The chaos variables generated are more uniform and can better traverse the feasible region. The logistic self-mapping function model is shown below:

$$y_{n+1} = 1 - 2 * y_n^2, \quad -1 < y_n < 1\,. \tag{3}$$

Where $n = 0, 1, 2, \ldots$ represents the number of iterations, $y_n \in (-1, 1)$ represents nth chaos variable of particle X_i. In actual application, chaos will happen if only the initial value to iteration is not zero. Fig. 1 shows the chaos ergodicity when $y_0 = 0.1475$ and the number of iteration is 2000. In theory, after long time iteration, chaos will traverse all values in specified section. Fig. 2 is the space distribution diagram of two chaos sequences when giving $y_{10} = 0.1475$ and $y_{20} = 0.14750001$. As shown in Fig. 2, as time goes on, any similar initial condition shows independent time evolution, which indicates its sensibility to the initial value.

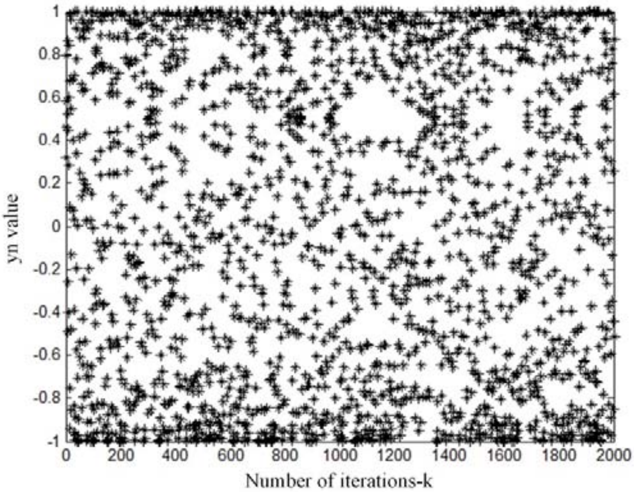

Fig. 1. Chaos ergodicity of logistic self-mapping

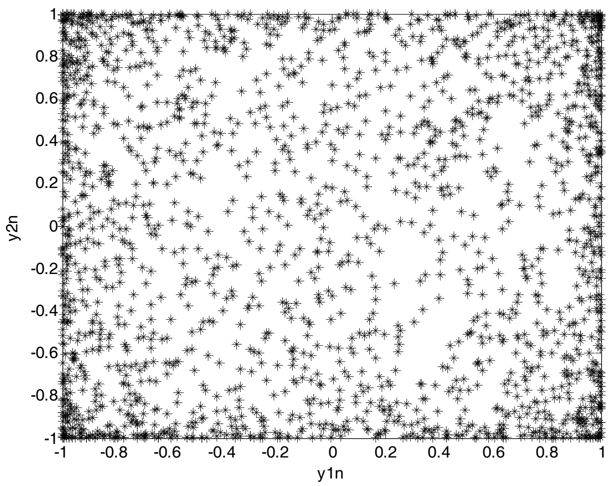

Fig. 2. Sensibility of chaos to initial value

3.2. Simulated annealing

The simulated annealing algorithm is a random combination optimization algorithm developed in the 1980s. Such algorithm simulates the thermodynamic process of hot metal temperature reduction, has probability jumping capacity, is able to avoiding trapping into the local minimum in search process, has asymptotic convergence and parallelism and has been widely used in combination optimization problem.

The main idea of simulated annealing algorithm is firstly using an initial temperature T_0 and then using the condition judgment criterion.

$$\min\{1, \exp[-(fitness(x') - fitness(x))/T_k]\} > random\,. \tag{4}$$

Judge whether it accepts x', in other words, if inequation is satisfied, use child particle x' to replace parent particle x; where fitness is the fitness function, random is the random number in [0, 1]. Then conduct the annealing operation by equation $T_{k+1} = CT_k$ (where C represents the annealing constant between 0 and 1), repeat above processes until the completion of annealing process after the condition is satisfied. As the simulated annealing algorithm has many disadvantages, such as heavy computation, great impacts of initial value on algorithm performance and parameter sensibility etc. Such algorithm is less used in the design of FIR digital filter. However, such algorithm is firstly used in the literature [11] for the design of FIR digital filter and verification of the effectiveness of such algorithm.

It is important to set up suitable initial temperature in the simulated annealing algorithm. If the initial temperature is higher, the global searching capacity will be improved but plenty of time will be spent; if the initial temperature is lower, the search time will be reduced but global optimum might not be found. Therefore, a new method of setting up the initial temperature has been proposed in this paper on the condition of taking into account of global optimum and search time, that is:

$$T_0 = fitness(\text{gbest})/\ln 5\,.$$

3.3. Particle swarm optimization algorithm based on hybridization

The premature convergence will occur on BPSO algorithm during evolution. In order to solve this problem and improve the algorithm precision, Angeline P [12] had integrated the particle swarm optimization algorithm with hybridization concept of genetic algorithm and proposed the hybrid particle swarm optimization algorithm.

The main idea of hybrid particle swarm optimization algorithm is selecting specified number of particles to put into hybridization pool based on hybridization probability, making hybridization between two random particles in pool and generating same number of child particle to replace parent particle. The particle trapped into local optimization will jump out by hybridization and find the optimal solution. The position of child particle can be known by arithmetic weighting of parent particle position, that is:

$$child_1(x) = p * parent_1(x) + (1-p) * parent_2(x)\,. \tag{5}$$

$$child_2(x) = p * parent_2(x) + (1-p) * parent_1(x)\,. \tag{6}$$

Where p is the D-dimension random number uniformly distributed in [0, 1].

The speed of child particle is calculated based on the following formula:

$$child_1(v) = \frac{parent_1(v) + parent_2(x)}{|parent_1(v) + parent_2(x)|} * |parent_1(v)| \,, \tag{7}$$

$$child_2(v) = \frac{parent_1(v) + parent_2(x)}{|parent_1(v) + parent_2(x)|} * |parent_2(v)| \,. \tag{8}$$

The child particle cannot replace the parent particle through the hybridization operation by the particle swarm, which increases the population diversity. But the superior parent particle may be missed because the hybrid PSO algorithm makes the child particle completely replace the parent particle. Therefore, the simulated annealing condition judgment method is increased in this paper on this basis to judge whether the parent particles should be replaced, which ensures the population diversity, avoids occurrence of missing the optimal position and improves the algorithm convergence precision.

3.4. Improved hybrid particle swarm optimization algorithm

The three optimization algorithms mentioned above have been incorporated in PSO algorithm and the pair-wise combined hybrid particle swarm optimization algorithm is proposed. This paper incorporates the three optimization thoughts into PSO algorithm in the same time, proposes an improved hybrid PSO algorithm by making full use of respective advantages and verifies the performance and superiority of new algorithm.

The main ideas of new algorithm are described as below: after particle hybridization, judge whether the parent particle should be updated by using the comparison strategy in simulated annealing thought, keep the historical optimal solution of current state; then conduct chaos local search for the new sub-population generated after hybridization by using chaos optimization; make the global optimal solution searched randomly replace parent particle; finally use the simulated annealing operation to conduct cyclic search until the requirements are satisfied. In this way, the population diversity can be kept, the global optimal solution searched in specified time is more superior and thus the algorithm precision and convergence rate will be improved. This paper also incorporates a new inertia weight calculation method into new algorithm and verifies the feasibility and superiority of such algorithm in optimization design of FIR digital filter.

Now we introduce the important parameters in particle swarm optimization algorithm- inertia weight ω. The inertial weight is firstly incorporated into the particle swarm optimization algorithm in the literature [5] and become the most important adjustable parameter in algorithm. The location updating equation is changed as below:

$$v_{id}^{k+1} = \omega v_{id}^{k} + c_1 rand_1^k (pbest_{id}^k - x_{id}^k) + c_2 rand_2^k (gbest_d^k - x_{id}^k) \,. \tag{9}$$

The frequently-used inertia weight calculation methods include the constant method, random weight method, linear decreasing method, exponential function decreasing method and concave function decreasing method etc. According to the formula (9), it can be discovered that the larger inertia weight can increase the global searching capacity in initial stage and the smaller inertia weight can increase the global searching capacity in later stage. The later three weight calculation methods have been simply analyzed in this paper to effectively balance the global and local searching capacity and improve the global optimal solution solving capacity, as shown below:

Linear decreasing: $\omega = \omega_{min} - (\omega_{max}^{-}\omega_{min}) * t/T$

Function decreasing: $\omega = \omega_{min} + (\omega_{max}^{-}\omega_{min}) exp[-20(t/T)^6]$

Concave function decreasing: $\omega = \omega_{min} * (\omega_{\max}/\omega_{min})^{\frac{1}{1+10*t/T}}$

Where ω_{max} and ω_{min} respectively represent the maximum inertia weight and inertia weight minimum, generally assigning 0.9 and 0.4; t represents the current number of iterations and T represents the maximum number of iterations. The weight in exponential function form is relatively larger in initial stage and its decreasing speed is too fast in later stage. However, the decreasing speed of linear decreasing weight is moderate in later stage. Therefore, the new inertia weight calculation method proposed in this paper is introduced as below: selecting the exponential weight before intersection point A and linear weight after intersection point A.

3.5. Algorithm performance test

The two Benchmark functions used for algorithm performance test are used to verify the performance of improved PSO algorithm, where Rastrigrin is unimodal function Griewank is multimodal function.

Rastrigrin function: $f_1(x) = \sum\limits_{i=1}^{n} \left[x_i^2 - 10\cos(2\pi x_i) + 10\right]$

Griewank function: $f_2(x) = \frac{1}{4000} \sum\limits_{i=1}^{n} x_i^2 - \prod\limits_{i=1}^{n} \cos(\frac{x_i}{\sqrt{i}}) + 1$

To compare the algorithm performance, the chaos PSO algorithm (CPSO), hybrid PSO algorithm (HPSO), chaos hybrid PSO method (CHPSO) and improved PSO algorithm (MPSO) are used to solve two functions; in MPSO algorithm, use three inertia weight methods to solve the function; the number of iterations is 1000; take the average value as the final result after 30 times of operation as shown in Table 1 to Table 3.

According to Table 1, it can be known that the solution accuracy and convergence rate of improved hybrid PSO algorithm are superior to other three hybrid PSO algorithm and the improved hybrid PSO algorithm has stable performance although such algorithm is relatively complex. According to Table 2 and Table 3, it can be known that relative ideal results are obtained on average convergence rate and average number of iterations. Therefore, such algorithm is used for optimization design of FIR digital filter.

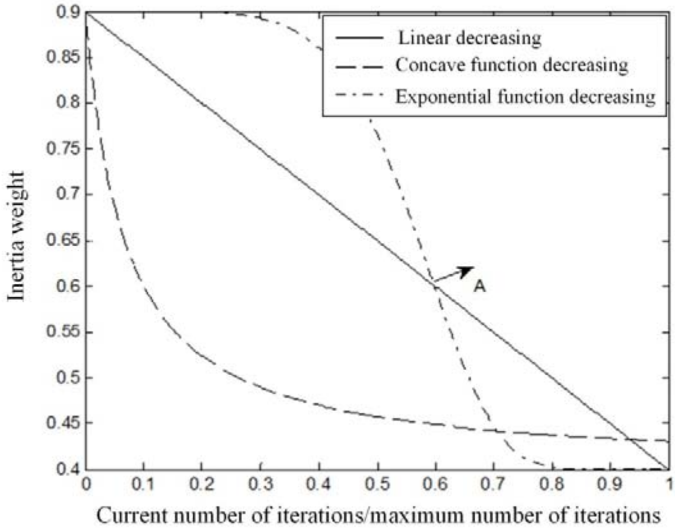

Fig. 3. Comparison and selection of inertia weight

Table 1. Performance comparison for each algorithm

Algorithm	Function	Optimal solution	Average convergence rate
HPSO	Rastrigrin	0.042142351	67%
	Griewank	0.370986521	83%
CPSO	Rastrigrin	0.061984378	84%
	Griewank	0.297875321	79%
CHPSO	Rastrigrin	0.020721876	91%
	Griewank	0.052317807	84%
MPSO	Rastrigrin	0.000275431	100%
	Griewank	0.001534178	81%

Table 2. Computational results of rastrigrin function solved by using MPSO

Inertia weight method	Number of dimensions	size	Average convergence rate	Average convergence generations
Linear decreasing	10	30	27/30	228
	30	80	30/30	219
Exponential decreasing	10	30	22/30	114
	30	80	30/30	195
Mixed decreasing	10	30	30/30	55
	30	80	30/30	87

Table 3. Computational results of griewank function solved by using MPSO

Inertia weight method	Number of dimensions	Population size	Average convergence rate	Average convergence generations
Linear decreasing	10	30	14/30	664
	30	80	5/30	922
Exponential decreasing	10	30	18/30	666
	30	80	3/30	872
Mixed decreasing	10	30	22/30	441
	30	80	8/30	622

4. FIR digital filter and its design

4.1. Linear-phase FIR digital filter

The unit impulse response $h(n)$ of FIR digital filter is finite ($0 \leq n \leq N-1$, N is the filter order) and its system function is as below:

$$H(z) = \sum_{n=0}^{N-1} h(n) z^{-n} . \qquad (10)$$

Taking z=ejw , the frequency response function of filter is as below:

$$H(e^{j\omega}) = \sum_{n=0}^{N-1} h(n) e^{-j\omega n} . \qquad (11)$$

If h(n) is the real number and meets any of the following conditions:

$$h(n) = h(N-1-n) . \qquad (12)$$

$$h(n) = -h(N-1-n) . \qquad (13)$$

Then the filter has strict and accurate linear phase. As the even number or odd number can be taken for N, $h(n)$ includes four types, corresponding to four types of linear-phase FIR digital filter.

The design of FIR digital filter is system function solving, namely, filter coefficient $h(n)$ solving, making the filter frequency response or amplitude-frequency characteristic actually designed optimally approximate the ideal filter. As the main advantage of FIR digital filter is the strict linear phase, this paper mainly discusses the optimization design problem existed in linear-phase FIR filter.

4.2. Optimization design criteria of FIR digital filter

The optimization design criteria must be determined before filter optimization design. There are two optimization criteria for FIR digital filter design, including mean square error minimization criteria and maximum error minimization criteria. The mean square error minimization criterion is adopted in this paper.

The mean square error minimization criterion is described as below. Set the frequency response of ideal filter as $H_d(e^{j\omega})$ and the frequency response of filter actually designed as$H(e^{j\omega})$, the error sum of squares of two frequency responses in the Mth discrete sampling point ω_i(i=1,2,...,M) is defined as below:

$$\begin{aligned} E &= \sum_{i=1}^{M} \left[\left|H(\mathrm{e}^{j\omega_i})\right| - \left|H_d(\mathrm{e}^{j\omega_i})\right|\right]^2 \\ &= \sum_{i=1}^{M} \left[\left|\sum_{n=0}^{N-1} h(n)e^{-j\omega_i n}\right| - \left|H_d(\mathrm{e}^{j\omega_i})\right|\right]^2. \end{aligned} \tag{14}$$

It is defined as filter error function. Obviously E is the multivariate nonlinear function of filter coefficient $h(n)$, which belongs to the combinational optimization problem. Such problem can be solved by using PSO algorithm. However, the FIR filter response design is to seek for the optimal filter coefficient $h(n)$ and make the minimum error function E.

5. FIR digital filter design by using the new algorithm

5.1. Fitness function selection

PSO algorithm determines the advantages and disadvantages of current position of particle based on fitness (namely, the objective function value). Therefore, the suitable fitness function Fitness should be selected first based on the requirements of actual problems. According to the analysis in above section, the formula (14) will be used as fitness function of FIR digital filter design, namely:

$$\text{Fitness}= \sum_{i=1}^{M} \left[\left|\sum_{n=0}^{N-1} h(n)e^{-j\omega_i n}\right| - \left|H_d(\mathrm{e}^{j\omega_i})\right|\right]^2. \tag{15}$$

The smaller the Fitness, the smaller the mean square error of filter coefficient of particle and the better the filter coefficient obtained. After end of algorithm, the parameter value of particle with minimum fitness value during entire operation is the optimal solution, namely, the filter coefficient required to be solved.

5.2. Detailed process of FIR digital filter design

The main steps of chaos local search algorithm involved in chaos optimization thoughts above are introduced as below:

1) Given k=0, map decision variable x_j^k as chaos variable s_j^k between 0 and 1 based on $s_j^k = (x_j^k - x_{\min,j})/(x_j^k - x_{\max,j})$, where $x_{\max,j}$ and $x_{\min,j}$ respectively represent for upper and lower bound of jth dimension of variable;

2) Calculate the iterated chaos variable $s_j^{k+1} = 1 - 2 * s_j^{k2}$;

3) Convert the chaos variable s_j^{k+1} into decision variable based on $x_j^{k+1} = x_{\min,j} + s_j^{k+1}(x_{\max,j} - x_{\min,j})$;

4) Evaluate the new solution based on the decision variable; if the new solution is superior to initial solution or the maximum number of iterations is reached in chaos search, the new solution can be used as algorithm search result; if not, set $k = k+1$ and turn to the step 2.

The main steps of linear-phase FIR digital filter design by using the new algorithm include the following:

1) Given the technical index of linear-phase FIR digital filter;

2) Set up the parameters of new algorithm, including the population size, initial annealing temperature, hybridization probability and maximum number of iterations etc.;

3) Randomly initialize the speed and position of each particle in parameter range and calculate the fitness value;

4) Update the particle speed and position and calculate the fitness value of each particle according to formula (9) and (2);

5) Judge whether pbest of particle and gbest of whole population should be updated by comparing the fitness value of each particle;

6) Select specified number of particles to put into hybridization pool based on hybridization probability Pc, make hybridization between two random particles in pool and generate same number of child particle, judge whether the child particle should be deemed as new individuals according to simulated annealing condition criteria (4), calculate the speed and position of new individuals based on formula (5) and (8) and keep pbest and gbest unchanged;

7) Carry out chaos local search for the new sub-population by using the chaos local search algorithm and update its pbest and gbest;

8) Carry out annealing operation based on equation $T_{k+1} = CT_k$;

9) Judge whether the precision requirements are satisfied or the maximum number of iterations is reached; if satisfied, end the algorithm; if not, return to step 4 for search;

10) Output gbest, obtain filter coefficient $a(n)$ and get all coefficients based on the relation between $h(n)$ and $a(n)$ and symmetry of $h(n)$.

6. Experiment simulation and analysis

In order to verify the effectiveness of algorithm proposed in this paper, the simulation experiment is conducted on computer by using MATLAB for design of linear-phase FIR digital low-pass filter and band-pass filter. To compare the algorithm performance, CPSO and CHPSO algorithms are used for filter optimization design. Each algorithm runs for 30 times and the average result is taken as final result. The parameters of improved hybrid particle swarm optimization algorithm are set as below: the population size is set as 40, the number of dimensions of parameters is (N+1)/2, the size of hybridization pool is 0.2 (Sp=0.2), the hybridization probability is 0.9 (Pc=0.9), the maximum step number of chaos search is 20, the annealing constant is 0.8, the value range of particle position is [-1, 1], the speed value range is [-2, 2] and the maximum number of iterations is 1000.

Example 1: design a digital low-pass filter whose step number is 21 and minimum attenuation of stop band is 20dB; the technical index is as below:

$$H_d(e^{j\omega}) = \begin{cases} 1, & \omega \in [0, 0.20\pi] \\ 0, & \omega \in [0.38\pi, \pi] \end{cases}$$

Example 2: design a digital band-pass filter whose step number is 32 and minimum attenuation of stop band is 30dB; the technical index is as below:

$$H_d(e^{j\omega}) = \begin{cases} 1, & \omega \in [0, 0.20\pi] \bigcup [0.80\pi, \pi]\,, \\ 0, & \omega \in [0.35\pi, 0.65\pi]\,. \end{cases}$$

The simulation results of example 1 are shown in Fig. 4 and Fig. 5; the simulation results of example 2 are shown in Fig. 6 and Fig. 7. According to Fig. 4 and Fig. 6, it can be known that the stop band characteristics of FIR digital filter designed by using improved particle swarm optimization algorithm meet the technical requirements in low pass and band pass and obviously superior to the digital filter designed by HPSO and CPSO; the stop band attenuation increases 30dB than the predicted index. According to the amplitude-frequency response curves in Fig. 5 and Fig. 7, the filter designed by using the improved particle swarm optimization algorithm has faster convergence rate and higher convergence precision; there are no ripples on stop band and the stop band is relatively stable. Such filter is more close to the ideal filter.

7. Conclusion

This paper incorporates the three thoughts (simulated annealing, chaos and hybridization) into particle swarm optimization algorithm, proposes an improved hybrid PSO algorithm by making full use of respective advantages. Although such algorithm is relatively complex, it keeps the population diversity, enhances the global optimal solution searching capacity to a great extent and improves the convergence

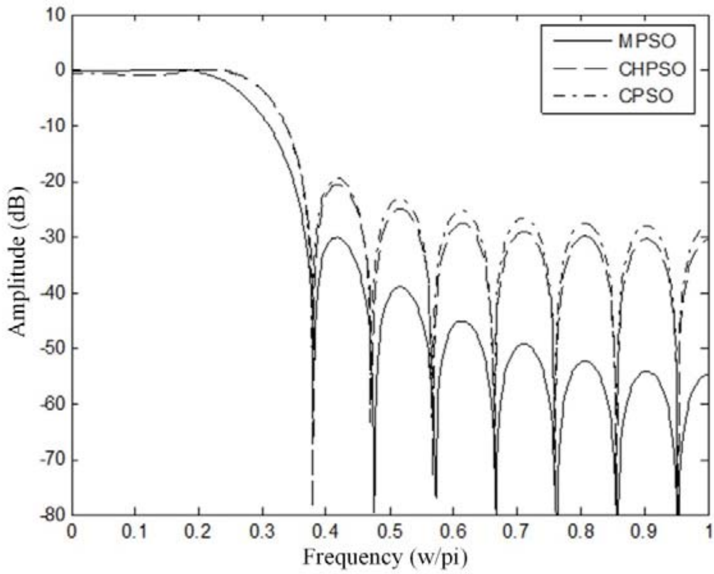

Fig. 4. Logarithm amplitude-frequency response of FIR low-pass digital filter

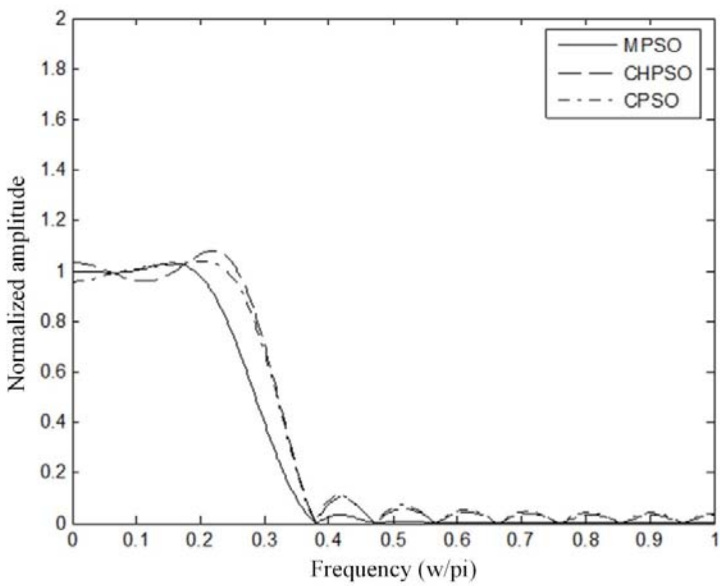

Fig. 5. Amplitude-frequency response of FIR Low-pass digital filter

rate and precision; its optimization effects are better than the same of other hybrid particle swarm optimization algorithm. In addition, the new inertia weight calculation method proposed in this paper provides new thoughts for further research. Such method is used in simulation experiment of FIR digital filter to verify its superiority.

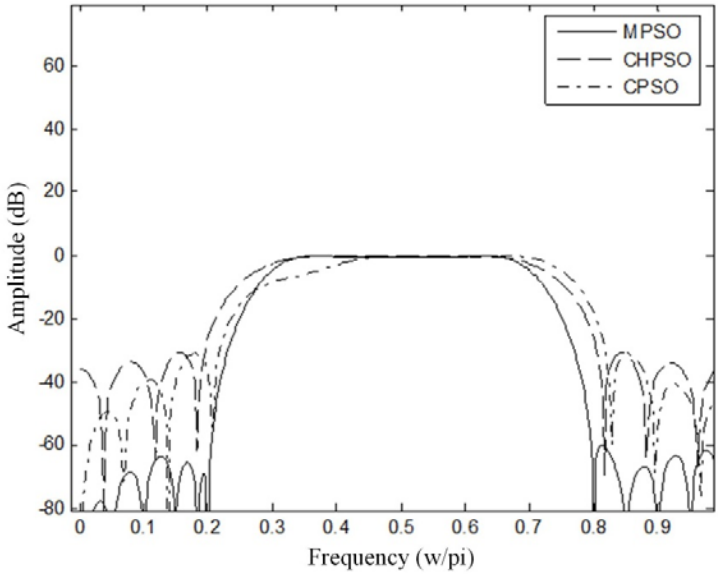

Fig. 6. Logarithm amplitude-frequency response of FIR band-pass digital filter

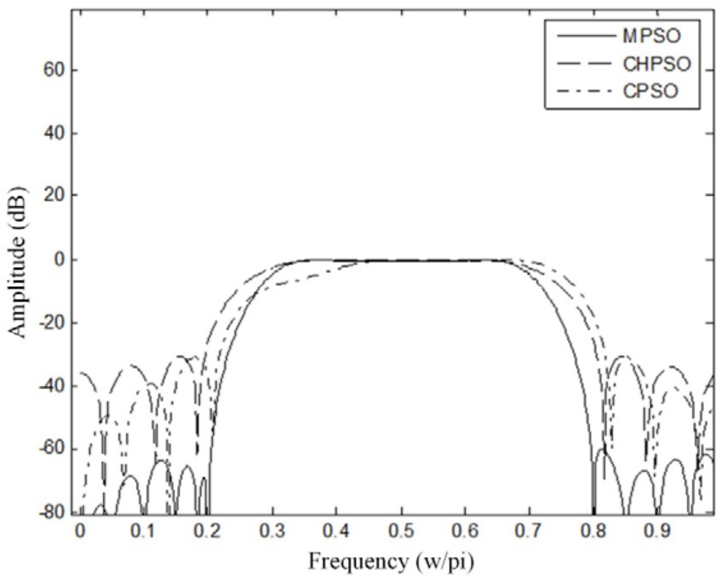

Fig. 7. Amplitude-frequency response of FIR band-pass digital filter

Although such algorithm is superior to HPSO, CPSO and CHPSO, there are many problems remained to be researched and solved in terms of integration problem, such as the deep impacts of relevant parameters (including maximum number of iterations of chaos, simulated annealing temperature parameter, hybrid particle number and population size etc.) on algorithm performance etc. The subsequent research will continue in this direction.

Acknowledgement

Scientific research project of Liaoning Provincial Education Department (L2016025).

References

[1] Shao P, Wu Z, Zhou X, et al. (2015) *FIR digital filter design using improved particle swarm optimization based on refraction principle*[J]. Soft Computing, 21(10):2631-2642.

[2] Vasundhara, Mandal D, Kar R, et al. (2014) *Digital FIR filter design using fitness based hybrid adaptive differential evolution with particle swarm optimization*[J]. Natural Computing, 13(1):55-64.

[3] Huang W, Zhou L, Qian J, et al. (2005) *FIR Frequency Sampling Filters Design Based on Adaptive Particle Swarm Optimization Algorithm*[M]// Advances in Natural Computation. Springer Berlin Heidelberg, 2005:289-298.

[4] Mandal S, Ghoshal S P, Kar R, et al. (2012) *Design of optimal linear phase FIR high pass filter using craziness based particle swarm optimization technique*[J]. Journal of King Saud University - Computer and Information Sciences, 24(1):83-92.

[5] Fang W, Sun J, Xu W, et al. (2006) *FIR Digital Filters Design Based on Quantum-behaved Particle Swarm Optimization*[C]// International Conference on Innovative Computing, Information and Control. DBLP, 2006:615-619.

[6] Saha S K, Kar R, Mandal D, et al. (2015) *Optimal IIR filter design using Gravitational Search Algorithm with Wavelet Mutation*[J]. Journal of King Saud University - Computer and Information Sciences, 27(1):25-39.

[7] Hashemi S A, Nowrouzian B (2012) *A Novel Discrete Particle Swarm Optimization for FRM FIR Digital Filters*[J]. Journal of Computers, 7(6).

[8] Tawhid M A, Ali A F (2016) *Simplex particle swarm optimization with arithmetical crossover for solving global optimization problems*[J]. Opsearch, 53:1-36.

[9] Rafi S M, Kumar A, Singh G K (2013) *An improved particle swarm optimization method for multirate filter bank design*[J]. Journal of the Franklin Institute, 350(350):757–769.

[10] Mondal S, Ghoshal S P, Kar R, et al. (2012) *Differential Evolution with Wavelet Mutation in Digital Finite Impulse Response Filter Design*[J]. Journal of Optimization Theory and Applications, 155(1):315-324.

[11] Saha S K, Kar R, Mandal D, et al. (2014) *Optimal linear phase FIR filter design using particle swarm optimization with constriction factor and inertia weight approach with wavelet mutation*[J]. International Journal of Hybrid Intelligent Systems, 11(2):81-96.

[12] De B P, Kar R, Mandal D, et al. (2015) *Optimal high speed CMOS inverter design using craziness based Particle Swarm Optimization Algorithm*[J]. Open Engineering, 5(1):256-273.

[13] Rajesh Kumar, Anupam Kumar (2010) *Design Of Two-Dimensional Infinite Impulse Response Recursive Filters Using Hybrid Multiagent Particle Swarm Optimization*[J]. Applied Artificial Intelligence, 24(4):295-312.

Received May 7, 2017