

Research Article

Phase Response Design of Recursive All-Pass Digital Filters Using a Modified PSO Algorithm

Wei-Der Chang

Department of Computer and Communication, Shu-Te University, Kaohsiung 824, Taiwan

Correspondence should be addressed to Wei-Der Chang; wdchang@stu.edu.tw

Received 5 October 2014; Revised 22 December 2014; Accepted 1 January 2015

Academic Editor: Rahib H. Abiyev

Copyright © 2015 Wei-Der Chang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper develops a new design scheme for the phase response of an all-pass recursive digital filter. A variant of particle swarm optimization (PSO) algorithm will be utilized for solving this kind of filter design problem. It is here called the modified PSO (MPSO) algorithm in which another adjusting factor is more introduced in the velocity updating formula of the algorithm in order to improve the searching ability. In the proposed method, all of the designed filter coefficients are firstly collected to be a parameter vector and this vector is regarded as a particle of the algorithm. The MPSO with a modified velocity formula will force all particles into moving toward the optimal or near optimal solution by minimizing some defined objective function of the optimization problem. To show the effectiveness of the proposed method, two different kinds of linear phase response design examples are illustrated and the general PSO algorithm is compared as well. The obtained results show that the MPSO is superior to the general PSO for the phase response design of digital recursive all-pass filter.

1. Introduction

An all-pass filter means that its magnitude response is exactly equal to some constant value at all frequencies and independent of frequencies. The function of the all-pass filter is mainly to offer a phase modification without changing the magnitude on a given filter. It is rather useful in the theory of minimum-phase systems, in transforming frequency-selective low-pass filters into other frequency-selective forms, and in obtaining variable-cutoff frequency-selective filters [1]. Due to these advantages, the all-pass filter has been applied in many signal processing applications, including the group-delay equalization, complementary filter banks, multirate filtering, and other fields [2–8]. A large number of methods for designing all-pass filter have been developed in recent years. In [2], for example, the author proposed a new design method for an all-pass filter where it has a least squares or an equiripple phase-error response. It is based on formulating a weighted error between the desired and the actual phase responses in a quadratic form. Filter coefficients can be solved by using a Toeplitz-plus-Hankel matrix. In [3], an IIR all-pass filter with equiripple phase response was designed based

on the eigenvalue problem and this design problem can be formulated as the representation of an eigenvalue problem via the Remez exchange algorithm. A Hopfield neural network was combined to the design of IIR all-pass digital filters [5]. In the case, filter coefficients can be evaluated by Hopfield neural networks in a parallelism manner in accordance with the error function that is formulated as a Lyapunov energy function. In addition, the authors developed a digital linear phase notch filter design scheme based on IIR all-pass filter. The designed filter can be realized by parallel connection of two IIR all-pass filters with approximately linear phase. Design algorithms exhibit fast convergence and easy initial values determination [7].

Unlike the above-mentioned design schemes, this paper attempts to utilize a modified particle optimization (MPSO) algorithm to solve the digital recursive all-pass filter design problem. This developed algorithm is a variant of the general PSO but it has a better searching capacity in solving optimized problems. The detailed description for such MPSO algorithm will be addressed later. The remainder of this paper is summarized as follows. Section 2 gives a brief description for the recursive all-pass digital filter. In Section 3, a modified

PSO algorithm is introduced in detail and the MPSO-based design steps for all-pass digital filter are also given. Section 4 will provide two different kinds of examples to confirm the applicability of the proposed method and some comparisons with the general PSO are further made. Finally, a conclusion about the proposed method is simply described in Section 5.

2. Recursive All-Pass Digital Filter

Let us consider a recursive all-pass digital filter whose transfer function is expressed by

$$H(z) = \frac{a_N + a_{N-1}z^{-1} + \dots + a_0z^{-N}}{a_0 + a_1z^{-1} + \dots + a_Nz^{-N}} = \frac{z^{-N}D(z^{-1})}{D(z)}, \quad (1)$$

where N represents the order of the filter, a_0 is always set to be 1, a_1, a_2, \dots, a_N are real coefficients, and $D(z) = \sum_{i=0}^N a_i z^{-i}$. Let $z = e^{j\Omega}$, where Ω denotes the digital frequency, substituting it into (1) to derive the frequency response. The following magnitude response can be easily obtained:

$$\begin{aligned} |H(\Omega)| &= \frac{|z^{-N}D(z^{-1})|}{|D(z)|} = \frac{|D(z^{-1})|}{|D(z)|} \\ &= \frac{|1 + a_1 e^{j\Omega} + \dots + a_N e^{jN\Omega}|}{|1 + a_1 e^{-j\Omega} + \dots + a_N e^{-jN\Omega}|} = 1. \end{aligned} \quad (2)$$

It is seen from (2) that the magnitude response is equal to one at all frequencies; that is, it is independent of the filter coefficients. Furthermore, its phase response is derived by

$$\begin{aligned} \theta(\Omega) &= -N\Omega + 2 \arctan\left(\frac{\sum_{i=1}^N a_i \sin(i\Omega)}{\sum_{i=1}^N a_i \cos(i\Omega)}\right) \\ &= -N\Omega + 2 \arctan\left(\frac{\mathbf{a}\mathbf{s}^T(\Omega)}{1 + \mathbf{a}\mathbf{c}^T(\Omega)}\right), \end{aligned} \quad (3)$$

where $\mathbf{s}(\Omega) \equiv [\sin(\Omega), \sin(2\Omega), \dots, \sin(N\Omega)]$, $\mathbf{c}(\Omega) \equiv [\cos(\Omega), \cos(2\Omega), \dots, \cos(N\Omega)]$, and

$$\mathbf{a} \equiv [a_1, a_2, \dots, a_N] \quad (4)$$

is a parameter vector consisting of all filter coefficients. This vector fully dominates the phase response behavior of the digital filter. In this paper, we want to design the parameter vector \mathbf{a} such that the phase response achieves certain design specification. Moreover, this vector \mathbf{a} is called the particle or individual of the PSO algorithm and many such particles then form a population. Some adjusting mechanisms are utilized on the full population. Moreover, it can be easily seen from (3) that a highly complicated nonlinear function $\arctan(\cdot)$ is involved and it is difficult to solve. Thus, (3) always needs to be modified as another form for the phase response design [3–5]. However, the proposed method in this paper can directly use (3) for the phase response design of recursive all-pass filter.

3. Modified Particle Swarm Optimization (MPSO) Algorithm

Kennedy and Eberhart initially proposed the PSO algorithm in 1995 and recently it became one of the popular and efficient

optimization algorithms [9]. Like most swarm intelligence algorithms, PSO is also a population-based search algorithm. It simulates the social behavior of organisms, such as fish schooling and bird flocking. Each fish or bird, viewed as a particle or an individual, represents a candidate solution to the optimized problem. By the velocity and position updating formulas, each particle moves through the search space toward the global solution. Based on the PSO algorithm, various engineering optimization applications have been successively developed and explored in recent years, such as power system stabilizer design [10], PID controller design [11, 12], FPGA implementations [13, 14], Volterra filter modeling [15], QRD-based multi-relay system design [16], automatic clustering [17], multifault classification [18], and aeroengine nonlinear programming model [19]. Besides, the authors developed a novel PSO algorithm in which the inertia weight is modified to enhance its search capability [20]. The proposed method has successfully been applied in the high pass FIR digital filter design. Another design method for the low pass FIR digital filter with linear phase properties was also developed [21]. A new definition for the velocity vector and swarm updating of the PSO algorithm was proposed.

At the beginning, PSO algorithm requires an objective function to judge the performance of the particle and also to guide the search direction of the algorithm. To solve the phase response design problem for the recursive all-pass digital filter, the objective function (OF) is defined by

$$\text{OF} = \int_{\Omega_{\min}}^{\Omega_{\max}} |\theta_d(\Omega) - \theta(\Omega)| d\Omega, \quad (5)$$

where θ_d is the desired phase response given by the designer, θ is the actual phase response of the all-pass digital filter as described by (3), and Ω_{\min} and Ω_{\max} are the integral lower and upper bounds, respectively. The algorithm is utilized to minimize this objective function OF to achieve the optimal phase response design. Each particle is changed according to the following velocity formula of (6) and position formula of (7) for original PSO algorithm:

$$\begin{aligned} v_{ij}(n+1) &= wv_{ij}(n) + c_1 r_1 (p_{ij}(n) - a_{ij}(n)) \\ &\quad + c_2 r_2 (g_j(n) - a_{ij}(n)), \end{aligned} \quad (6)$$

$$a_{ij}(n+1) = a_{ij}(n) + v_{ij}(n+1), \quad (7)$$

where n denotes the n th iteration of the algorithm, v_{ij} , a_{ij} , and p_{ij} represent the velocity, position, and individual best position for the i th particle with respect to the j th dimension, respectively, g_j represents the global best position with respect to the j th dimension among the population, w is called the inertia weight, c_1 and c_2 are two positive acceleration coefficients that pull each particle toward the individual best and the global best positions, respectively, and r_1 and r_2 are two uniform distribution random numbers chosen from the interval $[0, 1]$. The PSO algorithm uses these two updating mechanisms to achieve the optimization.

In this study, a modified PSO (MPSO) algorithm is taken into the phase response design of recursive all-pass

digital filter [11, 15]. The difference between the original and modification is to change the velocity formula. In the MPSO, the population needs to be further divided into some subpopulations at the beginning; for example, suppose that the initial population includes 50 particles and it will be divided into five subpopulations. Thus, the first subpopulation is composed of particles from number one to number ten, and the second then contains particles from number eleven to number twenty, and so forth. The best particle of each subpopulation needs to be recorded according to its objective function. Instead of the velocity formula of (6), the MPSO algorithm uses the following improved version:

$$v_{ij}(n+1) = wv_{ij}(n) + c_1r_1(p_{ij}(n) - a_{ij}(n)) + c_2r_2(g_j(n) - a_{ij}(n)) + c_3r_3(s_j(n) - a_{ij}(n)), \quad (8)$$

where s_j is a new variable called the local best and represents the position of the best particle of the subpopulation where the i th particle is located, c_3 is also a positive acceleration coefficient, and r_3 is a random number selected from the range $[0, 1]$ uniformly.

Design steps of MPSO-based for the phase response design of the recursive all-pass digital filter can be summarized in the following.

Data. Filter order N in (1) and (3), desired phase response θ_d and integral lower and upper bounds Ω_{\min} and Ω_{\max} in (5), number of particles (population size) P_s , number of subpopulations S , number of generations G , inertia weight w , and positive constants c_1 , c_2 , and c_3 in (8).

Goal. Derive a recursive all-pass digital filter with the phase response approaching the desired response θ_d .

- (1) Create an initial population consisting of P_s particles from the interval $[-1, 1]$ randomly.
- (2) Divide the population into S subpopulations by particle serial numbers.
- (3) If a prescribed number G of iterations are achieved, then the algorithm stops.
- (4) Evaluate the objective function of (5) for each particle and record the related individual best, local best, and global best positions.
- (5) Update each particle's velocity and position using (8) and (7).
- (6) Go back to Step (3).

4. Simulation Results

In this section, we consider two different examples with linear phase design to show the applicability of our proposed method [2, 5]. Some comparisons with the general PSO are also performed. In the PSO and MPSO, the variables of the algorithm are given by $w = 0.8$, $c_1 = c_2 = 0.5$, and $w = 0.8$, $c_1 = c_2 = c_3 = 0.5$, respectively, for all of the following simulations.

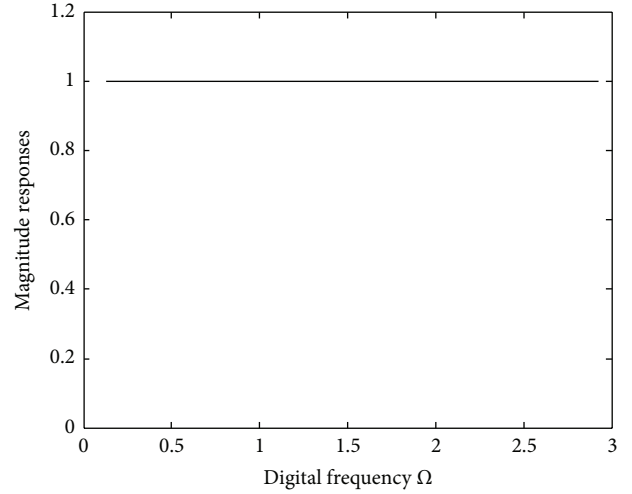


FIGURE 1: Magnitude response of Example 1.

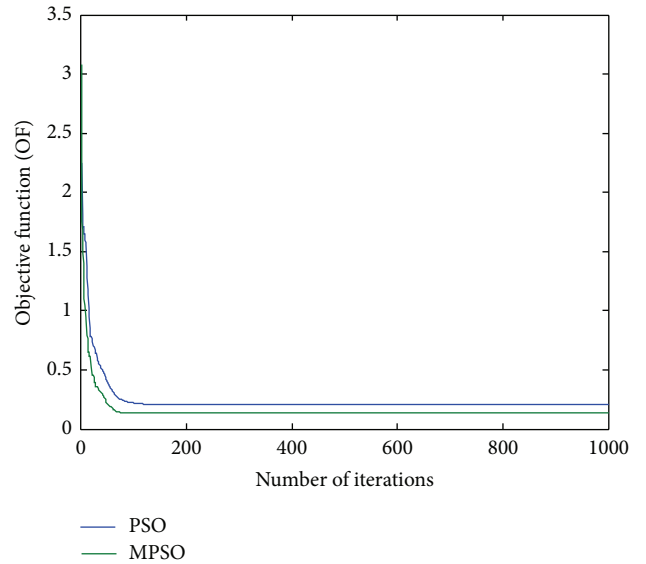


FIGURE 2: Trajectories of objective function (OF) for Run 1 of Example 1.

Example 1. In this example, the recursive all-pass filter is designed to approximate a desired Hilbert transformer whose phase response is given by

$$\theta_d(\Omega) = -N\Omega - \frac{\pi}{2}, \quad \Omega_{\min} \leq \Omega \leq \Omega_{\max}, \quad (9)$$

where the lower bound and upper bound are set to $\Omega_{\min} = 0.04\pi$ and $\Omega_{\max} = 0.94\pi$, respectively; N means the filter order and here it is chosen by $N = 10$. The magnitude response of such a recursive all-pass filter is plotted in Figure 1. In addition, the population size and number of generations are given by $P_s = 20$ and $G = 1000$ for the PSO and MPSO algorithm, and the number of subpopulations is simply set to $S = 4$ only for the MPSO. To verify the algorithm's robustness and efficiency, 20 independent runs with different initial conditions are executed for both

TABLE 1: Objective function values evaluated for Example 1.

	PSO algorithm	MPSO algorithm
Run 1	0.20570301	0.13606212
Run 2	0.13627085	0.13620043
Run 3	0.35740401	0.13679522
Run 4	0.13738652	0.13650124
Run 5	0.13672965	0.13680452
Run 6	0.14521057	0.13601797
Run 7	0.13619628	0.13673623
Run 8	0.15390549	0.13606745
Run 9	0.13755478	0.13762322
Run 10	0.14286368	0.13602069
Run 11	0.14405732	0.16636761
Run 12	0.14924150	0.13679615
Run 13	0.13622616	0.13703487
Run 14	0.13707200	0.13680303
Run 15	0.13634390	0.13701766
Run 16	0.21968176	0.22009637
Run 17	0.14488212	0.13649137
Run 18	0.14857613	0.13703097
Run 19	0.14250406	0.13624831
Run 20	0.27376677	0.14515192
Mean	0.16607883	0.14269337
Variance	0.00313722	0.00035958

TABLE 2: Digital filter coefficients derived by Run 1 of Example 1.

Filter coefficients	PSO algorithm	MPSO algorithm
a_1	-0.974	-0.9853
a_2	0.4350	0.4685
a_3	-0.3959	-0.4388
a_4	0.2323	0.2959
a_5	-0.1912	-0.2625
a_6	0.1048	0.1830
a_7	-0.0743	-0.1485
a_8	0.0268	0.0975
a_9	-0.0153	-0.0646
a_{10}	-0.0106	0.0290

algorithms. Final simulation results are listed in Tables 1 and 2 and shown in Figures 2–4, respectively. Table 1 lists the objective function values of 20 independent runs and it clearly reveals that the results by the MPSO are better than those by the PSO for most of independent runs. The mean and variance are evaluated by $\mu = 0.14269337$ and $\sigma^2 = 0.00035958$ for the MPSO and $\mu = 0.16607883$ and $\sigma^2 = 0.00313722$ for the PSO, respectively. To show the design outcomes, Figure 2 displays the convergence trajectories of the objective function for Run 1 of the proposed MPSO and PSO algorithms. As can be seen from Figure 2, the MPSO algorithm has a quicker convergence and lower objective function value than the PSO algorithm. Both phase responses and errors are further shown in Figures 3 and 4, respectively. A better simulation result can be obtained by the proposed

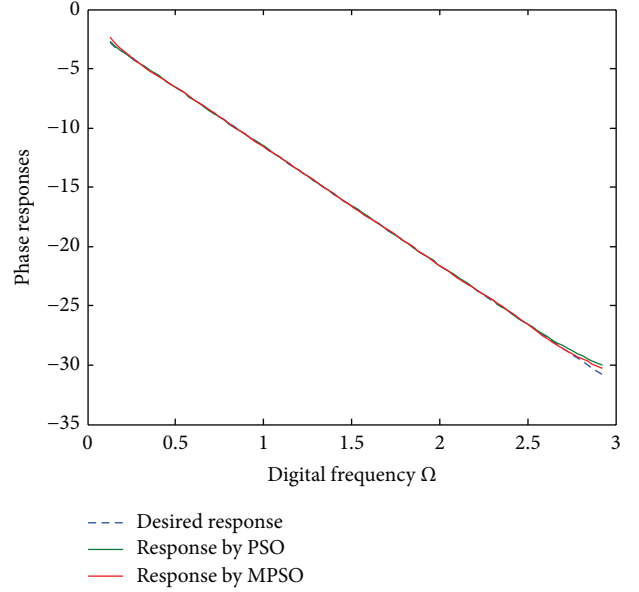


FIGURE 3: Phase responses for Run 1 of Example 1.

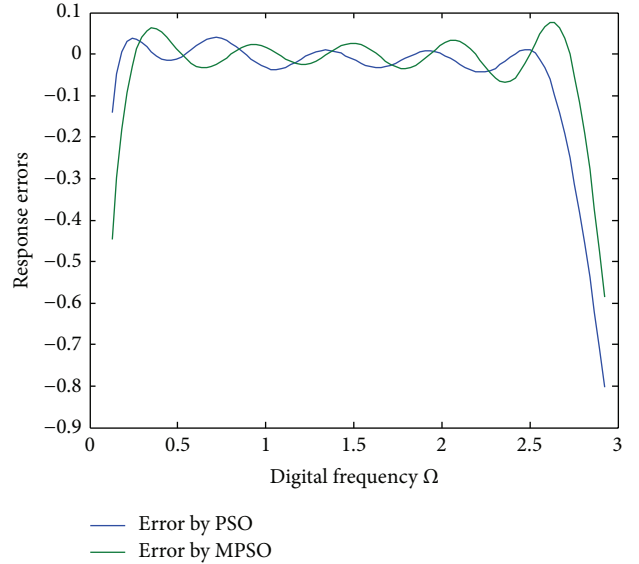


FIGURE 4: Phase response errors for Run 1 of Example 1.

method. In addition, all of digital filter coefficients derived by Run 1 of the PSO and MPSO algorithm are listed in Table 2 for comparisons.

Example 2. This example will design a recursive all-pass digital filter with a desired sinusoidal phase response expressed by

$$\theta_d(\Omega) = 4\pi(\cos \Omega - 1) - 52\Omega, \quad \Omega_{\min} \leq \Omega \leq \Omega_{\max}, \quad (10)$$

where $\Omega_{\min} = 0$ and $\Omega_{\max} = \pi$ are given. Its corresponding magnitude response is shown in Figure 5. In the simulation, a digital recursive filter with $N = 60$ is adopted and the population size and iterative number of the algorithms are

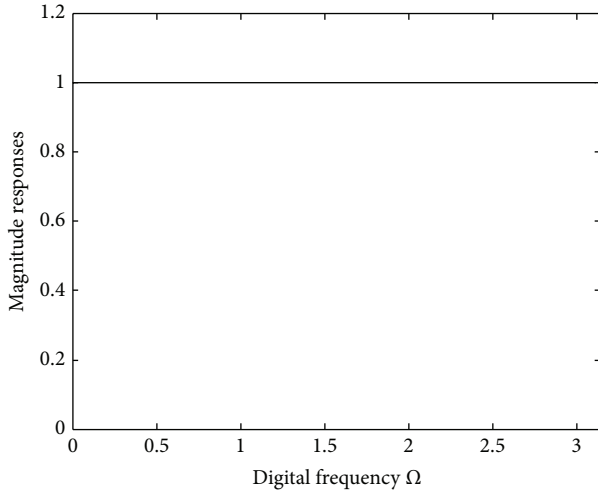


FIGURE 5: Magnitude response of Example 2.

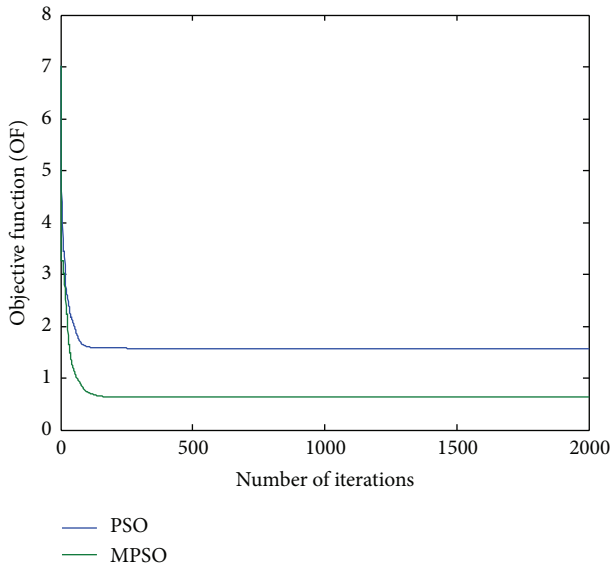


FIGURE 6: Trajectories of objective function (OF) for Run 1 of Example 2.

set to $P_s = 40$ and $G = 2000$, respectively, for solving such a higher-order digital filter. Moreover, as given in Example 1, the number of subpopulations is chosen by $S = 4$ and 20 independent runs with different sets of initial conditions are also performed for certifying the robustness of the algorithm. Table 3 lists a comparison of the objective function values evaluated by the proposed MPSO and PSO for Run 1 to Run 20, respectively. Some of digital filter coefficients derived are listed in Table 4 for comparisons. Figures 6–8 then show the related design outcomes only for Run 1 of the PSO and proposed algorithm. Again, it can be concluded from these results that the proposed MPSO is superior to the general PSO in the phase response design of recursive all-pass digital filter.

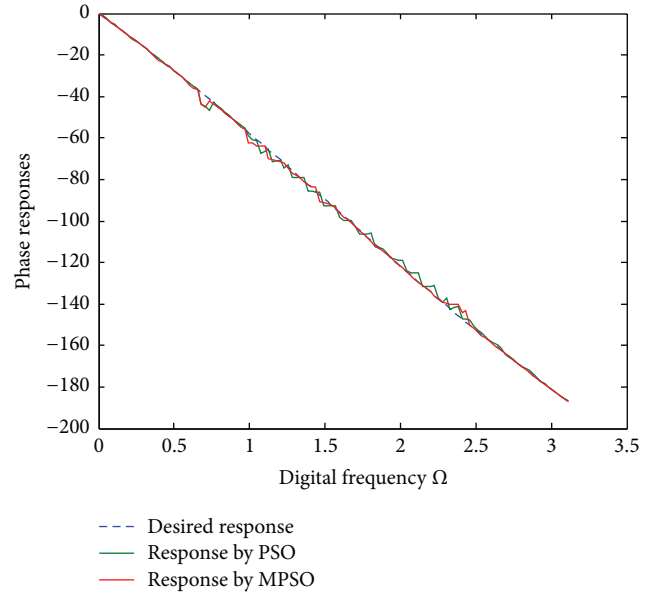


FIGURE 7: Phase responses for Run 1 of Example 2.

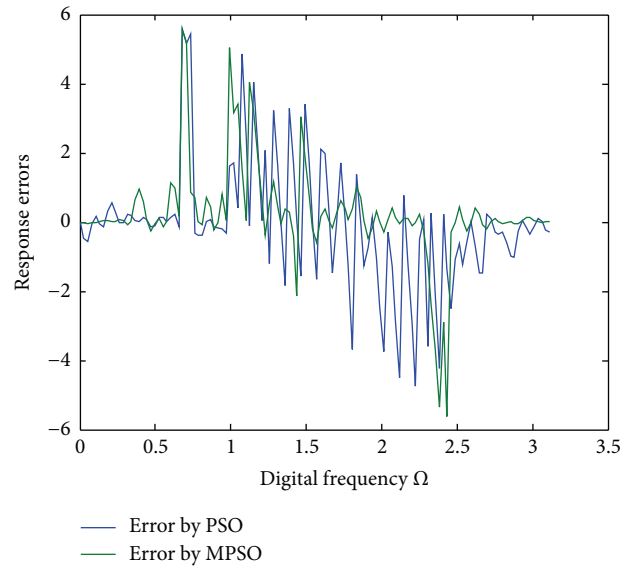


FIGURE 8: Phase response errors for Run 1 of Example 2.

5. Conclusions

This paper has developed a new design method for the phase response design of recursive all-pass digital filter. A modified PSO (MPSO) algorithm is suggested to design the filter coefficients such that the obtained phase response can approximate the desired response that is given previously. The difference between the MPSO and PSO is to modify the velocity updating formula of the algorithm. To improve the search capacity, a new factor of local-best particle for each subpopulation is introduced in the modified velocity formula. Finally, two different kinds of examples have been illustrated to verify the efficiency of the proposed method as compared

TABLE 3: Objective function values evaluated for Example 2.

	PSO algorithm	MPSO algorithm
Run 1	1.56939914	0.62645705
Run 2	1.85475888	0.57827778
Run 3	1.55263227	0.67573844
Run 4	1.57989126	0.67840205
Run 5	2.03270089	0.75624957
Run 6	1.63204094	0.67220282
Run 7	1.34828024	0.52125315
Run 8	1.65465543	1.18349923
Run 9	1.39729362	0.86232224
Run 10	1.55541996	0.65710581
Run 11	1.82677089	1.14715629
Run 12	1.68802195	0.62355711
Run 13	1.94169732	0.65469452
Run 14	1.61249872	0.48998564
Run 15	1.78693672	1.17072011
Run 16	1.75210045	1.02655545
Run 17	1.57667970	0.76927394
Run 18	1.64655678	0.80753180
Run 19	1.84461988	1.10135973
Run 20	1.56655437	0.44814752
Mean	1.67097547	0.77252451
Variance	0.02816276	0.05174895

TABLE 4: Digital filter coefficients derived by Run 1 of Example 2.

Filter coefficients	PSO algorithm	MPSO algorithm
a_1	0.0464	-0.1479
a_2	0.8023	1.1317
a_3	0.3584	0.0022
a_4	0.8848	0.8390
a_5	0.0292	-0.0636
a_6	0.4525	0.5609
\vdots	\vdots	\vdots
a_{58}	0.0093	0.0089
a_{59}	-0.1014	0.1059
a_{60}	0.1847	0.0181

with the general PSO algorithm. Simulation results have sufficiently proven that the proposed MPSO has a better design outcome than the PSO in the phase response design of recursive all-pass digital filter.

Conflict of Interests

The author declares that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

This work was partially supported by the Ministry of Science and Technology of Taiwan under Grant MOST 103-2221-E-366-004.

References

- [1] A. V. Oppenheim, R. W. Schaffer, and J. R. Buck, *Discrete-Time Signal Processing*, Prentice-Hall, London, UK, 2nd edition, 1999.
- [2] S. S. Kidambi, "Weighted least-squares design of recursive allpass filters," *IEEE Transactions on Signal Processing*, vol. 44, no. 6, pp. 1553–1557, 1996.
- [3] X. Zhang, "Design of IIR digital allpass filters based on eigenvalue problem," *IEEE Transactions on Signal Processing*, vol. 47, no. 2, pp. 554–559, 1999.
- [4] M. Lang, "Allpass filter design and applications," *IEEE Transactions on Signal Processing*, vol. 46, no. 9, pp. 2505–2514, 1998.
- [5] L. C. Su, Y. D. Jou, F. K. Chen, and C. M. Sun, "Neural type least squares design for IIR all-pass filters," in *Proceedings of the National Symposium on Telecommunications*, pp. 147–150, Taipei, Taiwan, 2007.
- [6] J.-W. Horng, "Current conveyors based allpass filters and quadrature oscillators employing grounded capacitors and resistors," *Computers and Electrical Engineering*, vol. 31, no. 1, pp. 81–92, 2005.
- [7] G. Stanić and S. Nikolić, "Digital linear phase notch filter design based on IIR all-pass filter application," *Digital Signal Processing*, vol. 23, no. 3, pp. 1065–1069, 2013.
- [8] W. D. Chang, S. T. Pan, K. H. Cheng, and M. C. Hsu, "Optimal design of allpass digital filters using artificial bee colony," in *Proceedings of the Workshop on Synthesis and System Integration of Mixed Information Technologies*, pp. 159–162, Beppu, Japan, 2012.
- [9] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, vol. 4, pp. 1942–1948, Perth, Australia, November–December 1995.
- [10] M. A. Abido, "Optimal design of power-system stabilizers using particle swarm optimization," *IEEE Transactions on Energy Conversion*, vol. 17, no. 3, pp. 406–413, 2002.
- [11] W.-D. Chang and S.-P. Shih, "PID controller design of nonlinear systems using an improved particle swarm optimization approach," *Communications in Nonlinear Science and Numerical Simulation*, vol. 15, no. 11, pp. 3632–3639, 2010.
- [12] J.-S. Chiou, S.-H. Tsai, and M.-T. Liu, "A PSO-based adaptive fuzzy PID-controllers," *Simulation Modelling Practice and Theory*, vol. 26, pp. 49–59, 2012.
- [13] M. A. Cavuslu, C. Karakuzu, and F. Karakaya, "Neural identification of dynamic systems on FPGA with improved PSO learning," *Applied Soft Computing Journal*, vol. 12, no. 9, pp. 2707–2718, 2012.
- [14] B. Vasumathi and S. Moorthi, "Implementation of hybrid ANNPSO algorithm on FPGA for harmonic estimation," *Engineering Applications of Artificial Intelligence*, vol. 25, no. 3, pp. 476–483, 2012.
- [15] W.-D. Chang, "Volterra filter modeling of nonlinear discrete-time system using improved particle swarm optimization," *Digital Signal Processing*, vol. 22, no. 6, pp. 1056–1062, 2012.
- [16] C. Zhang, M. Wu, and L. Luan, "An optimal PSO distributed precoding algorithm in QRD-based multi-relay system," *Future Generation Computer Systems*, vol. 29, no. 1, pp. 107–113, 2013.

- [17] S. J. Nanda and G. Panda, "Automatic clustering algorithm based on multi-objective Immunized PSO to classify actions of 3D human models," *Engineering Applications of Artificial Intelligence*, vol. 26, no. 5-6, pp. 1429–1441, 2013.
- [18] Z. Liu, H. Cao, X. Chen, Z. He, and Z. Shen, "Multi-fault classification based on wavelet SVM with PSO algorithm to analyze vibration signals from rolling element bearings," *Neurocomputing*, vol. 99, pp. 399–410, 2013.
- [19] Y. Wang, B. Li, T. Sun, K. Jiang, and X. Wang, "A modified PSO algorithm for aero-engine non-linear programming model," in *Proceedings of the 8th World Congress on Intelligent Control and Automation (WCICA '10)*, pp. 4069–4073, Jinan, China, July 2010.
- [20] S. Mondal, D. Chakraborty, R. Kar, D. Mandal, and S. P. Ghoshal, "Novel particle swarm optimization for high pass FIR filter design," in *Proceedings of the IEEE Symposium on Humanities, Science and Engineering Research (SHUSER '12)*, pp. 413–418, June 2012.
- [21] S. Mukherjee, R. Kar, D. Mandal, S. Mondal, and S. P. Ghoshal, "Linear phase low pass FIR filter design using improved particle swarm optimization," in *Proceedings of the IEEE Student Conference on Research and Development (SCORED '11)*, pp. 358–363, Cyberjaya, Malaysia, December 2011.