Chapter 8
Experimental Study on Improved Differential Evolution for System Identification of Hammerstein Model and Wiener Model

Weili Xiong, Minfang Chen, Le Yao and Baoguo Xu

Abstract For nonlinear system of the Hammerstein model and Wiener model, a method for nonlinear system identification is proposed based on differential evolution Algorithm (DE). The based idea of the method is that the problem of nonlinear system identification is changed into optimization problems in parameter space. In order to enhance the performance of the DE identification, put forward a kind of adaptive mutation differential evolution algorithm for scaling factor (MDE), and on this basis, we make an improvement on crossover to make a better performance. To make an analysis for particle swarm optimization (PSO), DE and improved DE, the improvement DE has higher accurate and recognition ability, stronger convergence.

Keywords DE · Hammerstein model · Winner model · Improvement

8.1 Introduction

For the linear system identification, theoretical studies have tended to mature, but in real life, actual system is almost nonlinear system, so the research for nonlinear system is necessary [1]. The Wiener model and Hammerstein model [2] which Narendra and Gallman proposed not only simple but also can effectively describe the nonlinear characteristics of dynamic system. Literature [3] based on PSO has a identification for Hammerstein model, although the algorithm has a good

W. Xiong (✉) · M. Chen · L. Yao · B. Xu
Key Laboratory of Advanced Process Control for Light Industry (Ministry of Education), Automation Institute of Jiangnan University, Wuxi, Jiangsu, China
e-mail: greenpre@163.com

W. Xiong · M. Chen · L. Yao · B. Xu
Room C421 in School of Internet of Things Engineering, Lihu Campus in Jiangnan University, Wuxi, Jiangsu Province

Z. Sun and Z. Deng (eds.), *Proceedings of 2013 Chinese Intelligent Automation Conference*, Lecture Notes in Electrical Engineering 254,
DOI: 10.1007/978-3-642-38524-7_8,

robustness, the running speed of the program is relatively slow; Literature [4] based on PSO has a identification for Hammerstein model, QPSO algorithm has more strong nonlinear recognition ability, the program running time is also relatively fast, however, to some extent, identification of the complexity of the operation increased.

Differential evolution (DE) algorithm was first introduced by Storn and Price for global optimization in 1995 [5], as a stochastic global optimizer, DE has appeared as a simple and very efficient optimization technique. The DE's advantages are easy to implement, require a few control parameters turning and exhibit fast convergence. The DE algorithm is a population-based algorithm using the following operators: crossover, mutation and selection. In recent years, DE is widely applied in neural network [6], parameter identification [7], function optimization [8, 9] and constraint optimization problem [10, 11] and other areas. However, it has been observed that the convergence rate of DE do not meet the expectation in cases of highly multimodal problems. Several variants of DE have been proposed to improve its performance.

8.2 Differential Evolution

In this section we will describe briefly the working of basic DE. Compared with other evolutionary algorithms (EA), DE is a simple yet powerful optimizer with fewer parameters [12]. Scale factor (F) and crossover rate (CR) are two very important control parameters of DE. Setting the parameters to inappropriate values may not only deteriorate the search efficiency, but also lead to solutions with poor quality. Differential evolution's basic steps can be described as follows [13]:

Mutation operation: The mutation operation of DE applies the vector differentials between the existing population members for determining both the degree and direction of perturbation applied to the individual subject of the mutation operation. The mutation process at each generation begins by randomly selecting three individuals $X_{r1,G}$, $X_{r2,G}$ and $X_{r3,G}$, in the population set of NP elements. The ith perturbed individual, $V_{i,G+1}$, is generated based on the three chosen individuals as follows:

$$V_{i,G+1} = X_{r3,G} + F * \left(X_{r1,G} - X_{r2,G}\right) \tag{8.1}$$

where, $i = 1 \cdots NP$, $r_1, r_2, r_3 \in \{1 \cdots NP\}$ are randomly selected such that $r_1 \neq r_2 \neq r_3 \neq i$, F is the control parameter such that $F \in [0, +1]$.

Crossover operation: once the mutant vector is generated, the perturbed individual, $V_{i,G+1} = \left(V_{1,i,G+1}, \ldots, V_{n,i,G+1}\right)$, and the current population member, $X_{i,G} = \left(X_{1,i,G}, \ldots X_{n,i,G}\right)$, are then subject to the crossover operation, that finally generates the population of candidates, or "trial" vectors, $U_{i,G+1} = \left(u_{1,i,G+1}, \ldots, u_{n,i,G+1}\right)$, as follows:

$$u_{j,i,G+1} = \begin{cases} v_{j,i,G+1} & if\ rand_j \leq C_r \vee j = k \\ x_{j,i,G} & otherwise \end{cases} \tag{8.2}$$

where, $j = 1 \cdots n$, $k \in \{1 \cdots n\}$ is a random parameter's index, chosen once for each i. The crossover rate, $C_r \in [0, 1]$, is set by the user.

Selection operation: If the new individual is better than the original one then the new individual is to be an offspring n the next generation $G = t + 1$ else the new individual is discarded and the original one is retained in the next generation.

$$X_{i,G+1} = \begin{cases} U_{i,G+1} & if f\left(U_{i,G+1}\right) \leq f\left(X_{i,G}\right) \\ X_{i,G} & otherwise \end{cases} \tag{8.3}$$

where, $f()$ is the fitness function. Each individual of the temporary population is compared with its counterpart in the current population. The one with the lower projective function value will survive generation.

8.2.1 Improvement of DE

To make DE more efficient for different scenarios, efforts are needed to improve its performance and a chaotic algorithm, simulated annealing algorithm or some adaptive methods have been applied in DE.

A. In the differential evolution algorithm, the constant of differentiation F is a scaling factor of the difference vector. It is an important parameter that controls the evolving rate of the population. One of the most useful differential evolution strategies are described in the following.

It is proposed a adaptive scaling factor formula [14]:

$$M = F_0 * 2^{\lambda} \quad \lambda = e^{\left(1 - \frac{T}{T+1-t}\right)} \tag{8.4}$$

where F_0 is the initial scaling factor, T is the number of iterations and t is the current evolution number. Early in the algorithm, $\lambda = 1$, make the $M = 2F_0$, the improvement called MDE.

Early in the algorithm, adaptive mutation operator was $F_0 - 2F_0$, it has great value and makes the individuals diversity in the population at the initial generations to overcome the premature. With the algorithm progress, mutation operator gradually reduced, mutation rates is closed to F_0 later, preserve the excellent individuals, enhance the probability of obtaining the global optimum.

B. As mentioned in the above subsection, *CR* is a very important control parameters of DE. At early stage, the diversity of population is large because the population individuals are all different from each other. Small values of *CR* increase the possibility of stagnation and slow down the search process. On the

other hand, if the *CR* value is relatively high, this will increase the population diversity and improve the convergence. Therefore, the *CR* must take a suitable value in order to avoid the premature convergence or slow convergence rate. Based on the analysis above, in order to balance between the diversity and the convergence rate, a dynamic nonlinear increased crossover probability method was proposed, which is called KDE, formula is as follows [15]:

$$CR = CR_{\max} + (CR_{\min} - CR_{\max}) * (1 - count/gen\_\max)^k \tag{8.5}$$

Which *count* is the current generation number, *gen*_max is the maximum number of generations, $CR_{\min}$ and $CR_{\max}$ denote the maximum and minimum value of the *CR*, and k is a positive number. The optimal setting for these parameters are $CR_{\min} = 0.5$, $CR_{\max} = 0.55$ and $k = 4$.

In this paper, a new method is put forward. Combining the two kinds of methods include KDE and MDE, which have mentioned above, trying to combine MDE and KDE, make use of MDE can increase the global search of the optimal value of the probability and KDE has a good performance in accuracy.

8.2.2 Hammerstein Model and Wiener Model

A. Hammerstein model is a special kind of nonlinear system, the structure series composition by a no memory nonlinear gain link and a dynamic linear link, as Fig 8.1 show.

The difference equation of Hammerstein model expressed as:

$$\begin{cases} A(q^{-1})y(k) = B(q^{-1})x(k) + C(q^{-1})w(k) \\ x(k) = f(u(k)) = u(k) + r_2u^2(k) + \cdots + r_pu^p(k) \\ A(q^{-1}) = 1 + a_1q^{-1} + \cdots + a_nq^{-n} \\ B(q^{-1}) = b_0 + b_1q^{-1} + \cdots + b_tq^{-t} \\ C(q^{-1}) = 1 + c_1q^{-1} + \cdots + c_mq^{-m} \end{cases} \tag{8.6}$$

Among them $u(k)$and $y(k)$ are identification system input and output sequence; $w(k)$ is the Gaussian white noise sequence with zero-mean and variance of σ^2, $u(k)$and $w(k)$ independent; $x(k)$ is unmeasured intermediate input signal, it is not only linear dynamic input but also the output of the nonlinear part. q^{-1} for lag

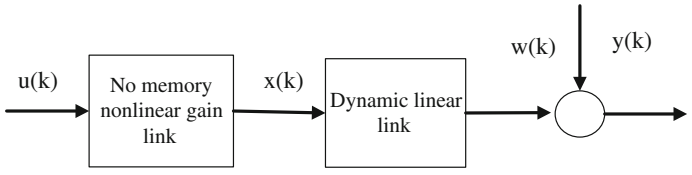

Fig. 8.1 Hammerstein model

operator, $A(q^{-1})B(q^{-1})$ and $C(q^{-1})$ are lagging operator polynomial; $f(\cdot)$ is no memory for the nonlinear gain. Introducing parameter vector for $\theta_1 = [a_1\ a_2 \cdots a_n\ b_1\ b_2 \cdots b_t\ c_1\ c_2 \cdots c_m\ r_1\ r_2 \cdots r_p]^T$, identify target is based on a given input $u(k)$ and system output $y(k)$ estimate parameter vector θ_1, set the estimate of parameter vector θ_1, $\hat{\theta}_1 = \left[\hat{a}_1\ \hat{a}_2 \cdots \hat{a}_n\ \hat{b}_1\ \hat{b}_2 \cdots \hat{b}_t\ \hat{c}_1\ \hat{c}_2 \cdots \hat{c}_m\ \hat{r}_1\ \hat{r}_2 \cdots \hat{r}_p\right]^T$.

B. Wiener model is a special kind of nonlinear system, the structure series composition by a no memory nonlinear gain link and a dynamic linear link, as Fig 8.2 shows.

The difference equation of Wiener model expressed as:

$$\begin{cases} A(q^{-1})z(k) = q^{-d}B(q^{-1})u(k) \\ y(k) = f[z(k)] + e(k) \\ A(q^{-1}) = 1 + a_1 q^{-1} + \cdots + a_n q^{-n} \\ B(q^{-1}) = b_0 + b_1 q^{-1} + \cdots + b_t q^{-t} \end{cases} \tag{8.7}$$

Among them $u(k)$ and $y(k)$ are identification system input and output sequence, $e(k)$ is white Gaussian noise, $z(k)$ is the linear part of the output. Definition parameter variables $\theta_2 = [a_1 \cdots a_n\ b_0 \cdots b_t]^T$, identify target is based on a given input $u(k)$ and system output $y(k)$ estimate parameter vector θ_2, set the estimate of parameter vector θ_2, $\hat{\theta}_2 = [\hat{a}_1 \cdots \hat{a}_n\ \hat{b}_0 \cdots \hat{b}_t]^T$.

The estimate of the deviation can use the following criterion function to measure.

$$J(k) = \sum_{i=0}^{L} \left[y(k-i) - \hat{y}(k-i) \right]^2 \tag{8.8}$$

The L for identification window length, $y(k-i)$ $\hat{y}(k-i)$ is $k-i(i = 1, \ldots L)$ moment output measurement signal and estimate.

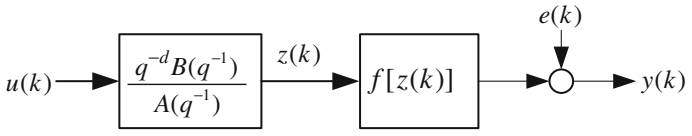

Fig. 8.2 Wiener model

8.3 Experiment Study

8.3.1 Test Function Used in Simulation Studies

In this paper, aim at the problem of Hammerstein model, using PSO, DE and improved DE do some simulation. For PSO, study gene $c_1 = 2, c_2 = 1.6$, inertia weight w linear decreases along with the iteration times 0.4 from 0.9, maximum speed limit in 1; For DE, the scale factor $F = 0.5$ and the crossover probability $CR = 0.3$; for MDE, $F_0 = 0.1$, and $CR = 0.4$; for MKDE, $F_0 = 0.3$, $CR_{\min} = 0.3$, $CR_{\max} = 0.55$. population size (NP) between $5D$ and $10D$, D is the number of the goal function decision variables, no less than 4, this lab take $NP = 40$, the maximum iterations are for 1,200. Each differential evolution algorithm based on each improvement is run 20 times, the results are presented in Tables 8.1, 8.2, where various standard statistical measures including mean, minimum and RMSE. The Hammerstein model select is as follows:

$$\begin{cases} A(q^{-1})y(k) = B(q^{-1})x(k) + C(q^{-1})w(k) \\ x(k) = f(u(k)) = u(k) + 0.5u^2(k) + 0.3u^3(k) + 0.1u^4(k) \\ A(q^{-1}) = 1 - 1.5q^{-1} + 0.7q^{-2} \\ B(q^{-}) = q^{-1} + 0.5q^{-2} \\ C(q^{-1}) = 1 + 1.5q^{-1} \end{cases} \tag{8.9}$$

Wiener model select is as follows:

$$\begin{cases} x(k) = 15x(k-1) - 07x(k-2) + u(k-1) + 0.5u(k-2) \\ y(k) = f[x(k)] + e(k), \\ f[x(k)] = \begin{cases} \sqrt{x(k)/2}, x(k) \geq 0, \\ -\sqrt{-x(k)/2}, x(k) > 0 \end{cases} \end{cases} \tag{8.10}$$

Table 8.1 The numerical comparison of four kinds of algorithm for Hammerstein model

Parameter	Truth-value	Estimate-value			
		PSO	DE	MDE	MKDE
a_1	−1.5	−1.5001	−1.5000	−1.5000	−1.5000
a_2	0.7	0.7000	0.7000	0.7000	0.7000
b_1	1.0	1.0430	1.0000	0.9997	1.0000
b_2	0.5	0.4998	0.5000	0.5000	0.5000
c_1	1.5	1.4566	1.5000	1.5003	1.5000
r_1	0.5	0.4855	0.5000	0.4999	0.5000
r_2	0.3	0.3342	0.3001	0.3021	0.3001
r_3	0.1	0.0483	0.1005	0.1319	0.1005
RMSE		5.3506e-004	3.7822e-004	2.5512e-008	2.5512e-008

Table 8.2 The numerical comparison of four kinds of algorithm for Wiener model

Parameter	Truth-value	Estimate-value			
		PSO	DE	MDE	MKDE
a_1	−1.5	−1.4998	−1.5000	−1.5000	−1.5000
a_2	0.7	0.6998	0.7000	0.7000	0.7000
b_0	1.0	1.0000	1.0000	1.0000	1.0000
b_1	0.5	0.4999	0.5000	0.5000	0.5000
RMSE		2.47e-008	1.32e-010	1.32e-012	4.81e-015

The input signal $u(k)$ is zero-mean, variance is 1 of the Gaussian white noise sequence, $w(k)$ is Gaussian white noise which variance is 0.1, identify window width is 500, to identify the parameters of the true value vector is, $\theta_1 = [-1.5\ 0.7\ 1.0\ 0.5\ 1.5\ 0.5\ 0.3\ 0.1]^{\mathrm{T}}$, $\theta_2 = [-1.5\ 0.7\ 1.0\ 0.5]^{\mathrm{T}}$. In simulation experiments, define root mean square error (RMSE) to measure precision.

$$RMSE = \sqrt{\frac{\sum_{j=1}^{L}\left[y(j) - \hat{y}(j)\right]^2}{L}} \tag{8.11}$$

8.3.2 Experimental Results

The experiment works on MATLAB7.1 software simulation platform. For Hammerstein model, There are 8 parameters ($a_1\ a_2\ b_1\ b_2\ c_1\ r_1\ r_2\ r_3$), this paper drawing results put five linear parameters ($a_1\ a_2\ b_1\ b_2\ c_1$) and three nonlinear parameter ($r_1\ r_2\ r_3$) separate display. Figures 8.3, 8.4, 8.5, 8.6 are respectively the parameter identification results of Hammerstein model system with PSO algorithm DE algorithm and improved DE algorithm, Figs. 8.7, 8.8, 8.9, 8.10 are the results of Wiener model.

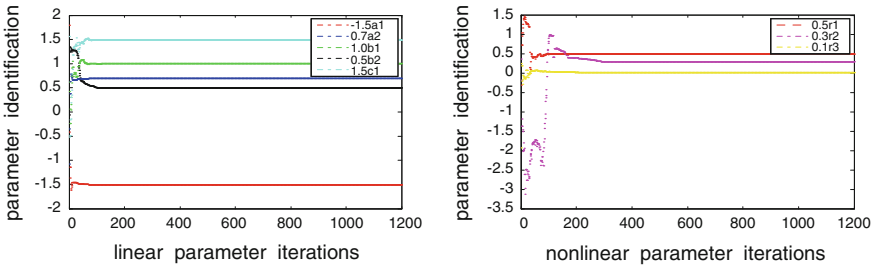

Fig. 8.3 Performance curves of PSO algorithm for Hammerstein model

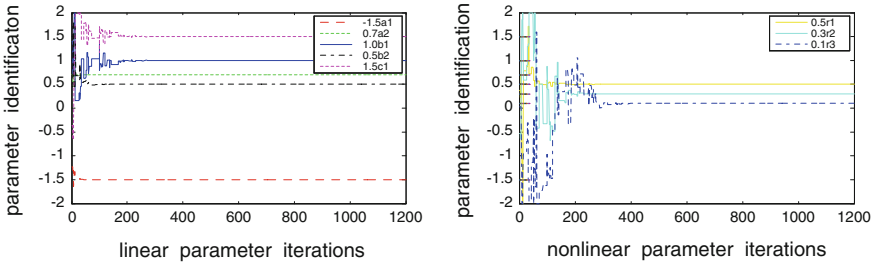

Fig. 8.4 Performance curves of DE algorithm for Hammerstein model

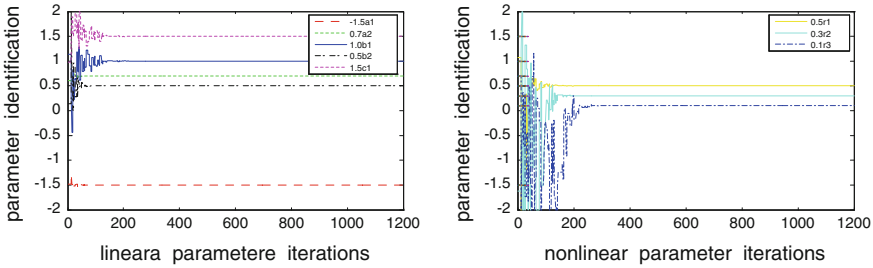

Fig. 8.5 Performance curves of MDE algorithm for Hammerstein model

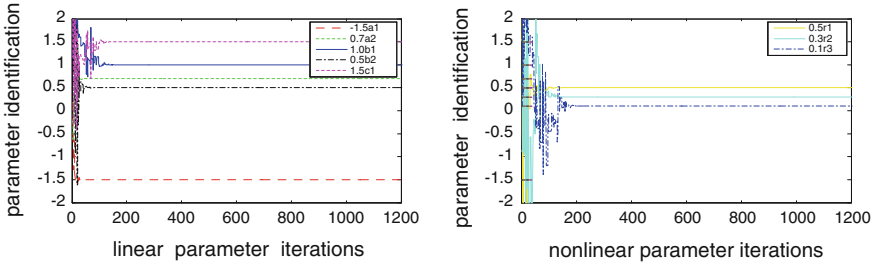

Fig. 8.6 Performance curves of MKDE algorithm for Hammerstein model

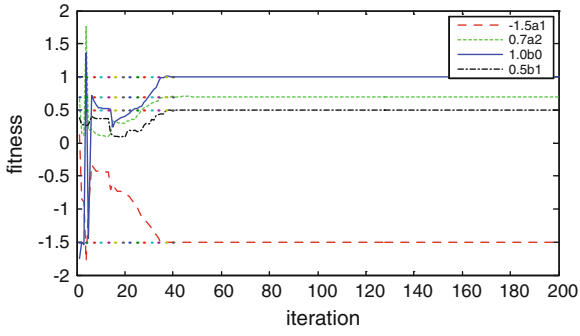

Fig. 8.7 Performance curves of PSO algorithm for Wiener model

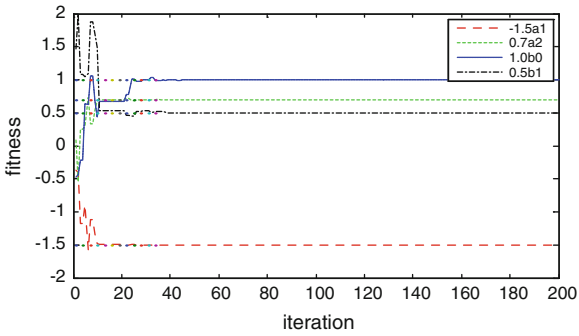

Fig. 8.8 Performance curves of DE algorithm for Wiener model

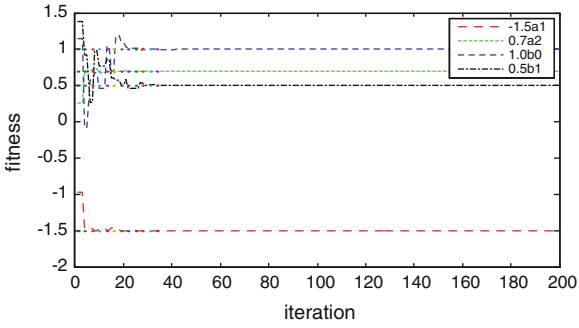

Fig. 8.9 Performance curves of MDE algorithm for Wiener model

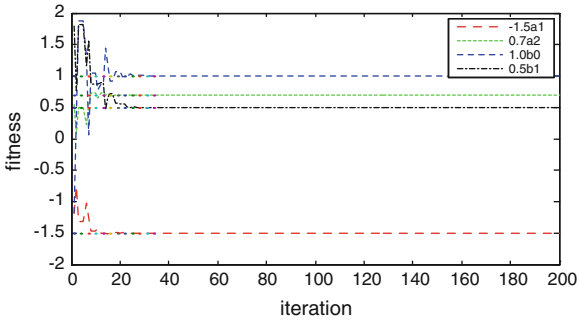

Fig. 8.10 Performance curves of MKDE algorithm for Wiener model

Figure 8.3 shows that PSO algorithm identifies Hammerstein model parameters with less iteration, the whole curve relatively smooth, identify the linear parameter within 100 generation, identify the nonlinear parameter in about 300 generation, but its program running time is many times as other algorithm.

Figure 8.4 shows that DE algorithm identify Hammerstein model parameters a little more iteration than PSO algorithm, identify the linear parameter around 200 generation, also identify the nonlinear parameter in about 340 generation, but its running faster and has more accurate identification parameter results.

We can see from Fig. 8.5 that MDE algorithm not only identify Hammerstein model parameters with less iteration, identify the linear parameter around 150 generation, identify the nonlinear parameter around 250 generation, but also its running faster with the most accurate result, and has the maximum probability of getting the global optimal solution.

It is shown in Fig. 8.6 that MKDE algorithm identifies the linear parameters with only about 100 iterations, identify the nonlinear parameters with about 200 iterations. Compare this method with the methods mentioned above, we know that the convergence rate of the MKDE is much fast than the other algorithms.

From Figs. 8.7, 8.8, 8.9, 8.10, we also can see that it is evident that the convergence of DE algorithm is faster than PSO algorithm. Figure 8.7 shows the identification iterations of PSO is about 36 generations and the oscillating amplitude is big before it achieves stability; Fig. 8.8 illustrates that DE algorithm can identify the parameters of Wiener model with 30 iterations, and the oscillating amplitude is smaller than the PSO algorithm; Fig. 8.9 shows that MDE algorithm identify the parameters of Wiener model in about 25 iterations; Fig. 8.10 shows that MKDE algorithm identify the parameters of Wiener model with only about 20 iterations.

In the same parameter setting conditions, we get the data result when the graph output. Take the average of the 30 group data, the results shown in Tables 8.1 and 8.2.

From Tables 8.1 and 8.2, it can be clearly observed that the DE algorithm and the improved DE algorithms perform more or less in a similar result although MKDE outperforms other algorithms. Although MDE can increase global search ability, and MKDE has a better performance both in accuracy and convergence. From Tables 8.1 and 8.2, we can see that the accuracy of PSO algorithm is smaller than the DE and the improved DE. PSO algorithm identifies the parameters is closed to the truth-value, however, the DE algorithm and improved DE can identify the parameters perfectly with little error, especially the MKDE algorithm.

8.4 Conclusions

In this paper we have investigated the performance of DE and the improved DE, the simulation of results showed that relative to the basic DE algorithm, improved DE algorithm some more advanced, the MKDE not only do better than other algorithms in accuracy, but also performs excellent at convergence speed. It turns out that the performance of DE algorithm is very sensitive to the choice of parameters and is related with the feature of problem. The next work we should do is to improve the stability of the DE algorithm.

Acknowledgments This work was supported by the National Natural Science Foundation of China (21206053,21276111); General Financial Grant from China Postdoctoral Science Foundation (2012M511678); A Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions.

References

1. Fang F, Reed E, Dickason DK, Simien HJ et al (2005) Technology review of multi-agent systems and tools. Bureau of Naval Personnel, Washington
2. Natrnfts K, Gallman P (1996) An iterative method for the identification of nonlinear systems using a Hammerstein model. IEEE Trans Autom Control 11(3):546–550
3. Lin X, Zhang H, Liu S et al (2006) The Hammerstein model identification based on PSO. Chin J Sci Instrum 27(1):76–79 (in Chinese)
4. Shen J, Sun J, Xu W (2009) System identification based on QPSO algorithm. Comput Eng Appl 45(9):67–70
5. Storn R, Price K (1997) Differential evaluation: a simple and efficient adaptive scheme for global optimization over continuous spaces. Global Optim 11:341–359
6. Ming Z, Guoxun W, Yipu Y (2010) Application of neural network and differential evolution algorithm in adaptive filterin. Process Automation Instrum 31(4):8–11 (in Chinese)
7. Chang W (2006) Parameter identification of Rossler's chaotic system by an evolutionary algorithm. Science 29(5):1047–1053 (in Chinese)
8. Xu Q, Wang L, He B et al (2011) Opposition-based differential evolution using the current optimum for function optimization. J Appl Sci Electron Inf Eng 29(3):309–314 (in Chinese)
9. Tang HS, Xue ST, Fan CX (2008) Differential evolution strategy for structural system identification. Comput Struct 86(21–22):2004–2012
10. Liu R, Jiao L, Lei Q et al (2011) New differential evolution constrained optimization algorithm. J Xidian Univ (Nat Sci Ed) 38(1):47–52 (in Chinese)
11. Qian WY, Ii AJ (2008) Adaptive differential evolution algorithm for multi-objective optimization problems. Appl Math Comput 201(1):431–440
12. Zhang D, Liu Y, Huang S (2012) Differential evolution based parameter identification of static and dynamic J-A models and Its application to inrush current study in power converters [J]. IEEE Trans Magnet 48(11):3482–3485
13. Yan X, Yu J, Qian F et al (2006) Kinetic parameter estimation of oxidatical water based on modified differential evolution. J East China Univ Sci Technol (Nat Sci Ed) 32(1):94–97 (in Chinese)
14. Mohamed AW, Sabry HZ (2012) Constrained optimization based on modified differential evolution algorithm. Inf Sci 194:171–208