AMSE JOURNALS-AMSE IIETA publication-2017-Series: Advances C; Vol. 72; N°2; pp 114-139

Submitted Mar. 07, 2017; Revised May 27, 2017; Accepted Jun. 26, 2017

https://doi.org/10.18280/ama_c.720202

Artificial Raindrop Algorithm for Optimal Parameter Preference in Digital IIR Filters

*Yuansheng Huang, ** Ying Qiao

[*] Economics and Management, North China Electric Power University, Beijing, China

[**] Mathematics and Information Science, Beifang University of Nationalities, Ningxia, China

(xjqiao88@163.com)

Abstract

The system identification of digital Infinite Impulse Response (IIR) filter, as a key knowledge domain, is an important research subject in the automatic control field. However, the error surface of digital IIR filter is usually nonlinear and multimodal, which makes the cost function rather difficult to minimize. Whilst some global optimization techniques such as metaheuristic algorithms are essential for avoiding local minima encountered in conventional IIR modeling mechanisms. In this paper, Artificial Raindrop Algorithm (ARA), a metaheuristic approach recently developed as a member of the family of nonlinear optimization, is applied to identify the unknown parameters in the design of digital IIR filter. The ARA is inspired by the phenomenon of natural rainfall, whose components include the generation of raindrop, the descent of raindrop, the collision of raindrop, the flowing of raindrop and the updating of vapor. The paper studies algorithm's performance by a comparative law, aiming at eight primal intelligence optimization algorithms, as some state-of-the-art models, and eight improved metaheuristic algorithms. The experimental results show that ARA can more accurately identify the parameters as most of chosen and widely used cases and

may become a promising candidate for digital IIR filter.

Keywords

Design of digital IIR filter, Global optimization, Artificial raindrop algorithm, System identification.

1. Introduction

The design of Digital IIR filter, as a subject of pinnacle priorities in the science and engineering fields, has recently aroused wider concern in the circle since it can usually incur less computational cost and presents a much better performance than its equivalent Finite Impulse Response (FIR) filter [1]. To our knowledge, it has many important applications in the disciplines of communication [2], image processing [3], control systems [4], signal processing [5], and so on. However, the optimization of error surface on digital IIR filter is usually quite challenging, which makes the cost function difficult to minimize. Traditionally, the gradient-based design approaches such as the Least Mean Square (LMS) and its variants [6] have been widely used as optimization tools for IIR model identification where it features the simplicity implementation and low complexity. However, the error surface of IIR filter model is usually nonlinear and multimodal in terms of the filter coefficients, which may easily plunge these conventional gradient-based algorithms into local optimum. That is, they can only get the local optimal solution instead of global one in most cases. As a result, it is of great necessity to find or design an effective and robust global optimization algorithm in order to better identify the digital IIR filter models.

Everyone knows that the system identification of digital IIR filter can be essentially defined as a multidimensional optimization which is characterized as nonlinearity and multimodality. In this case, it is very natural for researchers to apply metaheuristic algorithms instead of gradient-based approach to capture these difficult optimization solutions by taking the advantage of its global search capacity. Some metaheuristic algorithms have recently been applied for this solution. Typically, Seeker Optimization Algorithm (SOA) [6], based on the concept of human search behavior simulation, is proposed for the purpose of design on digital IIR filter [7], Gravitational

Search Algorithm (GSA) [8], based on the law of gravity and mass interactions, is used for identification of such IIR filter [9], Cuckoo Search (CS) [10] which combines the breeding behaviors of Cuckoos with the L´evy flight which has been widely observed in some species, is applied to it [11].

But beyond that, some other models, such as Particle Swarm Optimization (PSO) [12], Artificial Bee Colony (ABC) [13], Flower Pollination Algorithm (FPA) [14], Teaching-Learning-Based Optimization (TLBO) [15], and Differential Evolution (DE) [16], have also been used as the optimization tools for the system identification of digital IIR filter [17-19]. According to the No Free Lunch Theorem (NFLT) [20], however, there is no explicit approach to be used for all optimization solutions.

In the current workspace, the artificial raindrop algorithm (ARA), a recently developed meta-heuristic approach, has been used as a new optimization tool for optimal parameter preference in the design of digital IIR filters, which is inspired by the natural rainfall phenomenon. The basic idea of ARA was first proposed in [21], and its availability has been proven in the parameter identification of nonlinear chaotic systems [22] and in the benchmark functions [23]. It has been found that ARA is a promising meta-heuristic approach, which has motivated researchers to apply it to optimization of digital IIR filters. The purpose of the paper is to show the performance and potentials of ARA to capture practical solutions. The NFLT also supports the research motivation since a new optimizer may perform the optimization better than other algorithms on the digital IIR filters.

The rest of this paper is outlined as follows: In Section 2, the problems occurred in the design of digital IIR filter are presented. In Section3, ARA and its implementation are briefly described. Then, in Section 4m, ARA is applied to the digital IIR filter, and the corresponding experimental results are obtained. Finally, in Section 5, the conclusion are drawn.

2. Design formulation

Consider the following IIR filter with the input-output relationship formulated by a difference equation [7]:

$$y(k)+\sum_{i=1}^{N} b_i y(k-i)=\sum_{i=0}^{M} a_i x(k-i) \tag{1}$$

where $x(k)$ and $y(k)$ are the input signal and output signal of the IIR filter at time k, respectively. M is the numerator orders (the order of the filter), M is the denominator orders $(N \geq M)$. a_i and b_i are adjustable filter coefficients, where $b_0 = 1$.

It is generally known that the transfer function of the filter model can be depicted as follows:

$$H(z)=\frac{\sum_{m=0}^{M} a_m z^{-m}}{1+\sum_{n=1}^{N} b_n z^{-n}} \tag{2}$$

As a consequence, the frequency response of IIR filter can be calculated as below:

$$H(\omega)=\frac{\sum_{m=0}^{M} a_m e^{-jm\omega}}{1+\sum_{n=1}^{N} b_n e^{-jn\omega}} \tag{3}$$

where $H(\omega)$ represents the Fourier transform vector, as the filter frequency response.

Assume the structure of filter system is known in advance. Then the estimated filter model can be stated as follows:

$$\hat{H}(\omega)=\frac{\sum_{m=0}^{M} \hat{a}_m e^{-jm\omega}}{1+\sum_{n=1}^{N} \hat{b}_n e^{-jn\omega}} \tag{4}$$

Where $\hat{a}_m, \hat{b}_n$ are the identifications of ideal filter coefficients $\hat{a}_m, \hat{b}_n$, respectively.

In calculation, the frequency ω can be sampled from interval [0, 1] with N points, which is denoted by $\omega = (\omega_1, \omega_2, \cdots, \omega_N)$. As a result, the magnitude response for the ideal filter and the estimated filter can be given as follows:

$$H(\omega) = (H(\omega_1), H(\omega_2), \cdots, H(\omega_N))^T \tag{5}$$

$$\hat{H}(\omega) = (\hat{H}(\omega_1), \hat{H}(\omega_2), \cdots, \hat{H}(\omega_N))^T \tag{6}$$

Besides, the $H(\omega)$ for LP, HP, BP and BS filters can be formulated in Eq.(7) to Eq.(10), respectively.

$$H(\omega) = \begin{cases} 1, 0 \le \omega \le \omega_C \\ 0, otherwise \end{cases} \tag{7}$$

$$H(\omega) = \begin{cases} 0, 0 \le \omega \le \omega_C \\ 1, otherwise \end{cases} \tag{8}$$

$$H(\omega) = \begin{cases} 1, \omega_{pl} \le \omega \le \omega_{ph} \\ 0, otherwise \end{cases} \tag{9}$$

$$H(\omega) = \begin{cases} 0, \omega_{pl} \le \omega \le \omega_{ph} \\ 1, otherwise \end{cases} \tag{10}$$

where is the cut-off frequency of the LP and HP filters, and are the upper and the lower pass band/stop bland edge frequencies of the BP and BS filters, respectively.

In order to have better control on the transition width and achieve higher stop band attenuation, the frequency cost function proposed in [7] is adopted and given in Eq.(11).

$$J = \frac{1}{S_p} \sum_{\omega_p} abs\left[abs\left(\left| \overline{H}(\omega) \right| - 1 \right) - \delta_p \right] + \frac{1}{S_s} \sum_{\omega_s} \left[abs\left(\left| \overline{H}(\omega) \right| \right) - \delta_s \right] \tag{11}$$

where, and are the ripples in the pass band and in the stop band, and are pass band and stop band, respectively.

Therefore, the parameter identification for a digital IIR filter can be formulated as follows: stop band normalized cut off frequencies, and are the numbers of samples used in the pass band and in the stop band, respectively.

Therefore, the parameter identification for a digital IIR filter can be formulated as follows:

$$\min\ J = \frac{1}{S_p}\sum_{\omega_p} abs\left[abs\left(\left|\overline{H}(\omega)\right|-1\right)-\delta_p\right]+\frac{1}{S_s}\sum_{\omega_s}\left[abs\left(\left|\overline{H}(\omega)\right|\right)-\delta_s\right]$$

$$s.t.\begin{cases} \overline{H}(\omega) = H(\omega)-\hat{H}(\omega) \\ \omega = (\omega_1,\omega_1,\cdots,\omega_N) \\ L \le \hat{a}_i,\hat{b}_j \le U, i = 0,1\cdots,M;\ j = 0,1\cdots,N. \end{cases} \tag{12}$$

It is obvious that parameter identification of digital IIR filter is a multi-dimensional $(M+N+1)$ continuous optimization of problem, $\left(\hat{a}_0,\hat{a}_1,\cdots,\hat{a}_M,\hat{b}_0,\hat{b}_1,\cdots,\hat{b}_N\right)$ is the decision vector and the optimization goal is to minimize *J*. *L* and *U* are the upper and lower boundary of $\hat{a}_i$ and $\hat{b}_j$, respectively.

3. Artificial raindrop algorithm

The basic physical principle is that ARA is inspired by the phenomenon of natural rainfall. The flow of the algorithm, as is intuitive and straightforward, follows the closed-loop journey of finite raindrops. The core idea is to trace the raindrops to where they are found in the lowest energy state but with the largest number-the Raindrop Pool (RP). And more specifically, the fitness value of a raindrop in ARA is evaluated at an appropriate altitude, and the lowest altitude corresponds to the optimal solution. The entire operating process of ARA can be divided into six phases: raindrop generation process $\rightarrow$raindrop descent process $\rightarrow$raindrop collision process $\rightarrow$raindrop flowing process $\rightarrow$RP updating process $\rightarrow$vapor updating process. The simulation scenario of ARA and its cyclical framework are respectively shown in Figure 1. (a) and in Figure 1. (b).

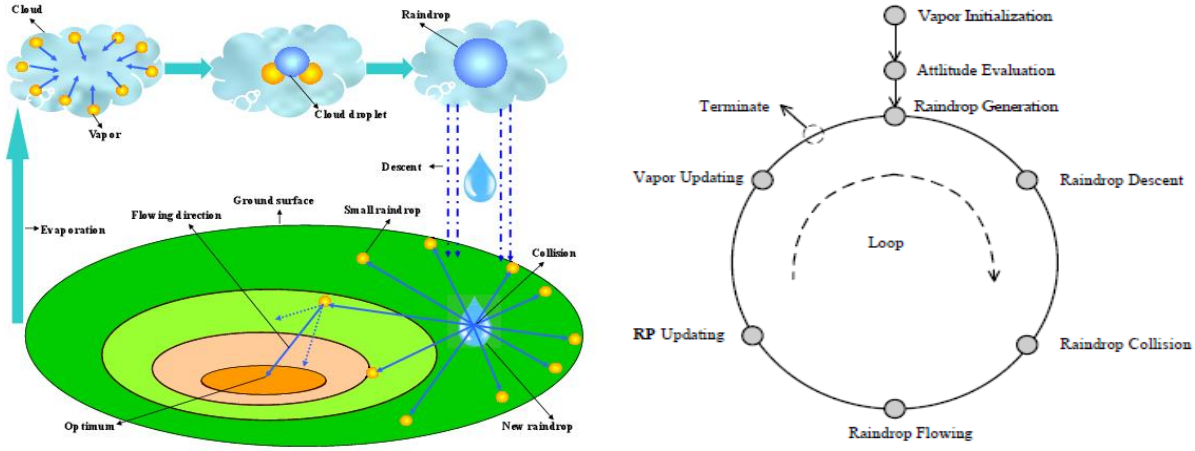


Fig 1. (a) The simulation scene graph of ARA [22] (b) The cyclical framework of ARA [23]

Like other population-based meta-heuristic algorithms, ARA starts by randomly placing some vapors in a search space, and each vapor corresponds to a point of the search space, as shown below:

$$Vapor_i = \left(x_i^1, \cdots, x_i^d, \cdots, x_i^D\right), i = 1,2, \cdots, N. \tag{13}$$

where N is the number of vapors (i.e. population size), D is the dimension of problem, and x_i^d is the position of the ith vapor in the dth dimension.

At a specific time t, the position of raindrop generation is considered as the geometric center of ambient vapors, which is defined as follows:

$$Raindrop(t) = \left(\frac{1}{N}\sum_{i=1}^{N} x_i^1(t), \cdots, \frac{1}{N}\sum_{i=1}^{N} x_i^d(t), \cdots, \frac{1}{N}\sum_{i=1}^{N} x_i^D(t)\right) \tag{14}$$

When the effect of external factors is not considered, the $Raindrop(t)$ will drop from the cloud to the ground by the free-fall motion. This implies that some component of $Raindrop(t)$ may be changed and the $Raindrop(t)$ will move to a new location denoted as

$New\_Raindrop(t)$. To be more specific, let $Raindrop^{(d_i)}(t)$ be the component of $Raindrop(t)$ in the $d_i th$ dimension, where $d_i (i = 1,2,3,4)$ is selected randomly from the set $\{1,2,\cdots,D\}$. Then $New\_Raindrop^{(d_1)}(t)$ is obtained by linear combination of $Raindrop^{(d_2)}(t)$, $Raindrop^{(d_3)}(t)$ and $Raindrop^{(d_4)}(t)$, and the other components of $New\_Raindrop(t)$ remain the same with $Raindrop(t)$. Thus, the $New\_Raindrop(t)$ can be formulated as the following descent operator of raindrop:

$$\begin{cases} \text{New\_Raindr op}^{(d)}(t) = \text{Raindrop}^{(d_2)}(t) + \phi \cdot \left(\text{Raindrop}^{(d_3)}(t) - \text{Raindrop}^{(d_4)}(t)\right), if\ d = d_1; \\ \text{New\_Raindr op}^{(d)}(t) = \text{Raindrop}^{(d)}(t), otherwise. \end{cases} \tag{15}$$

where ϕ is a uniform random number in the range $(-1,1)$, $d = 1,2,\cdots,D$. Accept $New\_Raindrop(t)$ if it locates at the lower elevations.

When the new raindrop falls to the ground, it will be split into a number of small raindrops because of the speed and the weight. Then these small raindrops will be splashing in all directions. During iteration, the number of these small raindrops is assumed to be equal to the population size in order to keep the stability of population scale, and the raindrop collision operator is depicted below:

$$Small\_Raindrop_{ij}(t) = Small\_Raindrop_j(t) + sign(a_j - 0.5) \cdot \log(\beta_j) \cdot \left(New\_Raindrop_j(t) - Vapor_{kj}(t)\right) \tag{16}$$

where $i(i = 1,2,\cdots,N)$ and $j(j = 1,2,\cdots,D)$ are the index of *ith* small raindrop and the corresponding dimension, respectively. $k \in \{1,2,\cdots,N\}$ is random index, α_j and β_j are two uniformly distributed random numbers in the range $(0,1)$ and sign $(\cdot)$ represents sign function.

Under the action of gravity, these small raindrops will flow from high altitude to low altitude,

and most of them eventually stop at thel lower altitude location (i.e. the better solutions). In the process of the evolution of the algorithm, these better solutions can provide additional information about the promising progress direction.

As a result, a RP is designed to track these lower positions found so far during the search, and the updating operation of RP is made as follows:

RP is initiated to any feasible solution of search space;

The optimal solution of current population is added to the raindrop pool after each iteration;

If the raindrop pool exceeds population N in size, then some solutions in the raindrop pool will be randomly deleted in order to keep the size of raindrop pool being stable and reduce calculation amount.

The flowing direction $d_i(t)$ of the *ith* small raindrop is also constructed by the linear combination of two vectors $d_{1i}(t)$ and $d_{2i}(t)$, which are further denoted as follows:

$$d_{1i}(t) = sign\left(F\left(RP_{k_1}(t)\right) - F\left(Small\_Raindrop_i(t)\right)\right) \cdot \left(RP_{k_1}(t) - Small\_Raindrop_i(t)\right) \tag{17}$$

$$d_{2i}(t) = sign\left(F\left(RP_{k_2}(t)\right) - F\left(Small\_Raindrop_i(t)\right)\right) \cdot \left(RP_{k_2}(t) - Small\_Raindrop_i(t)\right) \tag{18}$$

$$d_i(t) = \tau_1 \cdot rand_1 \cdot d_{1i}(t) + \tau_2 \cdot rand_2 \cdot d_{2i}(t) \tag{19}$$

where RP_{k_1} and RP_{k2} are any two of candidate solutions in $RP\left(k_1, k_2 \in \{1,2,\cdots,N\}\right)$, which can be chosen by the tournament selection procedure. $F(\cdot)$ represents fitness function, τ_1 and τ_2 are two step parameters of small raindrop stream, and $rand_1$ and $rand_2$ are two uniformly distributed random numbers in the range (0, 1).

Therefore, the descent operator of raindrop can be defined as follows:

$$New\_Small\_Raindrop_i(t) = Small\_Raindrop_i(t) + d_i(t), (i = 1,2,\cdots,N) \tag{20}$$

If the altitude of $New\_Small\_Raindrop_i$ is higher than that of $Small\_Raindrop_i$, the flowing direction $d_i(t)$ is considered as a wrong direction and terminated; otherwise, $Small\_Raindrop_i$ will flow along the direction and accept $New\_Small\_Raindrop_i$ if it gives a better fitness value. However, the $Small\_Raindrop_i(i=1,2,\cdots,N)$ could not have been flowing in a real environment. It is necessary to introduce a parameter Max_Flow_Number to control the maximum flowing times. Then, they will stay in the locations with a relatively lower elevation or evaporate after several flowing times, as shown in Figure 2:

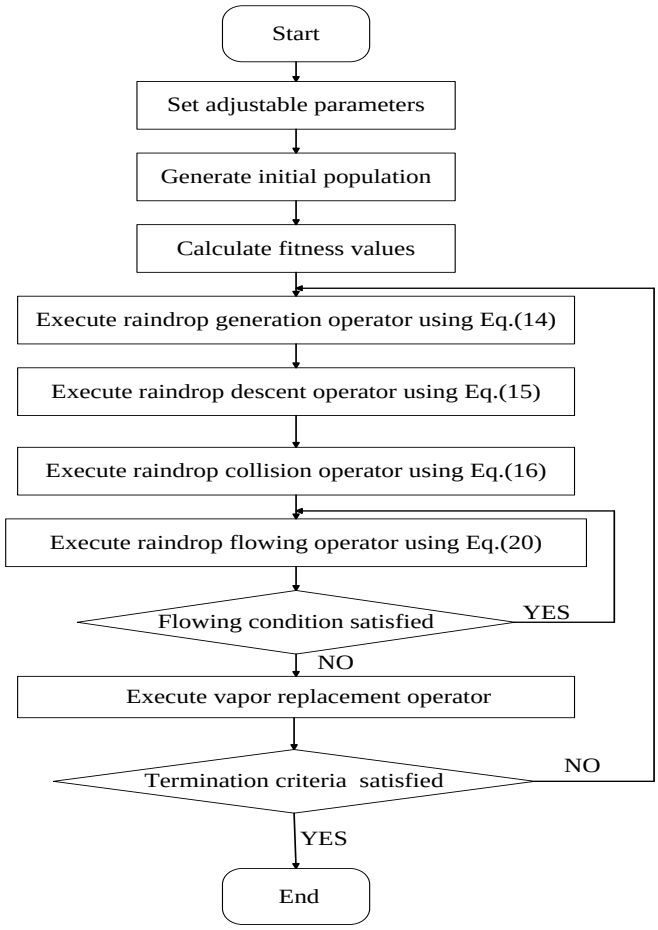


Fig 2. The flow chart of ARA

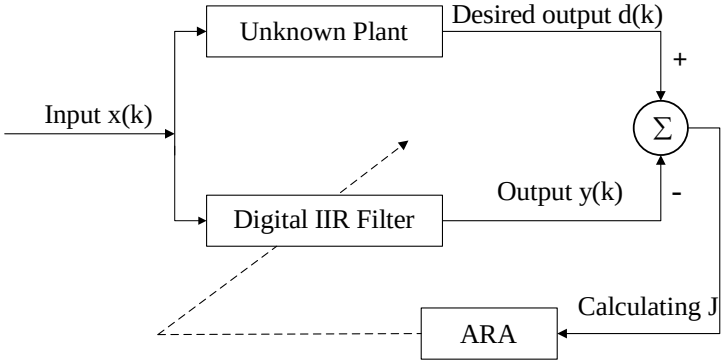


Fig 3. The principle of parameter identification for digital IIR filter using ARA

Finally, the water vapor will enter the atmosphere by evaporation to further form the new raindrop. In the design process of the algorithm, we select the N best individuals from $New\_Small\_Raindrop \bigcup Vapor$ using priority method as the next population.

4. Optimal design for digital IIR filter using ARA

4.1. Identification principle

In this section, the design of digital IIR filter discussed in Section 2 is converted to an ARA based optimization. More specifically, the estimated parameter vector of the IIR model is represented by N vapors in the search space, which are assigned by random initialization. The positions of the vapors are adjusted or updated using ARA to minimize the error occurred during evaluation of the IIR model. Besides the identification principle of digital IIR filter using an evolutionary algorithm can be illustrated in Figure 3.

As explained above, the procedures of ARA for digital IIR filter are given below:

Step 1. Initialize and set the control parameters of ARA:

$N, D, \lambda, \tau_1, \tau_2, Max\_Flow\_Number$;

Step 2. Randomly generate the initial population:

$Vapor(0) = \{Vapor_1(0), Vapor_2(0), \cdots, Vapor_N(0)\}$;

Step 3. Evaluate the initial population:

$J(Vapor(0)) = \{J(Vapor_1(0)), J(Vapor_2(0)), \cdots, J(Vapor_N(0))\}$ by the Eq.(11);

Step 4. Set $RP(0) = \arg\min_{1 \le i \le N} J(Vapor_i(0))$, and $t = 0$;

Step 5. Raindrop Generation: Get $Raindrop(t)$ by raindrop generation operator Eq.(14);

Step 6. Raindrop Descent: Get $New\_Raindrop(t)$ by raindrop descent operator Eq.(15);

Step 7. Raindrop Collision: Get $Small\_Raindrop_i(t)(i = 1,2,\cdots,N)$ by raindrop collision operator Eq.(16);

Step 8. Raindrop Flowing: Get $New\_Small\_Raindrop(t)(i = 1,2,\cdots,N)$ by raindrop flowing operator Eq.(20), and update each $Small\_Raindrop_i(t)$ by $New\_Small\_Raindrop_i(t)$;

Step 9. Raindrop Updating: Get Vapor(t+1) by applying raindrop updating operator to $Vapor(t) \cup Small\_Raindrop(t)$ and update Raindrop pool $RP(t)$;

Step 10. Termination Test: If termination condition is satisfied, export the vapor with the smallest objective function value in $Vapor(t+1)$, terminate the algorithm; otherwise, $t = t+1$, go to Step 5.

In order to thoroughly evaluate the performance of ARA on digital IIR filter, the results from ARA are compared with that of eight primal intelligence algorithms and eight improved metaheuristic algorithms for some typical problems arisen from IIR filter. The following sections first describe the problems about the test on IIR filter, performance metrics, and test settings used in this study. Then the paper presents and explores the identification results.

4.2. Problems of IIR filter

Example 1: In this example [24], the transfer function of unknown plant is unknown and the specifications are given in Table 1. In this test, the filter orders for IIR LP, HP, BP and BS are set to 8, and the number of samples is set to 128.

Example 2: In this example, a second-order IIR filter is taken from [25], and its transfer function is defined below:

$$H(z)=\frac{1}{1-1.2z^{-1}+0.6z^{-2}} \tag{21}$$

Tab 1. Specifications of IIR LP, HP, BP and BS.

Type of filter	Pass band ripple δ_p	band ripple δ_s	Pass band edge ω_p	Stop band edge ω_s
low-pass (LP)	0.01	0.001	0.45	0.5
high-pass (HP)	0.01	0.001	0.5	0.45
band-pass (BP)	0.01	0.001	0.3 and 0.7	0.25 and 0.75
band-stop (BS)	0.01	0.001	0.25 and 0.75	0.3 and 0.7

The second order plant $H(z)$ can be estimated by second order IIR filter $\hat{H}(z)$. The transfer function of the estimated IIR filter model can be hereby assumed as

$$\overline{\hat{H}}(z)=\frac{1}{1+b_1z^{-1}+b_2z^{-2}} \tag{22}$$

In this test, the input $x(k)$ in the system and in the filter is a white noise sequence, which is $L=100$ in length.

Example 3: In this example, a second order IIR plant is considered [26], and its transfer function is stated as follows:

$$\hat{H}(z)=\frac{0.05-0.4z^{-1}}{1-1.131z^{-1}+0.25z^{-2}} \tag{23}$$

In the case, a second order plant can be estimated by a first order IIR filter defined in Eq. (24):

$$\hat{H}(z)=\frac{a_0}{1+b_1z^{-1}} \tag{24}$$

In this test, the input $x(k)$ in the system and in the filter is a white-noise sequence, whose length L is set to 100.

Example 4: In this example, a third order IIR plant (taken from [17]) is considered, and its transfer function is defined as below:

$$H(z)=\frac{-0.2-0.4z^{-1}+0.5z^{-2}}{1-0.6z^{-1}+0.25z^{-2}-0.2z^{-3}} \tag{25}$$

In the case, the third order plant $H(z)$ can be modelled using a third order IIR filter $\hat{H}(z)$ described in Eq. (26):

$$\hat{H}(z)=\frac{a_0+a_1z^{-1}+a_2z^{-2}}{1-b_1z^{-1}+b_2z^{-2}+b_3z^{-3}} \tag{26}$$

In this test, the system input is $x(k)$, a unit-variance of white Gaussian pseudonoise sequence, whose length L is set to 100.

Example 5: In this example, a third order IIR plant (taken from [7]) is considered, and its transfer function is defined as follows:

$$H(z)=\frac{-0.3+0.4z^{-1}-0.5z^{-2}}{1-1.2z^{-1}+0.5z^{-2}-0.1z^{-3}} \tag{27}$$

In the case, the third order plant $H(z)$ can be modelled using a second order IIR filter $\hat{H}(z)$ described in Eq. (28):

$$\hat{H}(z)=\frac{a_0+a_1z^{-1}}{1+b_1z^{-1}+b_2z^{-2}} \tag{28}$$

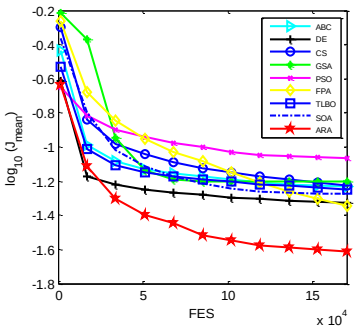
In this test, the system input $x(k)$ is a white-noise sequence uniformly distributed, taking values from $(-0.5,0.5)$, where $SNR = 30dB$; the data length L is set to 100.

Tab 2. The results of SOA, GSA, CS, PSO, ABC, FPA, TLBO, DE and ARA over 25 independent tests on eight IIR filter with D*10000 FES

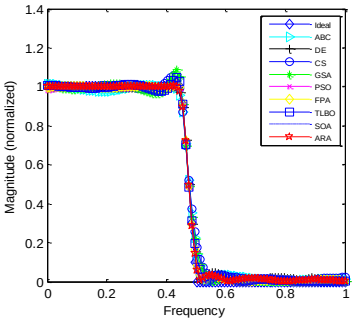
IIRProblem	Result	SOA	GSA	CS	PSO	ABC	FPA	TLBO	DE	ARA
Exa mple1 (LP)	J_{mean}	5.31E-02‡	6.26E-02‡	5.93E-02‡	8.59E-02‡	5.83E-02‡	4.51E-02‡	5.60E-02‡	4.70E-02‡	**2.44E-02**
	S td	3.05E-02	2.76E-02	1.55E-02	3.33E-02	**1.39E-02**	2.35E-02	2.95E-02	2.34E-02	2.02E-02
	p	5.12E-06	2.45E-07	8.55E-08	9.51E-08	9.51E-08	2.85E-04	4.67E-06	3.02E-05	–
	h	1	1	1	1	1	1	1	1	–
Exa mple1 (HP)	J_{mean}	4.27E-02‡	5.41E-02‡	5.55E-02‡	9.62E-02‡	6.16E-02‡	4.60E-02‡	5.07E-02‡	4.62E-02‡	**2.19E-02**
	S td	2.08E-02	2.40E-02	1.28E-02	4.24E-02	1.89E-02	2.17E-02	2.46E-02	2.19E-02	**1.17E-02**
	p	2.78E-05	5.56E-07	1.46E-08	8.28E-09	3.21E-08	1.55E-04	8.10E-06	2.15E-05	–
	h	1	1	1	1	1	1	1	1	–
Exa mple1 (BP)	J_{mean}	1.84E-01‡	1.87E-01‡	1.90E-01‡	2.34E-01‡	1.98E-01‡	1.62E-01‡	1.81E-01‡	1.84-01‡	**1.57E-01**
	S td	2.53E-02	2.57E-02	**1.11E-02**	2.99E-02	1.41E-02	1.83E-02	1.82E-02	1.99E-02	2.41E-02
	p	1.46E-03	3.31E-04	5.12E-06	1.17E-08	1.45E-07	6.70E-01	5.14E-04	2.85E-04	–
	h	1	1	1	1	1	0	1	1	–
Exa mple1 (BS)	J_{mean}	1.32E-01‡	1.52E-01‡	1.54E-01‡	2.24E-01‡	1.61E-01‡	1.23E-01‡	1.24E-01‡	1.19E-01‡	**8.11E-02**
	S td	3.69E-02	6.10E-02	**2.53E-02**	8.28E-02	2.82E-02	4.22E-02	5.21E-02	4.87E-02	4.04E-02
	p	5.91E-05	1.97E-05	3.34E-07	3.21E-08	1.18E-07	1.37E-03	3.19E-03	3.19E-03	–
	h	1	1	1	1	1	1	1	1	–
Exa mple 2	J_{mean}	**0.00E+0§**	4.11E-19‡	8.78E-20‡	1.72E-09‡	2.41E-12‡	1.35E-15‡	**0.00E+00§**	**0.00E+00§**	**0.00E+00**
	S td	**0.00E+00**	4.30E-19	1.66E-19	5.58E-09	3.51E-12	3.15E-15	**0.00E+00**	**0.00E+00**	**0.00E+00**
	p	NaN	9.73E-11	9.73E-11	9.73E-11	9.73E-11	9.73E-11	**NaN**	**NaN**	–
	h	0	1	1	1	1	1	0	0	–
Exa	J_{mean}	**1.78E-01**†	1.83E-01†	1.79E-01†	2.29E-01†	1.88E-01†	1.99E-01†	1.88E-01†	1.84E-01†	2.35E-01

mple 3	*S td*	9.38E-17	2.04E-03	3.05E-17	1.24E-08	1.80E-08	4.91E-15	1.00E-16	**1.50E-17**	6.66E-17
	p	3.36E-10	9.29E-10	6.65E-10	9.30E-10	8.27E-10	9.27E-10	3.44E-10	3.51E-10	–
	h	1	1	1	1	1	1	1	1	–
	J_{mean}	**0.00E+00**†	2.01E-19‡	2.83E-28‡	8.18E-09‡	2.72E-21‡	1.24E-17‡	2.74E-34†	2.94E-06‡	6.85E-33
Exa	*S td*	**0.00E+00**	8.15E-20	1.11E-27	1.13E-08	6.92E-21	1.39E-17	1.37E-33	1.47E-05	1.00E-32
mple 4	*p*	7.54E-06	1.21E-09	1.36E-09	1.21E-09	1.21E-09	1.21E-09	3.70E-05	6.85E-03	–
	h	1	1	1	1	1	1	1	1	–
	J_{mean}	1.28E-02‡	1.46E-02‡	1.31E-02‡	1.82E-02‡	1.11E-02†	1.65E-02‡	1.39E-02‡	**9.74E-03**†	1.19E-02
Exa	*S td*	6.56E-18	6.91E-18	**1.62E-18**	6.12E-03	1.97E-05	2.11E-15	3.56E-18	6.23E-09	5.47E-18
mple 5	*p*	9.86E-10	7.33E-10	4.64E-10	1.08E-09	1.08E-09	1.07E-09	7.19E-10	6.19E-10	–
	h	1	1	1	1	1	1	1	1	–
		5	7	7	7	6	6	5	5	
		2	1	1	1	2	1	2	2	
		1	0	0	0	0	1	1	1	

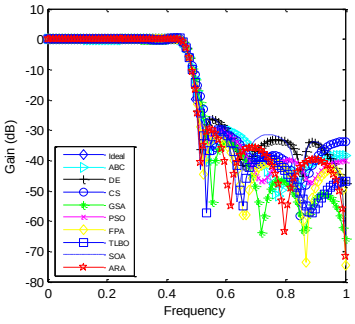
"‡", "†", and "§" denote that the performance of ARA is better than, worse than, and equivalent to that of the other algorithm, respectively.



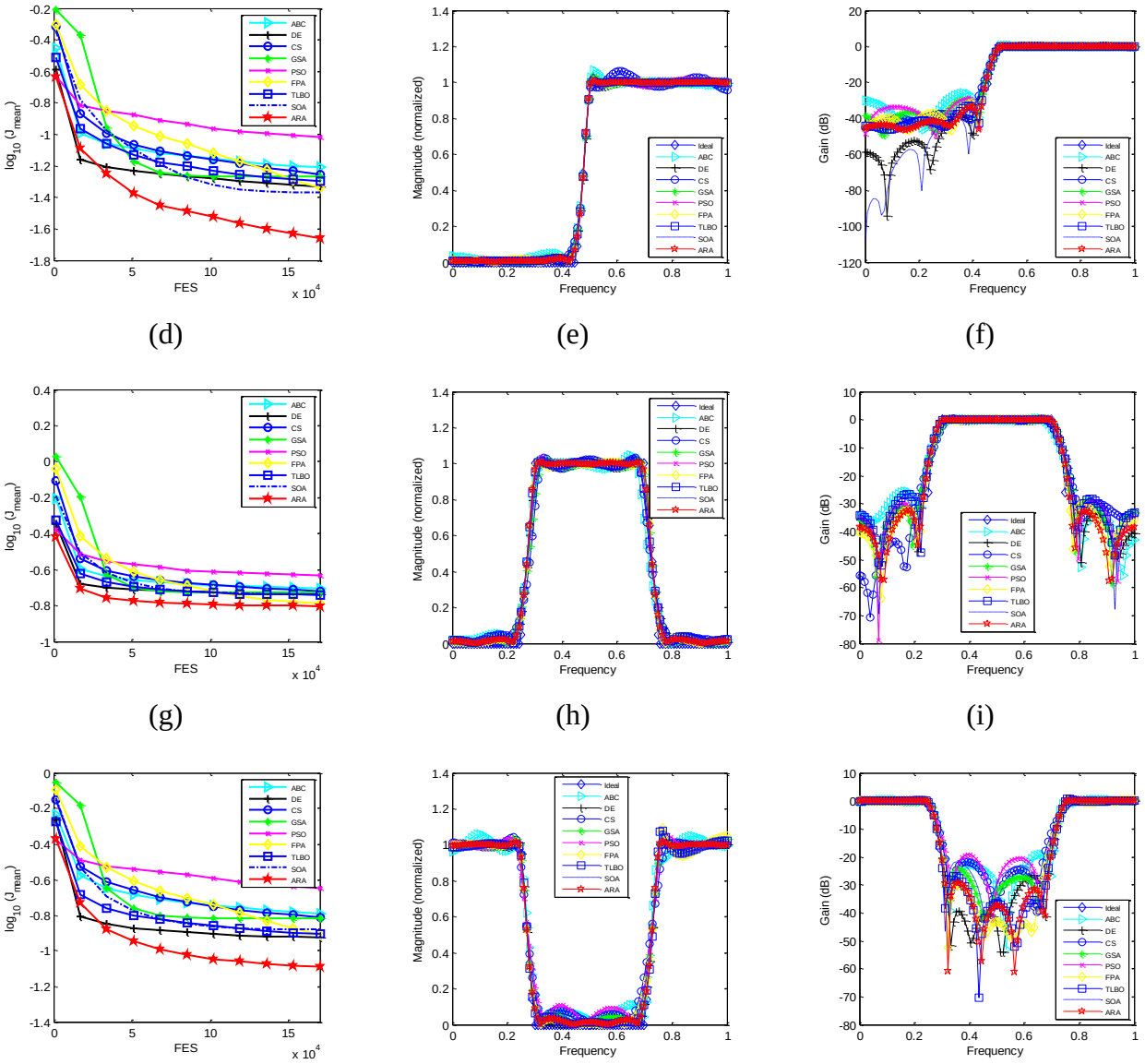
(a)



(b)



(c)



(d) (e) (f)

(g) (h) (i)

4.3. Performance metric and test settings

25 independent tests are carried out on one machine with a Celoron 3.40 GHz CPU, 4GB memory, and windows 7 operating system with Matlab 7.9, and conducted with the maximum number of function evaluations (MAX_FES) as the termination criterion. In the test, the mean error value (Jmean) andits standard deviation (Std) are recorded for determining the performance of each algorithm. To statistically compare ARA to its peers, the Wilcoxon's rank sum test (nonparametric statistical test) [27] at a 0.05 significance level is usually used as a statistical tool to determine whether the median fitness values of two sets of results are statistically different from each other. A $p-value$ less than 0.05 means that the performances of two competitive algorithms statistically

differ with 95% certainty $(h=1)$, or the performances of two competitive algorithms have no significant difference $(h=0)$. Whilst the mean error values in 25 runs for each problem at specified checkpoint (0.01,0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0)* MAX_FES are recorded to reveal the convergence characteristics of algorithms.

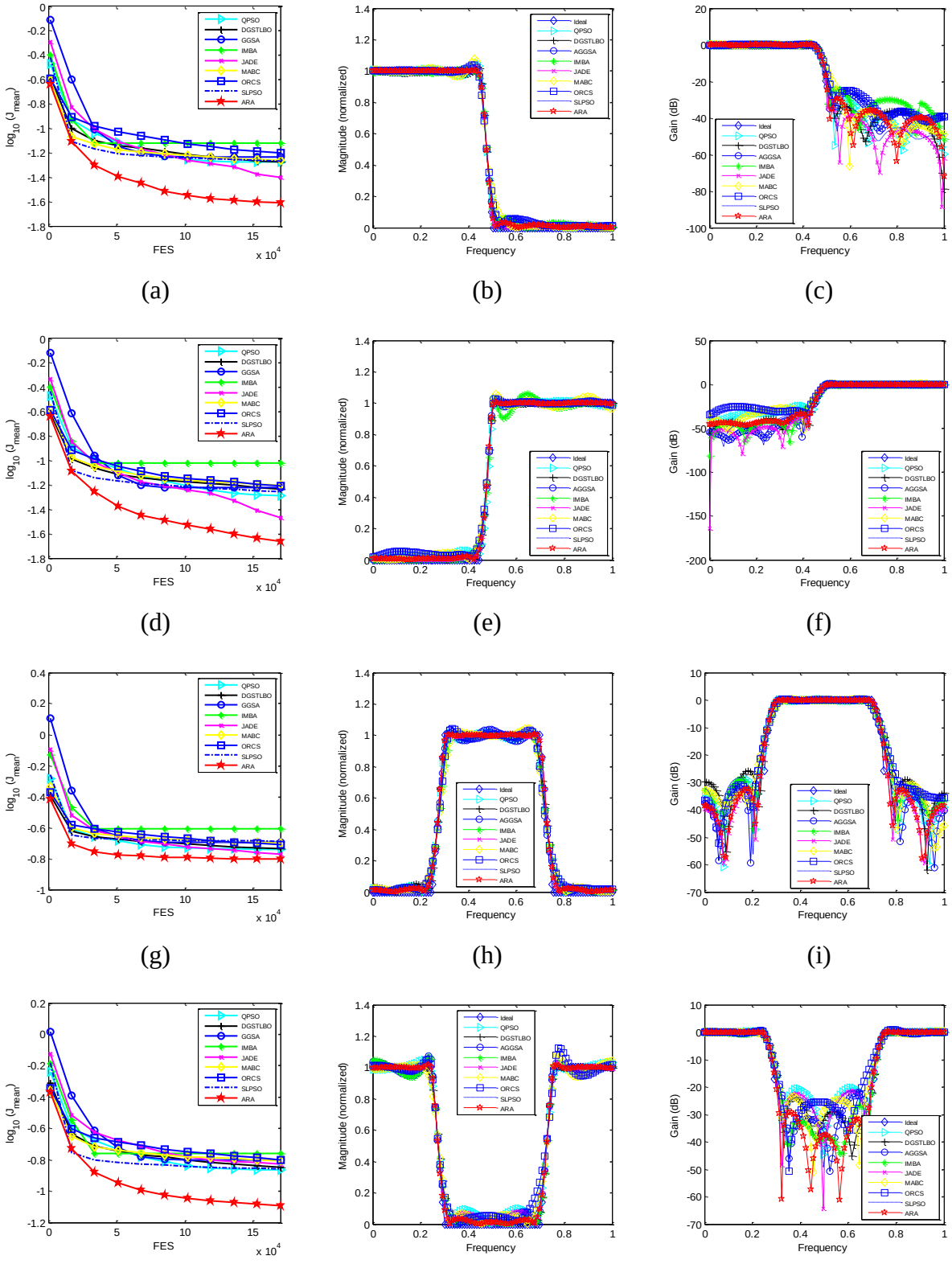
In the numerical tests, MAX_FES is set to D*10000 for all algorithms. To our knowledge, there are also several studies on the control parameters of PSO and DE. As suggested in [28], the linearly decreasing inertia ω from 0.9 to 0.4 is adopted for the search, and the acceleration coefficients c_1, c_2 are both set to 1.49445 in PSO. For DE, as suggested in [29], the scaling factor F and crossover probability CR are set to 0.4717 and 0.8803, respectively. The parameters of other algorithms agree well with the original literatures.

Tab.3. The results of DGSTLBO, AGGSA, IMBA, JADE, MABC, ORCS, QPSO, SLPSO and ARA over 25 independent tests on eight IIR filter with D_10000 FES.

IIRProblem	Result	DGSTLBO	AGGSA	IMBA	JADE	MABC	ORCS	QPSO	SLPSO	ARA
	J_{mean}	5.28E-02‡	5.85E-02‡	7.62E-02‡	3.97E-02‡	5.47E-02‡	6.29E-02‡	5.29E-02‡	5.43E-02‡	**2.44E-02**
Example1	*S td*	2.15E-02	2.45E-02	2.60E-02	2.89E-02	**9.40E-03**	1.31E-02	1.64E-02	2.81E-02	2.02E-02
(LP)	*p*	2.20E-06	1.36E-06	6.18E-08	1.87E-02	6.18E-08	4.46E-08	6.80E-07	8.10E-06	–
	h	1	1	1	1	1	1	1	1	–
	J_{mean}	5.86E-02‡	5.97E-02‡	9.54E-02‡	3.39E-02‡	6.28E-02‡	6.24E-02‡	5.17E-02‡	5.57E-02‡	**2.19E-02**
Example1	*S td*	3.40E-02	2.59E-02	3.47E-02	2.52E-02	1.65E-02	**1.12E-02**	3.14E-02	2.98E-02	1.17E-02
(HP)	*p*	1.27E-05	1.62E-07	4.13E-09	3.83E-02	5.21E-09	2.90E-09	2.15E-05	4.26E-06	–
	h	1	1	1	1	1	1	1	1	–
	J_{mean}	1.84E-01‡	2.01E-01‡	2.48E-01‡	1.69E-01‡	2.00E-01‡	1.96E-01‡	1.82E-01‡	2.05-01‡	**1.57E-01**
Example1	*S td*	2.89E-02	2.31E-02	4.00E-02	2.14E-02	**1.34E-02**	1.50E-02	1.68E-02	4.63E-02	2.41E-02
(BP)	*p*	7.35E-04	1.65E-06	1.17E-08	1.81E-02	7.68E-08	6.80E-07	3.84E-04	2.15E-05	–
	h	1	1	1	1	1	1	1	1	–
Example1	J_{mean}	1.42E-01‡	1.58E-01‡	1.73E-01‡	1.50E-01‡	1.57E-01‡	1.59E-01‡	1.37E-01‡	1.39E-01‡	**8.11E-02**

(BS)	*S td*	5.74E-02	5.55E-02	6.07E-02	**3.12E-02**	2.33E-02	2.48E-02	4.18E-02	4.90E-02	4.04E-02
	p	2.45E-04	1.38E-05	5.56E-07	2.66E-06	6.80E-07	2.72E-07	3.02E-05	2.78E-05	–
	h	1	1	1	1	1	1	1	1	–
	J_{mean}	1.31E-31§	2.55E-22‡	1.48E-04‡	1.13E-31‡	**0.00E+00§**	**0.00E+00§**	**0.00E+00§**	**0.00E+00§**	**0.00E+00**
Example 2	*S td*	3.51E-31	3.28E-22	1.38E-04	2.81E-31	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	p	4.11E-02	9.73E-11	9.73E-11	4.12E-02	**NaN**	**NaN**	**NaN**	**NaN**	–
	h	0	1	1	1	0	0	0	0	–
	J_{mean}	2.01E-01†	1.73E-01†	1.89E-01†	1.95E-01†	1.78E-01†	**1.29E-01**†	**1.29E-01**†	2.66E-01‡	2.35E-01
Example 3	*S td*	4.24E-17	3.76E-17	8.72E-07	6.31E-17	4.16E-17	**3.30E-17**	1.01E-02	1.12E-16	6.66E-17
	p	5.70E-10	4.52E-10	9.30E-10	3.51E-10	4.12E-10	3.51E-10	5.58E-10	8.10E-11	–
	h	1	1	1	1	1	1	1	1	–
	J_{mean}	1.40E-29‡	6.92E-23‡	3.19E-17‡	0.00E+00†	**0.00E+00**†	**0.00E+00**†	**0.00E+00**†	**0.00E+00**†	6.85E-33
Example 4	*S td*	6.98E-29	4.68E-23	2.41E-17	0.00E+00	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.00E-32
	p	7.24E-04	1.21E-09	1.21E-09	7.54E-06	7.54E-06	7.54E-06	7.54E-06	7.54E-06	–
	h	1	1	1	1	1	1	1	1	–
	J_{mean}	1.49E-02‡	1.17E-02†	1.27E-02‡	1.13E-02†	1.41E-02‡	**1.09E-02**†	1.26E-02‡	1.49E-02‡	1.19E-02
Example 5	*S td*	9.57E-18	1.58E-18	9.25E-04	**2.24E-18**	1.77E-05	1.87E-03	2.11E-04	1.64E-17	5.47E-18
	p	5.97E-10	6.68E-10	1.08E-09	6.79E-10	1.08E-09	7.42E-03	9.83E-10	1.07E-09	–
	h	1	1	1	1	1	1	1	1	–
		6	6	7	6	5	4	5	6	
		1	2	1	2	2	3	2	1	
		1	0	0	0	1	1	1	1	

"‡", "†", and "§" denote that the performance of ARA is better than, worse than, and equivalent to that of the other algorithm, respectively.



4.4. Results of simulation and comparisons

In this section, the performance of ARA is compared with eight primal intelligence algorithms and eight improved metaheuristic algorithms. The statistical results, in terms of the mean value

(*Jmean*) and standard deviation (*Std*) obtained from 25 independent tests by each algorithm, are reported in Table 1 and in Table 2, respectively. The test results are listed in the last three rows of Table 1 and Table 2. It is obvious that the best statistical results among nine algorithms in each table are highlighted in boldface.

The ARA is compared to eight primal intelligence algorithms named SOA [6], GSA [8], CS [10], PSO [12], ABC [13], FPA [14], TLBO [15] and DE [16]. The statistical results are listed in Table1 where the means and standard deviations are concerned. Based on the results from Wilcoxon's rank sum test, it is noted that the performance of ARA is significantly better than that of other eight primal algorithms as the optimal value is reported. More specifically, ARA outperforms SOA by five, GSA by seven, CS by seven, PSO by seven, ABC by six, FPA by six, TLBO by five, DE by five out of eight problems occurred in the IIR filter, respectively. Whereas, SOA, GSA, CS, PSO, ABC, FPA, TLBO, DE perform better than ARA only in one or two aspects, respectively.

When the ARA is compared to eight improved metaheuristic algorithms: in order to comprehensively evaluate the performance of ARA, here come the other eight improved intelligence algorithms called teaching-learning-based optimization with dynamic group strategy (DGSTLBO) [30], adaptive gbest-guided gravitational search algorithm (AGGSA) [31], improved mine blast algorithm (IMBA) [32], adaptive differential evolution (JADE) [33], modified artificial bee colony (MABC) [34], one rank cuckoo search algorithm (ORCS) [35], quantum-behaved particle swarm optimization (QPSO) [36] and social learning particle swarm optimization (SLPSO) [37]. It is worth mentioning that JADE proposed by J. Q. Zhang and A. C. Sanderson is a more popular and efficient algorithm in the area. In JADE, a novel mutation strategy "DE/current-to-pbest" with optional external archive can improve optimization performance. From the statistical results of Table 2, it is clear that the ARA performs significantly better than DGSTLBO, AGGSA, IMBA, JADE, MABC, ORCS, QPSO and SLPSO after the Wilcoxon's rank sum test. To make it more concrete, ARA outperforms DGSTLBO by six, AGGSA by six, IMBA by seven, JADE by six, MABC by five, ORCS by four, QPSO by five, and SLPSO by six out of eight problems simulated on the IIR filter, respectively. Instead, DGSTLBO, AGGSA, IMBA, JADE, MABC,

ORCS, QPSO, SLPSO perform better than ARA only in one, two or three aspects, respectively.

From the above results from comparison, the overall performance of ARA is the best among the sixteen intelligence algorithms in the terms of calculation accuracy. It is suggested that ARA have a better exploration and exploitation capacities. However, what we must point out is that ARA has a higher time complexity than some popular heuristic algorithms such as PSO and DE [23]. But beyond that, in order to intuitively show the effectiveness of ARA, there are some classic simulations of ARA and other competitive algorithms, as shown in Figure 3 and in Figure 4, from where we can see that ARA has a faster convergence speed than that of other algorithms, and can identify the system characteristics quite well.

Conclusion

The ARA is a recently developed meta-heuristic optimization algorithm based on simulating the flow process of raindrop. The result shows that ARA can be considered as a new computation tool for various designs and optimization tasks. In this paper, a design approach for ARA-based digital filter has been proposed, and some benefits of ARA for designing digital IIR filters have been explored. The simulation and comparison results reveal that ARA has overall better performance than eight primal intelligence algorithms and eight improved meta-heuristic algorithms so that it is best for most of the chosen and widely used cases in this paper. Thus, it is believed that ARA will become a promising candidate for design process of digital IIR filter.

In the future, much work remains on the table. Some important and interesting research problems include (i) the ARA is applied to solve other benchmark functions for further testing its performance, and (ii) the ARA is streamlined to solve multi-objective optimization problems.

Acknowledgments

This work is partly supported by the National Natural Science Foundation of China under Project Code (61561001), Beifang University of Nationalities under Project Code (21500880).

References

1. J. Luukko, K. Rauma, Open-loop adaptive filter for power electronics applications, Feb. 2008, IEEE Transactions on Industrial Electronics, vol. 55, no.2, pp. 910-917
2. M. Albaghdadi, B. Briley, M. Evens, Event stormdetection and identification in communication systems, May. 2006, Reliability Engineering & System Safety, vol. 91, no. 5, pp. 602-613.
3. H.-C. Chung, J. Liang, S. Kushiyama, M. Shinozuka, Digital image processing for non-linear system identification, Jul. 2004,International Journal of Non-Linear Mechanics, vol. 39, no. 5, pp. 691-707.
4. X. Zhou, C. Yang, W. Gui, Nonlinear system identification and control using state transition algorithm, Jan. 2014,Applied Mathematics and Computation, vol. 226, pp. 169-179.
5. P. FrankPai, B.-A. Nguyen, M. J. Sundaresan, Nonlinearity identification by time-domain-only signal processing,Sep. 2013, International Journal of Non-Linear Mechanics, vol. 54, pp. 85-98.
6. C. H. Dai, W. R. Chen, Y. H. Song, Y. H. Zhu, Seeker optimization algorithm: a novel stochastic search algorithm for global numerical optimization, Apr. 2010, Journal of Systems Engineering and Electronics, vol. 21, no. 2, pp. 300-311.
7. C. H. Dai, W. R. Chen, Y. H. Zhu, Seeker optimization algorithm for digital IIR filter design, May. 2010, IEEE Transactions on Industrial Electronics, vol. 57, no. 5, pp. 1710-1718.
8. E. Rashedi, H. Nezamabadi-pour, S. Saryazdi, GSA: a gravitational sarch algorithm, Jun. 2009, Information Sciences, vol. 179, no. 13, pp. 2232-2248.
9. X. S. Yang, S. Deb, Cuckoo search via L´evy flights, Dec. 2009, World Congress on Nature & Biologically Inspired Computing, pp. 210-214.
10. E. Rashedi, H. Nezamabadi-pour, S. Saryazdi, Filter modeling using gravitational search algorithm, Feb. 2011,Engineering Applications of Artificial Intelligence, vol. 24, no. 1, pp. 117-122.
11. A. P. Patwardhan, R. Patidar, N. V. George, On a cuckoo search optimization approach towards feedback system identification, Sep. 2014, Digital Signal Processing, vol. 32, pp. 156-163.

12. J. Kennedy, R. Eberhart, Particle swarm optimization, Nov. 1995,IEEE International Conference on Neural Networks, pp. 1942-1948.
13. D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, Nov. 2007, Journal of Global Optimization, vol. 39, no. 3, pp. 459-471.
14. X. S. Yang, Flower pollination algorithm for global optimization, Lecture Notes in Computer Science, vol. 7445, pp. 240-249, Sep. 2012.
15. R. V. Rao, V. J. Savsani, D. P. Vakharia, Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems, Mar. 2011, Computer-Aided Design, vol. 43, no. 3, pp. 303-315.
16. R. Storn, K. Price, Differential evolution-a simple and efficient heuristic strategy for global optimization over continuous spaces, Dec. 1997,Journal of Global Optimization, vol. 11, no. 4, pp. 341-359.
17. P. Upadhyay, R. Kar, D. Mandal, S. P. Ghoshal, Craziness based particle swarm optimization algorithm for IIR system identification problem, May. 2014, AEU-International Journal of Electronics and Communications, vol. 68, no. 5, pp. 369-378.
18. E. Cuevas, J. G´alvez, S. Hinojosa, O. Avalos, D. Zald´lvar, M. P´erez-Cisneros, A comparison of evolutionary computation techniques for IIR model identification, Journal of Applied Mathematics, http://dx.doi.org/10.1155/2014/827206, 2014.
19. R. Singh, H. K. Verma, Teaching-learning-based optimization algorithm for parameter identification in the design of IIR filters, Dec. 2013, Journal of The Institution of Engineers (India): Series B, vol. 94, no. 4, pp. 285-294.
20. D. H. Wolpert, W. G. Macready, No free lunch theorems for optimization, Apr. 1997, IEEE Transactions on Evolutionary Computation, vol. 1, no. 1,pp. 67-82.
21. Q. Y. Jiang, L.Wang, X. H. Hei, et al, Optimal approximation of stable linear systems with a novel and efficient optimization algorithm, Jul. 2014 ,IEEE Congress on Evolutionary Computation, pp. 840-844.

22. Q. Y. Jiang, L. Wang, X. H. Hei, Parameter identification of chaotic systems using artificial raindrop algorithm, May. 2015, Journal of Computational Science, vol. 8, pp. 20-31.
23. Q. Y. Jiang, L. Wang, X. H. Hei, et al., The performance comparison of a new version of artificial raindrop algorithm on global numerical optimization, Oct. 2015, Neurocomputing, doi: 10.1016/j.neucom.2015.09.093.
24. S. K. Saha, R. Kar, D. Mandal, S. P. Ghoshal, Optimal IIR filter design using novel particle swarm optimization technique, Sep. 2012, International Journal of Circuits, Systems and Signal Processing, vol. 6, no. 2, pp. 151-162.
25. N. Karaboga, A. Kalinli, D. Karaboga, Designing digital IIR filters using ant colony optimisation algorithm, Apr. 2004, Engineering Applications of Artificial Intelligence, vol. 17, no. 3, pp. 301-309.
26. A. Kalinli, N. Karaboga, Artificial immune algorithm for IIR filter design, Dec. 2005, Engineering Applications of Artificial Intelligence, vol. 18, no. 8, pp. 919-929.
27. Y. Wang, Z. X. Cai, Q. F. Zhang, Differential evolution with composite trial vector generation strategies and control parameters, Feb. 2011, IEEE Transactions on Evolutionary Computation, vol. 15, no. 1, pp. 55-66.
28. J. J. Liang, A. K. Qin, P. N. Suganthan, S. Baskar, Comprehensive learning particle swarm optimizer for global optimization of multimodal functions, Jun. 2006, IEEE Transactions on Evolutionary Computation, vol. 10, no. 3, pp. 281-295.
29. M. E. H. Pedersen, Good parameters for differential evolution, Technical Report HL1002, Hvass Laboratories.
30. F. Zou, L.Wang, X. H. Hei, D. B. Chen, D. D. Yang, Teaching-learning-based optimization with dynamic group strategy for global optimization, Jul. 2014, Information Sciences, vol. 273, no. 20, pp. 112-131, Jul. 2014.
31. S. Mirjalili, A. Lewis, Adaptive gbest-guided gravitational search algorithm, Dec. 2014, Neural Computing and Applications, vol. 25, no. 7-8, pp. 1569-1584.
32. A. Sadollah, D. G. Yoo, J. H. Kim, Improved mine blast algorithm for optimal cost design of water distribution systems, Nov. 2014, Engineering Optimization.

33. J. Zhang, A. C. Sanderson, JADE: adaptive differential evolution with optional external archive, Oct. 2009, IEEE Transactions on Evolutionary Computation, vol. 13, no. 5, pp. 945-958.
34. W. F. Gao, S. Y. Liu, A modified artificial bee colony algorithm, Mar. 2012, Computers and Operations Research, vol. 39, no. 3, pp. 687-697.
35. S. T. Ahmed, A. B. Amr, F. A. R. Ibrahim, One rank cuckoo search algorithm with application to algorithmic trading systems optimization, Feb. 2013, International Journal of Computer Applications, vol. 4, no. 6, pp. 30-37.
36. J. Sun, W. B. Xu, B. Feng, A global search strategy of quantum-behaved particle swarm optimization, Dec. 2004, IEEE Conference on Cybernetics and Intelligent Systems, pp. 111-116.
37. R. Cheng, Y. C. Jin, A social learning particle swarm optimization algorithm for scalable optimization, Jan. 2015, Information Sciences, vol. 291, no. 10, pp. 43-60.