

7-2-2020

Identification of Wiener Model Based on Improved Differential Evolution (SADE) Algorithm

Xiaoping Xu

1. *School of sciences, Xi'an University of Technology, Xi'an 710054;;*

Bai Bo

1. *School of sciences, Xi'an University of Technology, Xi'an 710054;;*

Fucai Qian

2. *School of Automation and information Engineering, Xi'an University of Technology, Xi'an 710048;*

Follow this and additional works at: <https://dc-china-simulation.researchcommons.org/journal>



Part of the Artificial Intelligence and Robotics Commons, Computer Engineering Commons, Numerical Analysis and Scientific Computing Commons, Operations Research, Systems Engineering and Industrial Engineering Commons, and the Systems Science Commons

This Paper is brought to you for free and open access by Journal of System Simulation. It has been accepted for inclusion in Journal of System Simulation by an authorized editor of Journal of System Simulation.

Identification of Wiener Model Based on Improved Differential Evolution (SADE) Algorithm

Abstract

Abstract: To identify the parameters of nonlinear Wiener model, a new identification method was put forward based on a SADE algorithm. Its basic idea is as follows: *the Sigmoid function and adaptive mutation operator were adopted to improve the mutation operation part of the basic differential evolution algorithm, accordingly, the disadvantages of the basic differential evolution algorithm, such as premature convergence, instability, etc, were effectively overcome. The proposed algorithm was used to parameter identification problem of Wiener model; moreover, the accuracy of identification was well improved.* In the numerical simulation, compared with other relevant existing algorithms, simulation results show that the proposed method is rational and effective.

Keywords

differential evolution algorithm, Sigmoid function, adaptive operator, SADE algorithm, Wiener model, identification

Recommended Citation

Xu Xiaoping, Bai Bo, Qian Fucai. Identification of Wiener Model Based on Improved Differential Evolution (SADE) Algorithm[J]. Journal of System Simulation, 2016, 28(1): 147-153.

基于改进差分进化算法的 Wiener 模型辨识

徐小平¹, 白博¹, 钱富才²

(1.西安理工大学 理学院, 西安 710054; 2.西安理工大学 自动化与信息工程学院, 西安 710048)

摘要: 针对非线性 Wiener 模型的参数辨识问题, 提出了一种基于 Sigmoid 函数及自适应算子改进差分进化(improved differential evolution algorithm with Sigmoid function and adaptive mutation operator, SADE)算法的参数辨识方法。利用 Sigmoid 函数及自适应变异算子改进了基本差分进化算法的变异操作部分, 改进的方法能够有效地克服基本差分进化算法的过早收敛和不稳定性等缺点。将该改进差分进化算法应用于对非线性 Wiener 模型的参数辨识问题, 达到了较高的辨识精度。在仿真试验中, 与其它已有方法进行比较, 仿真结果说明了所给的参数辨识方法是合理和有效的。

关键词: 差分进化算法; Sigmoid 函数; 自适应算子; SADE 算法; Wiener 模型; 辨识

中图分类号: TP391.9 文献标识码: A 文章编号: 1004-731X (2016) 01-0147-07

Identification of Wiener Model Based on Improved Differential Evolution (SADE) Algorithm

Xu Xiaoping¹, Bai Bo¹, Qian Fucal²

(1. School of sciences, Xi'an University of Technology, Xi'an 710054;

2. School of Automation and information Engineering, Xi'an University of Technology, Xi'an 710048)

Abstract: To identify the parameters of nonlinear Wiener model, a new identification method was put forward based on a SADE algorithm. Its basic idea is as follows: *the Sigmoid function and adaptive mutation operator were adopted to improve the mutation operation part of the basic differential evolution algorithm, accordingly, the disadvantages of the basic differential evolution algorithm, such as premature convergence, instability, etc, were effectively overcome. The proposed algorithm was used to parameter identification problem of Wiener model; moreover, the accuracy of identification was well improved. In the numerical simulation, compared with other relevant existing algorithms, simulation results show that the proposed method is rational and effective.*

Keywords: differential evolution algorithm; Sigmoid function; adaptive operator; SADE algorithm; Wiener model; identification

引言

系统辨识是对输入和输出观测数据的基础上, 在一类系统中确定一个与被辨识系统等价的系统。对于线性系统的辨识研究已经趋于成熟, 但是, 在

现实中, 实际系统几乎都是非线性系统, 所以对于非线性系统的研究就十分必要了^[1-3]。

目前, Wiener 模型作为一种典型的非线性系统, 已被广泛地应用于工业生产过程中的, 如 pH 中和过程, 热工系统和流体控制等领域^[4-5], 因而, 对该模型的辨识研究是当前非线性系统辨识领域的一个重要研究课题。

Wiener 模型是由一线性子系统和一无记忆非线性增益串联而成。对 Wiener 模型的辨识, 文献



收稿日期: 2014-07-29 修回日期: 2015-01-27;
基金项目: 国家自然科学基金(61273127); 陕西省教育厅专项科研计划项目(14JK1538); 陕西省自然科学基金基础研究计划项目(2014JM8325);
作者简介: 徐小平(1973-), 男, 博士, 副教授, 研究方向为进化算法, 系统建模理论等。

[6]提出了应用不同幅值的三电平伪随机 m 序列作为辨识输入信号来对该模型进行辨识,该方法对非线性增益系数的辨识比较成功,但是,无法获得线性环节的参数模型;文献[7]利用多项式逼近的建模方法来辨识 Wiener 模型,但多项式阶次的设置往往会影响到辨识精度,且该算法计算量太大;文献[8]提出了利用神经网络来辨识 Wiener 模型,但神经网络的拓扑结构往往影响到辨识结果,而对其的确定也较为困难;文献[9]采用遗传算法逼近非线性增益的逆函数,利用最小二乘法辨识线性子系统的参数,但确定神经网络结构带有一定的随机性,辨识精度不高,计算量和复杂度太大;文献[10]采用粒子群算法来辨识 Wiener 模型,但扰动对该算法的性能以及辨识的结果有一定程度的影响,导致精度不高。

差分进化(differential evolution, DE)算法是一种新兴的进化计算技术,目前已得到广泛地应用^[11-14]。本文尝试应用 DE 算法辨识 Wiener 模型。首先,分析了基本 DE 算法的优缺点。然后,提出了 SADE 算法。接着,利用该算法辨识 Wiener 模型。最后,实验说明了所给方法是有效的。

1 基本 DE 算法

基本 DE 算法是一种基于群体进化的算法,具有记忆个体最优解和种群内信息共享的特点,通过种群内个体之间的合作与竞争来求解优化问题^[14-16]。其基本原理是:首先,随机产生初始种群。然后,通过变异操作作为当前种群中的每一个个体生成一个目标个体,再通过交叉操作将当前部分变量用目标个体的对应变量替换,从而生成测试个体。最后,采用贪婪选择的方式,在测试个体和当前个体中选择较好的个体进入下一代搜索,这样就保证了下一代种群中的个体至少不比当前种群个体差,从而使种群的平均性能提高,并达到最优解。

考虑下面函数最小化问题:

$$\begin{aligned} \min f(x), \quad x &= [x_1, x_2, \dots, x_d] \\ x_i^L &\leq x_i \leq x_i^U, \quad i = 1, 2, \dots, d \end{aligned}$$

其中: x_i^U 和 x_i^L 分别表示第 i 维变量的搜索上界和下界; d 为问题维数。基本的 DE 算法的步骤如下。

(1) 初始化种群: 随机产生 NP 个个体来构成初始种群, 如(1)式所示。

$$\begin{aligned} x_{k,i}(0) &= x_i^L + \text{rand}() \times (x_i^U - x_i^L) \\ i &= 1, 2, \dots, d, \quad k = 1, 2, \dots, NP \end{aligned} \quad (1)$$

其中, $\text{rand}()$ 表示 $[0, 1]$ 之间均匀分布的随机数。

(2) 变异操作: 在每一代搜索中, 基本 DE 算法通过变异操作作为当前种群中的每一个个体 $x_j(g)$ 生成一个目标个体 $t_j(g)$, 其中, g 表示进化代数。目前, 基本 DE 算法有许多种不同版本的变异算子, 可表示为 $\text{DE}/a/b$ 的形式。其中, a 表示变异操作基的类型, 一般取 rand 和 best 两个值, rand 表示随机选择个体作为变异算子的基, best 表示选择当前最优个体作为变异算子的基; b 表示变异时差分项的数目。典型的变异算子有以下 2 种。

DE/rand/1 变异算子:

$$t_i(g) = x_{r_1}(g) + F \cdot [x_{r_2}(g) - x_{r_3}(g)] \quad (2)$$

DE/rand/2 变异算子:

$$\begin{aligned} t_i(g) &= x_{\text{best}}(g) + F \cdot [x_{r_2}(g) - x_{r_3}(g)] + \\ &F \cdot [x_{r_4}(g) - x_{r_5}(g)] \end{aligned} \quad (3)$$

其中, $r_1, r_2, r_3, r_4, r_5 \in \{1, 2, \dots, N\}$ 为 $\neq i$ 的互不相同的整数, $x_{\text{best}}(g)$ 表示第 g 代种群中的最好个体, F 为取值位于 $0 \sim 1$ 的缩放因子。

(3) 交叉操作: 基本 DE 算法的交叉操作在变异操作后进行。通过将当前个体 x_i 的部分变量用目标个体 t_i 的对应变量替换, 从而生成测试个体 v_i 来保留个体中较优良的变量, 并增强局部区域的探索。常见的交叉方式主要是二项交叉, 首先对每一个变量都生成一个 $0 \sim 1$ 均匀分布随机数 r 。若 $r \leq cr$, 则接受目标个体的对应分量, 否则, 保留当前个体对应分量, 如下(4)式进行。

$$v_{i,j}(g) = \begin{cases} t_{i,j}(g), & r \leq cr \text{ or } j = \text{rnd} \\ x_{i,j}(g), & \text{otherwise} \end{cases} \quad (4)$$

其中: cr 为用于控制接受目标个体中变量的交叉概率因子; rnd 为区间 $[1, d]$ 之间均匀分布的整数, 用

于确保至少有一维分量继承与目标个体 t_i 。

(4) 选择操作: 基本 DE 算法采用贪婪选择方式, 对于测试个体 $v_i(g)$ 和当前个体 $x_i(g)$, 选择较好的个体进入下一代搜索。即

$$x_i(g+1) = \begin{cases} t_i(g), & f[t_i(g)] < f[x_i(g)] \\ x_i(g), & \text{otherwise} \end{cases} \quad (5)$$

其中: f 为适应度函数; $f(u_i(g+1))$ 为实验个体 $u_i(g+1)$ 对应的适应度值。

2 SADE 算法及数值仿真

2.1 SADE 算法描述

基本 DE 算法的搜索能力与放缩因子及变异算子选取密切相关^[13]。缩放因子值太大, 该算法在寻优过程中最优解易遭到破坏; 缩放因子值太小, 该算法容易陷入局部最优解, 出现早熟现象。针对该算法在搜索初期收敛速度过快, 导致容易陷入局部最优的缺点, 本文引入 Sigmoid 函数与自适应变异算子对该算法的变异操作部分进行改进来提高该算法性能, 具体改进策略如下所述。

变异方式采用差分策略, 即利用种群中个体间的差分向量对个体进行扰动, 实现个体变异。对于放缩因子 F , 由变异方式可知, 其通过对差分向量放缩来扰动基向量。当 F 取较大值时, 能够增强算法的探索能力, 但相应地消减了算法的局部搜索能力; 较小的放缩因子则容易导致算法快速地收敛到局部极小点, 而且很难跳出。

Sigmoid 函数是神经网络中一类重要的激励函数, 具有光滑性, 严格单调性, 饱和性和能够在线性与非线性之呈现出较好的平滑过渡性的特点, 其表达式如(6)式所示:

$$y = f(x) = \frac{1}{1+e^{-x}} \quad (6)$$

其定义域为 $(-\infty, +\infty)$, 值域为 $(-1, 1)$ 。

由于 Sigmoid 函数具有如上良好的性质, 所以构造出如下非线性函数作为放缩因子。

$$F(t) = 1 / \left\{ 1 + \exp \left[\frac{(\ln 1.5 + \ln 1000)t}{T} - \ln 1000 \right] \right\} \quad (7)$$

其中: T 表示最大迭代次数; t 表示当前迭代次数。这样算法在进行迭代过程中放缩因子 $F(t)$ 的值始终出在 0.4~1。而当 $F(t) \in [0.4, 1]$ 时, 算法的性能会大幅度地提升^[13]。就是说, 利用 Sigmoid 函数一方面可以克服变异算子固定或者线性改变而带来的不足, 另一方面也可以使得算法具有较好的搜索策略, 即初始阶段值大, 维持种群多样性, 提高算法的全局搜索能力, 而后期值小, 可以提高局部搜索能力, 加快收敛速度。

对于变异算子的选择, 以 DE/rand/1 和 DE/best/2 最为典型, DE/rand/1 有利于保持群体的多样性, 但是收敛速度慢; DE/best/2 强调算法的收敛速度, 缺点是容易陷入局部极值。为了克服这些缺点, 本文采用自适应变异算子对算法进行改进, 其方程为(8)式所示:

$$t_i(g) = (1 - \lambda) \{ x_{best}(g) + F \cdot [x_{r_4}(g) - x_{r_5}(g)] \} + F \cdot [x_{r_2}(g) - x_{r_3}(g)] + \lambda x_{r_1} \quad (8)$$

式中 λ 设置为线性因子, 形式为如式(9)所示:

$$\lambda = (T - t) / T \quad (9)$$

其中: T 表示算法最大迭代次数; t 表示算法当前迭代次数。随着算法的进行, λ 由 1 逐渐变化为 0, 这样, 算法在初期具有较强的全局搜索能力, 避免过早陷入局部最优, 而在算法的后期则具有较强的局部搜索能力。从而就可以在保证了种群多样性基础上, 还提高了算法精度和收敛速度。

综上所述, 本文提出的 SADE 算法的变异操作函数如(10)式所示:

$$t_i(g) = (1 - \lambda) \{ x_{best}(g) + F(t) \cdot [x_{r_4}(g) - x_{r_5}(g)] \} + F(t) \cdot [x_{r_2}(g) - x_{r_3}(g)] + \lambda x_{r_1} \quad (10)$$

其中: $F(t)$ 可以根据(7)式计算得到。

2.2 数值仿真

为了验证提出的 SADE 算法的有效性, 下面以函数 Griewank 和 Rastrigrin 作为适应度函数进行测试, 其最小值都为 0, 且函数表达式分别为

(1) Griewank 函数:

$$f_1 = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1, |x_i| \leq 600.$$

(2) Rastrigrin 函数:

$$f_2 = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10), |x_i| \leq 5.12.$$

在仿真中, SADE 算法的参数设置为: 种群规模 $NP=100$, 杂交参数 $CR=0.9$, 放缩比例因子 $F=0.6$, 最大迭代次数为 1 000。以 50 次试验的最优解, 解的平均值及方差作为评价标准, 得到相应函数的寻优仿真结果如表 1 所示。

为了进一步说明所提方法的有效性, 对上面的两个函数分别利用基本 DE 算法, 基于 Sigmoid 缩放因子 DE 算法(SDE), 基于自适应变异算子 DE 算法(ADE)和粒子群算法(PSO)^[10]分别对上函数进行 50 次寻优, 结果都罗列在表 1 中。

表 1 仿真结果

函数	算法	最优值	平均值	方差
Griewank	SADE	2.830e-007	1.028e-005	1.762e-010
	基本 DE	0.025	0.036	5.299e-005
	SDE	0.001	0.010	8.870e-005
	ADE	0.007	0.064	7.125e-006
	PSO	0.028	0.052	5.886e-003
Rastrigrin	SADE	3.552e-005	1.516e-004	7.312e-012
	基本 DE	5.126	8.566	8.246
	SDE	2.415	4.343	5.811
	ADE	2.209	3.879	1.119
	PSO	31.321	37.818	64.537

从表 1 可看出, 所提 SADE 算法在寻优的精度和稳定性方面明显优于其它算法, 较好地得到函数的最优值。图 1 和图 2 分别给出了 5 种算法对 2 个函数的寻优过程图。从图中可看出, 所给 SADE 算法在搜索初期克服了收敛速度过快和易陷入局部最优的缺点, 但到后期收敛速度明显变快, 并且很快找到最优解, 且精度明显提高。这是因为, 一方面, SADE 算法利用了 Sigmoid 函数来构造的放缩因子 $F(t)$ 在初期值较大, 从而全局搜索能力得到加强, 克服陷入局部最优, 而随着算法的迭代, $F(t)$ 逐步减小, 使得算法在后期有较强的局部搜索能力。另一方面, SADE 算法还利用了自适应变异

来实施变异操作, 这样也会使得算法在初期具有较强的全局搜索能力, 且在后期局部搜索能力也会增强。总之, 在寻优过程中, SADE 算法使用 Sigmoid 函数而构造的 $F(t)$ 和自适应变异算子来实施变异操作可避免陷入局部最优, 而且可以保证具有高的精确度和稳定性。

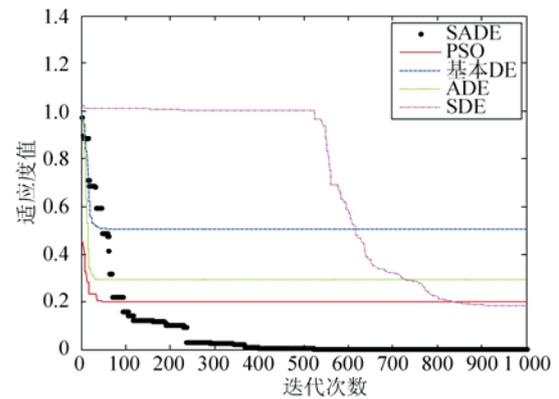


图 1 对 Griewank 函数的寻优过程曲线

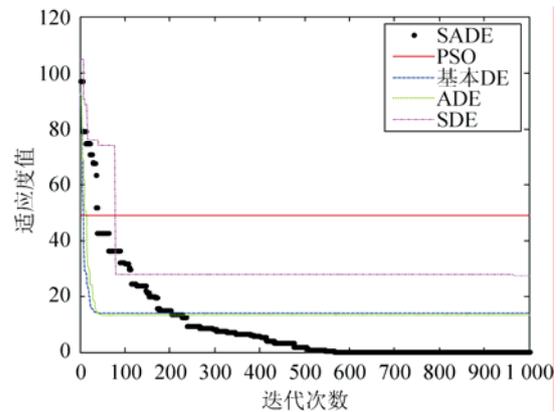


图 2 对 Rastrigrin 函数的寻优过程曲线

3 辨识 Wiener 模型

3.1 问题描述

非线性 Wiener 模型是由一个线性子系统和一个无记忆非线性增益环节串联而成, 链接方式如图 3 所示, 其数学表达式为(11)式所示。

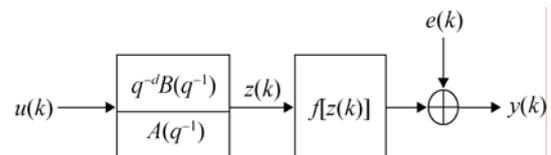


图 3 Wiener 模型

$$\begin{cases} A(q^{-1})z(k) = q^{-d} B(q^{-1})u(k) \\ y(k) = f[z(k)] + e(k) \end{cases} \quad (11)$$

其中: $u(k)$ 和 $y(k)$ 分别是系统的输入和输出; $e(k)$ 为正态白噪声; $z(k)$ 为线性部分的输出; $f(\cdot)$ 表示非线性函数; $A(q^{-1})$ 和 $B(q^{-1})$ 的具体表达式为:

$$\begin{aligned} A(q^{-1}) &= 1 + a_1 q^{-1} + \dots + a_n q^{-n}; \\ B(q^{-1}) &= b_0 + b_1 q^{-1} + \dots + b_m q^{-m}. \end{aligned}$$

其中, q^{-1} 为时延算子。假设 n, m, d 和 $f(\cdot)$ 均已已知, 且定义参数向量 $\theta = [a_1, a_2, \dots, a_n, b_0, b_1, \dots, b_n]^T$ 。

本文辨识的目标是根据给定系统输入 $u(k)$ 和输出 $y(k)$ 来估计参数向量 θ 。假设参数向量 θ 的估计值为 $\hat{\theta} = [\hat{a}_1, \hat{a}_2, \dots, \hat{a}_n, \hat{b}_0, \hat{b}_1, \dots, \hat{b}_n]^T$, 使得 k 时刻的误差预测函数的平方和最小。

$$\min_{\theta} J(k) = \sum_{j=1}^L [y(k-j) - \hat{y}(k-j)]^2 \quad (12)$$

其中: L 为用于对参数辨识的数据点个数; $y(k-j)$, $\hat{y}(k-j)$ 分别为在 $k-j$ 时刻实际系统和辨识模型的输出; $\hat{y}(k-j)$ 可由式(13)得到。

$$\begin{cases} \hat{z}(k-i) = -\hat{a}_1 \hat{z}(k-i) - \dots - \hat{a}_n \hat{z}(k-i-n) + \\ \hat{b}_0 u(k-i-d) + \dots + \\ \hat{b}_m u(k-i-d-m) \\ \hat{y}(k-i) = f[\hat{z}(k-i)] + e(k-i) \end{cases} \quad (13)$$

同时, Wiener 模型的参数满足

$$\theta^{\min} \leq \hat{\theta} \leq \theta^{\max} \quad (14)$$

这样, Wiener 模型参数辨识问题就等价于求解(12)式的最小值问题, 参数估计量 $\hat{\theta}$ 为待优化变量。

3.2 应用 SADE 算法辨识 Wiener 模型

将(12)式作为适应度函数 f , 利用 SADE 算法对其进行寻优, 得到 Wiener 模型的参数向量 θ 的估计值。在利用该算法进行寻优过程中, 令 $x = \hat{\theta}$, $f(k) = J(k)$, 其具体实现步骤如下:

Step 1. 初始化参数, 产生均匀白噪声 $e(k)$;

Step 2. 随机生成 NP 个解来作为初始种群, 并进行评价;

Step 3. 对每个个体 x_i , 随机选择另外 3 个解

来执行变异操作, 从而生成临时解 t_i ;

Step 4. 将临时个体与当前个体进行交叉, 生成新解 v_i ;

Step 5. 将新解与当前解进行比较, 来选择较好的个体;

Step 6. 若满足终止条件, 输出估计值 $\hat{\theta}$, 完成对 Wiener 模型的参数辨识; 否则, 返回 Step 2。

3.3 数值仿真

考虑下面的非线性 Wiener 模型。

$$\begin{cases} x(k) = 1.5x(k-1) - 0.7x(k-2) + \\ u(k-1) + 0.5u(k-2) \\ y(k) = f[x(k)] + e(k) \\ f[x(k)] = \begin{cases} \sqrt{x(k)/2}, & x(k) \geq 0 \\ -\sqrt{x(k)/2}, & x(k) < 0 \end{cases} \end{cases} \quad (15)$$

其中: 输入 $u(k)$ 是 0 均值, 方差为 1 的高斯白噪声, $e(k)$ 是 0 均值和方差为 0.1 的高斯白噪声。用于参数辨识的数据点的个数 $L=500$, 参数选择范围限制在 $[-2, 2]$ 。要辨识的参数向量真值为

$$\theta = [a_1 \ a_2 \ b_0 \ b_1]^T = [-1.5 \ 0.7 \ 1.0 \ 0.5]^T。$$

这里定义均方根误差(root mean square error, RMSE)来衡量辨识精度。

$$RMSE = \sqrt{\frac{\sum_{j=1}^M [y(j) - \hat{y}(j)]^2}{M}} \quad (16)$$

利用 SADE 算法, PSO 算法, ADE 算法, SDE 算法和基本 DE 算法对 Wiener 模型进行参数辨识, 结果罗列在表 2 中。从表 2 可看出, 与其它算法比较, 利用本文所提 SADE 算法来辨识得到的结果精度明显最高, 且均方根误差也最小。并且图 4~8 给出了相应算法对 Wiener 模型进行参数辨识的过程曲线图。从图 4 可以明显看出, SADE 算法在迭代代数 50 左右的时候就可以比较准确的辨识出 Wiener 模型的 4 个参数。相对于其它算法, 所给算法收敛速度快, 辨识精度高, 具有良好的稳定性。

表 2 辨识结果

参数	真值	估计值									
		SADE	RMSE	SDE	RMSE	ADE	RMSE	基本 DE	RMSE	PSO	RMSE
a_1	1.5	1.500 1	2.071e-4	1.499 7	3.022e-3	1.500 1	2.512e-3	1.499 8	4.506e-3	1.500 2	3.312e-2
a_2	-0.7	-0.700 0		-0.700 6		-0.699 9		-0.699 8		-0.709 1	
b_1	1.0	1.000 1		1.000 1		1.000 2		1.000 4		1.000 5	
b_2	0.5	0.500 0		0.499 8		0.499 8		0.499 6		0.499 4	

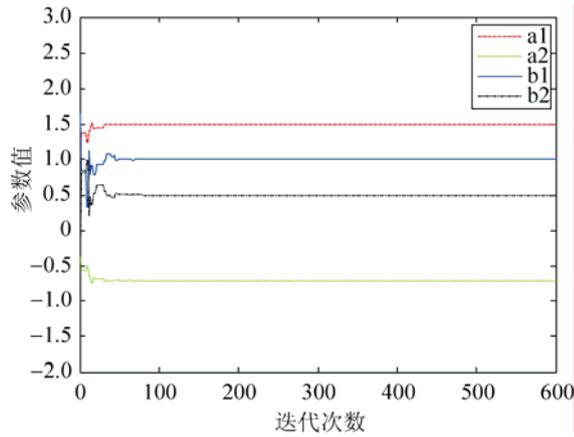


图 4 SADE 算法辨识参数过程曲线图

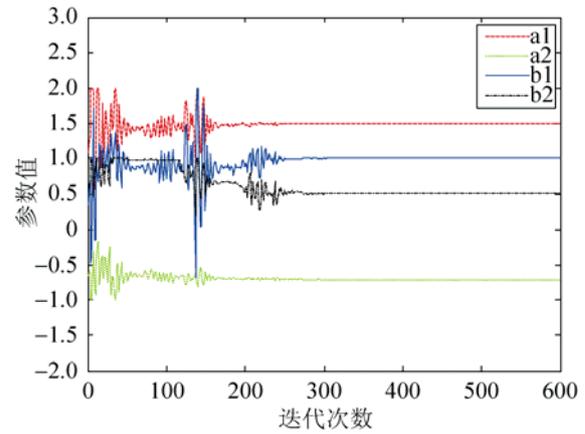


图 7 基本 DE 算法辨识参数过程曲线图

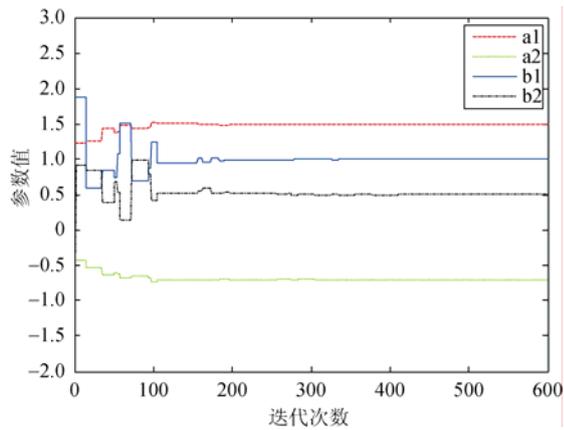


图 5 ADE 算法辨识参数过程曲线图

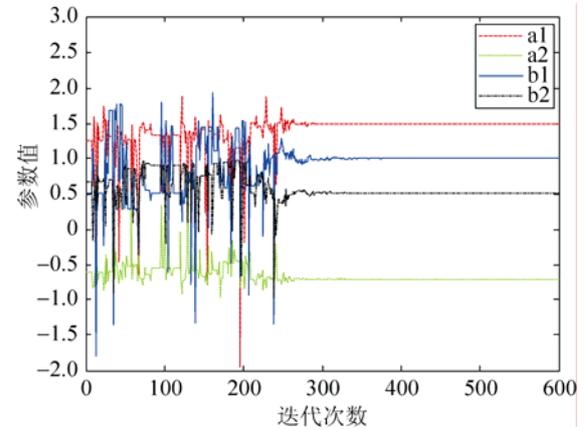


图 8 PSO 算法辨识参数过程曲线图

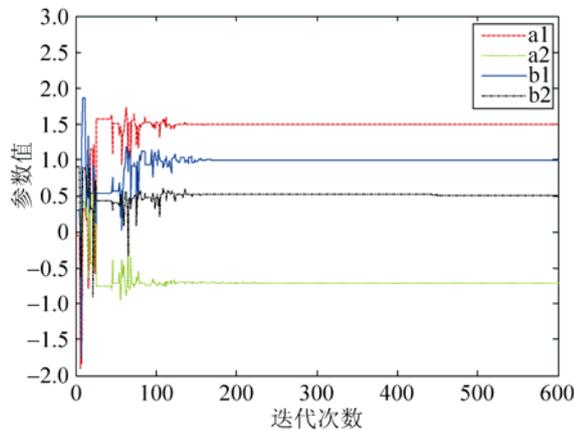


图 6 SDE 算法辨识参数过程曲线图

根据以上的结果,可以明显看出,利用本文所给 SADE 算法对 Wiener 模型进行参数辨识是有效的和可行的。

4 结论

对非线性 Wiener 模型,利用差分进化算法给出了一种参数辨识方法。为了进一步提高算法性能,提出了一种改进差分进化算法。仿真结果表明,所给参数辨识方法是可行的。

参考文献:

- [1] 萧德云. 系统辨识理论及其应用 [M]. 北京: 清华大学出版社, 2014: 56-68.
- [2] 徐小平, 钱富才, 刘丁, 等. 基于 PSO 算法的系统辨识方法 [J]. 系统仿真学报, 2008, 20(13): 3525-3528. (Xu Xiaoping, Qian Fucai, Liu Ding, et al. Method of System Identification Based on PSO Algorithm [J]. Journal of System Simulation (S1004-731X), 2008, 20(13): 3525-3528.)
- [3] 钱富才, 黄娇茹, 秦新强. 基于鲁棒优化的系统辨识算法研究 [J]. 自动化学报, 2014, 40(5): 988-993.
- [4] 马海芳, 朱凌云, 王富强. 基于 Wiener 模型的 pH 中和过程非线性 DMC 控制 [J]. 化工自动化及仪表, 2010, 37(9): 33-36.
- [5] 刘长良, 任燕燕, 王东风, 等. 热工系统 Hammerstein-Wiener 模型辨识 [J]. 计算机仿真, 2013, 30(9): 394-397.
- [6] 胡德文, 王正志. 非线性系统 Wiener 模型辨识 [J]. 自动化学报, 1991, 17(2): 151-159.
- [7] 罗秋斌, 李秀英, 韩志刚, 等. 一种非线性系统多项式逼近的建模方法 [J]. 系统仿真学报, 2008, 20(4): 829-832. (Luo Qiubin, Li Xiuying, Han Zhigang, et al. Polynomial Approach for Nonlinear Modeling [J]. Journal of System Simulation, 2008, 20(4): 829-832.)
- [8] 吴德会. 非线性动态系统的 Wiener 神经网络辨识法 [J]. 控制理论与应用, 2009, 26(11): 1192-1196.
- [9] Hatanaka T, Uosaki K, Koga M. Evolutionary Computation Approach to Wiener Model Identification [C]// Proceedings of IEEE Congress on Evolutionary. Piscataway, USA: IEEE Press, 2002: 914-919.
- [10] 张燕, 李少远, 王笑波, 等. 基于粒子群优化的 Wiener 模型辨识与实例研究 [J]. 控制理论与应用, 2006, 23(6): 992-994.
- [11] Storn R, Price K. Differential Evolution: a Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces [J]. Global Optimization (S1354-1013), 1997, 11(4): 341-359.
- [12] Wang L, Li L P. Fixed-Structure H^∞ Controller Synthesis Based on Differential Evolution with Level Comparison [J]. IEEE Transactions on Evolutionary Computation (S1089-778X), 2011, 15(1): 120-129.
- [13] Onwubolu G, Davendra D. Scheduling Flow Shops Using Differential Evolution Algorithm [J]. European Journal of Operational Research (S0377-2217), 2006, 171(2): 674-692.
- [14] 王凌, 钱斌. 混合差分进化与调度算法 [M]. 北京: 清华大学出版社, 2012: 112-121.
- [15] 王永皎. 改进自适应差分进化算法求解大规模整数任务分配 [J]. 计算机应用, 2012, 32(8): 2165-2167.
- [16] Chang W D. Parameter Identification of Rossler's Chaotic System by an Evolutionary Algorithm [J]. Science (S0036-8075), 2006, 29(5): 1047-1053.

(上接第 146 页)

- [5] Zhou Y R. Stochastic resonance in a time-delayed monostable system with correlated multiplicative and additive white noise [J]. Chin. Phys. B, 2011, 20(1): 010501-1-010501-6.
- [6] 周涛, 王嘉. α 稳定分布综述 [J]. 电声技术, 2011, 35(3): 57-60.
- [7] 李娟娟, 徐勇, 冯晶. Duffing 系统中 Lévy 噪声诱导的随机共振与相转移 [J]. 动力学与控制学报, 2012, 10(3): 278-5.
- [8] Zeng L Z, Xu B H. Effects of asymmetric Lévy noise in parameter-induced aperiodic stochastic resonance [J]. Physica A (S0378-4371), 2010, 389(22): 5128-5136.
- [9] 张广丽, 吕希路, 康艳梅. α 稳定噪声环境下过阻尼系统中的参数诱导随机共振现象 [J]. 物理学报, 2012, 61(4): 040501-1-040501-8.
- [10] 张文英, 王自力, 张卫东. 利用随机共振实现 Lévy 噪声中的信号检测 [J]. 控制工程, 2009, 16(5): 638-640.
- [11] 焦尚彬, 何童. 基于双稳随机共振的多频弱信号检测 [J]. 计算机工程与应用, 2013, 50(5): 221-226.
- [12] 杨定新, 胡鸢庆. 随机共振在微弱信号检测中的数值仿真 [J]. 国防科技大学学报, 2003, 25(5): 91-94.
- [13] 焦尚彬, 任超, 黄伟超, 等. 稳定噪声环境下多频微弱信号检测的参数诱导随机共振现象 [J]. 物理学报, 2013, 62(21): 210501-1-210501-10.
- [14] 焦尚彬, 任超, 李鹏华, 等. 乘性和加性 α 稳定噪声环境下的过阻尼单稳随机共振现象 [J]. 物理学报, 2014, 63(7): 070501-1-070501-9.