ICIC Express Letters
Part B: Applications
Volume **7**, Number **5**, May **2016**

ICIC International ©2016 ISSN 2185-2766
pp. 1173–1178

DE-BASED PARAMETER ESTIMATION SCHEME FOR BILINEAR POLYNOMIAL FILTERS

Wei-Der Chang*, Shun-Peng Shih and Ching-Lung Chi

Department of Computer and Communication
Shu-Te University
No. 59, Hengshan Rd., Yanchao, Kaohsiung County 82445, Taiwan
*Corresponding author: wdchang@stu.edu.tw

Received October 2015; accepted January 2016

Abstract. *This paper focuses on the parameter estimation for bilinear polynomial filters. The design scheme is the differential evolution (DE) algorithm which is one of evolutionary computations. This kind of algorithm is full with real-valued operations during the optimization. Due to some excellent properties in the DE algorithm, it has successfully been applied in solving a variety of practical engineering optimization problems. With the use of the DE algorithm, the parameter estimation problem especially for the bilinear polynomial filter is discussed in this paper. By minimizing the error signal between the actual output and model output, the filter coefficients can be accurately estimated. Finally, several independent runs with different sets of initial conditions are examined to confirm the robustness and feasibility of the DE algorithm.*

Keywords: Differential evolution (DE) algorithm, Bilinear polynomial filter, Parameter estimation

1. **Introduction.** The differential evolution (DE) was proposed by Storn and Price in 1997 [1]. It has been proven to be an excellent searching algorithm for solving optimized problems. Like other evolutionary computations, the DE is also a multi-direction searching algorithm because it is based on a population consisting of many designed parameter vectors. In recent years, a variety of engineering optimization problems have been solved and explored by using the DE algorithm [2-9]. In [3], the authors proposed an optimization scheme in which the DE is used to choose shape parameter and node distribution when applying the radial basis function meshless numerical method. In [5], a specialized DE technique to solve the transmission expansion planning (TEP) problem was developed and some comparisons were performed with other swarm methods. Moreover, an optimization problem of heliostat field layout in solar central receiver systems on annual basis has been solved using the DE algorithm [6].

Bilinear polynomial filter is a nonlinear digital system and is an extended version of the infinite impulse response (IIR) digital filter. In addition to the input signal and output signal, the bilinear polynomial filter also contains the products of input and output signals. Thus, its modeling capacity is absolutely superior to the original IIR digital system, and conversely the complexity to design the bilinear digital system is more difficult than the IIR due to the nonlinearity. Recently, a large number of related researches on the bilinear polynomial filter have been developed and investigated such as nonlinear system modeling [2], parameter estimation [10,11], system identification [12-14], and multichannel active control [15].

In References [10] and [11], they utilized the adaptive particle swarm optimization and bilinear recursive least square (BRLS) algorithm, respectively, to solve the parameter estimation problem of the bilinear filter. However, in this paper we will propose another novel parameter estimation scheme for bilinear polynomial filter via the DE optimal algorithm. According to the design steps, the coefficients of bilinear digital system can be correctly estimated. At the beginning of the algorithm, all of designed filter coefficients are collected to be a parameter vector, and many such parameter vectors further form a so-called population. To achieve optimization, three evolutionary mechanisms including the mutation, crossover, and selection operation are employed in the DE algorithm. All parameter vectors in the population will be evolved by these three operations. The remainder of this paper is organized as follows. In Section 2, the difference equation expression for the bilinear digital system is introduced. Section 3 will explain the DE algorithm in detail and the DE-based design steps for parameter estimations are also given. Some simulation results including several independent run tests are provided to show the applicability and robustness of the developed method. Finally, a simple conclusion and future research direction are addressed in Section 5.

2. **Bilinear Polynomial Filter.** The IIR digital filter is a basic and very important filter and is broadly used in the digital signal processing (DSP) field. Its present output is fully influenced by both the previous output signals and present and previous input signals. For the digital filter design, this kind of filter can use fewer coefficients than the finite impulse response (FIR) digital filter to achieve the same filtering performance. However, the stability problem for designing the IIR digital system should always be taken into account. Equation (1) shows the difference equation structure for the IIR digital filter

$$\begin{aligned} y[n] &= \sum_{k=1}^{N} a_k y[n-k] + \sum_{k=0}^{M} b_k x[n-k] = a_1 y[n-1] + a_2 y[n-2] + \cdots + a_N y[n-N] \\ &\quad + b_0 x[n] + b_1 x[n-1] + \cdots + b_M x[n-M] \end{aligned} \tag{1}$$

where x and y denote the system input and output signals, respectively, a_k and b_k represent the system coefficients, N is the number of past outputs and is also referred to as the system order, and M is the number of the past input signals. Bilinear polynomial filter considered in this study is an extended version of the IIR digital filter and can be expressed by Equation (2)

$$\begin{aligned} y[n] &= \sum_{k=1}^{N} a_k y[n-k] + \sum_{k=0}^{M} b_k x[n-k] + \sum_{k_1=0}^{M} \sum_{k_2=1}^{N} c_{k_1 k_2} x[n-k_1] y[n-k_2] \\ &= a_1 y[n-1] + a_2 y[n-2] + \cdots + a_N y[n-N] \\ &\quad + b_0 x[n] + b_1 x[n-1] + \cdots + b_M x[n-M] \\ &\quad + c_{01} x[n] y[n-1] + c_{02} x[n] y[n-2] + \cdots + c_{0N} x[n] y[n-N] \\ &\quad + c_{11} x[n-1] y[n-1] + c_{12} x[n-1] y[n-2] + \cdots + c_{1N} x[n-1] y[n-N] + \cdots \\ &\quad + c_{M1} x[n-M] y[n-1] + c_{M2} x[n-M] y[n-2] + \cdots + c_{MN} x[n-M] y[n-N] \end{aligned} \tag{2}$$

where $c_{k_1 k_2}$ is also the system coefficient which corresponds to the product of the input and output signals. For simplification and convenience to the use of DE algorithm, Equation (2) can further be rewritten as a vector form

$$y[n] = \Theta U^T \tag{3}$$

where Θ is a collection of all the system coefficients defined by

$$\Theta = [\theta_1, \theta_2, \cdots, \theta_K] = [a_1, \cdots, a_N, b_0, \cdots, b_M, c_{01}, \cdots, c_{MN}] \tag{4}$$

with the vector length $K = N + (M+1) + (M+1)N = (M+1)(N+1) + N$, and U is another collection of all output, input, and their product signals given by

$$U = [y[n-1], \cdots, y[n-N], x[n], \cdots, x[n-M], x[n]y[n-1], \cdots, x[n-M]y[n-N]]. \tag{5}$$

Equation (4) is called the parameter vector in the viewpoint of DE algorithm and this parameter vector will be correctly solved based on the DE algorithm according to a series of input-output signal pairs.

3. DE-based Design Steps for Parameter Estimation of Bilinear Polynomial Filter. As described in Section 1, the DE algorithm is composed of three evolutionary mechanisms including the mutation, crossover, and selection operations. In this section, we will further explain these operations [2]. In the beginning of the algorithm, a system cost function to evaluate the performance of each parameter vector should be defined. In this study, it is simply defined by Equation (6)

$$CF = \sum_{n=0}^{L} e^2[n] = \sum_{n=0}^{L} [y[n] - \hat{y}[n]]^2 \quad (6)$$

where L represents the sampling number and e is the error signal between the actual system output y and estimated model output $\hat{y}$. A parameter vector with less cost function stands for a better one. With the definition of cost function in Equation (6), the following operations are executed for each parameter vector inside the population. Here the population size is denoted by PS.

In the mutation operation, a new mutated vector $V = [v_1 v_2, \cdots, v_K]$ is obtained using Equation (7)

$$V = \Theta_\alpha + F \cdot (\Theta_\beta - \Theta_\gamma) \quad (7)$$

where Θ_α, Θ_β, and Θ_γ are three different parameter vectors randomly chosen from the population and $F \in [0, 2]$ is a mutation constant factor that controls the amplification of the differential variation $\Theta_\beta - \Theta_\gamma$. Equation (7) reveals that the mutated vector V is a full combination of Θ_α, Θ_β, and Θ_γ. The derived mutated vector V will further cross with a target vector $\Theta = [\theta_1, \theta_2, \cdots, \theta_K]$. In the crossover operation, it is to interchange some elements between mutated vector V and target vector Θ. In order to achieve that, a new vector $[r_1, r_2, \cdots, r_K]$ is generated where r_i is a uniformly random number generated from the interval $[0, 1]$ for $i = 1, 2, \cdots, K$, and another set of binary sequence $[p_1, p_2, \cdots, p_K]$ is derived by Equation (8)

$$p_i = \begin{cases} 1, & \text{if } r_i < CR \\ 0, & \text{otherwise} \end{cases}, \quad \text{for } i = 1, 2, \cdots, K \quad (8)$$

where $CR \in [0, 1]$ is the crossover rate and it is always set to 0.5. Based on this binary sequence, a trial vector $W = [w_1, w_2, \cdots, w_K]$ can eventually be obtained by

$$w_i = \begin{cases} \theta_i, & \text{if } p_i = 1 \\ v_i, & \text{if } p_i = 0 \end{cases}, \quad \text{for } i = 1, 2, \cdots, K \quad (9)$$

It is concluded from the crossover formula of Equation (9) that the trial vector W is a full exchange outcome between V and Θ. Next step is to execute the selection operation on both the trial vector W and the original target vector Θ. In brief, the selection is to keep the excellent parameter vector and discard the bad one. As a result, the cost functions of both W and Θ need to be calculated. If $CF(W) < CF(\Theta)$, i.e., the trial vector is superior to the target vector, then the algorithm keeps this derived trial vector and discards the original target vector; otherwise the target vector still survives in the population and omits the trial vector.

To perform the above three DE operations one time is called a generation or an iteration of the algorithm. In general, there are two kinds of conditions to stop the algorithm: the assigned number of iterations G is attained or the cost function derived during the optimization is already met. In this paper, the algorithm stops when the former is satisfied. Finally, Figure 1 displays the complete system block diagram for parameter estimation of bilinear polynomial filter via the DE algorithm approach.

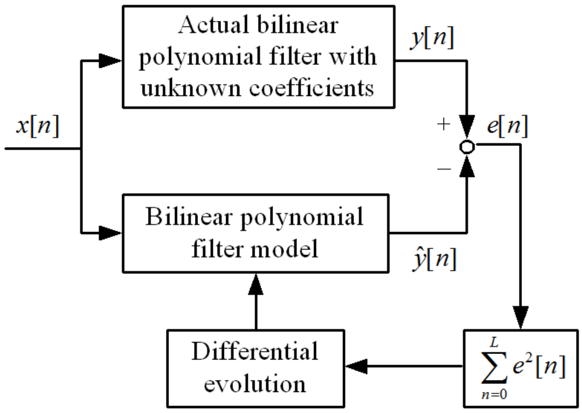

FIGURE 1. System block diagram for parameter estimation of bilinear polynomial filter using the DE algorithm

4. **Simulation Results.** To verify the applicability of the proposed method, some simulation examinations are provided in this section. Equation (10) shows the bilinear polynomial filter and its corresponding coefficients will be estimated

$$y[n] = -0.65y[n-2]+1.5x[n-1]+0.76x[n]y[n-1]-2x[n]y[n-2]+0.4x[n-1]y[n-1] \quad (10)$$

Referring to Equation (2), the actual filter coefficients are $a_2 = -0.65$, $b_1 = 1.5$, $c_{01} = 0.76$, $c_{02} = -2$, and $c_{11} = 0.4$, respectively. To solve the coefficient estimation problem of Equation (10), the related parameter settings used in the DE algorithm are listed in Table 1. To excite both the bilinear polynomial filter and bilinear polynomial model as shown in Figure 1, the input signal $x[n]$ is the random number generated from the interval $[-1, 1]$ only for $0 \leq n \leq 50$, and the initial value of each parameter vector is also chosen from $[-1, 1]$ randomly. Furthermore, five independent runs with different sets of initial conditions (Run 1 $\sim$ Run 5) are performed to show the robustness of the algorithm.

TABLE 1. Related parameter setting used in the DE algorithm

Number of iterations G	Sampling number L	Population size PS	Mutation constant factor F
2000	50	40	0.5

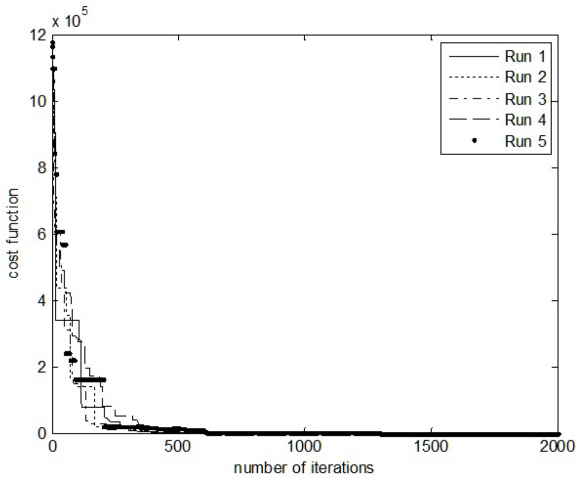

FIGURE 2. Convergence trajectories of all cost functions for Run 1 $\sim$ Run 5

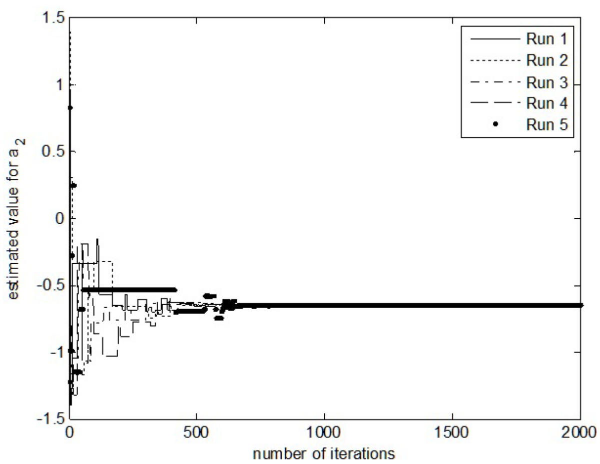

FIGURE 3. Convergence trajectories of estimated parameters a_2 for Run 1 $\sim$ Run 5

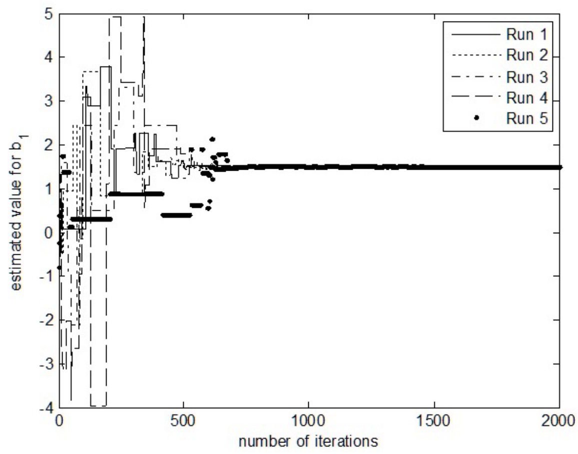

FIGURE 4. Convergence trajectories of estimated parameters b_1 for Run 1 $\sim$ Run 5

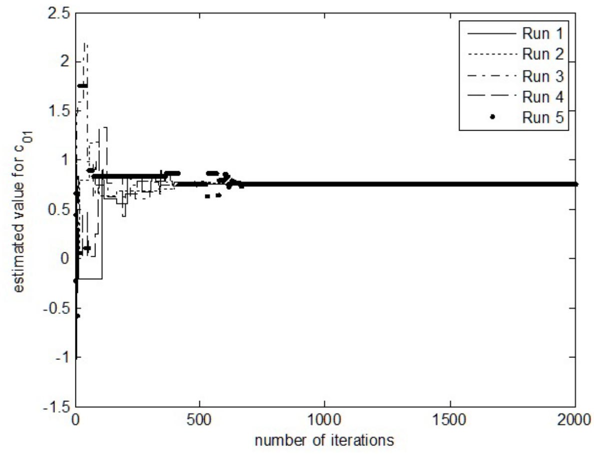

FIGURE 5. Convergence trajectories of estimated parameters c_{01} for Run 1 $\sim$ Run 5

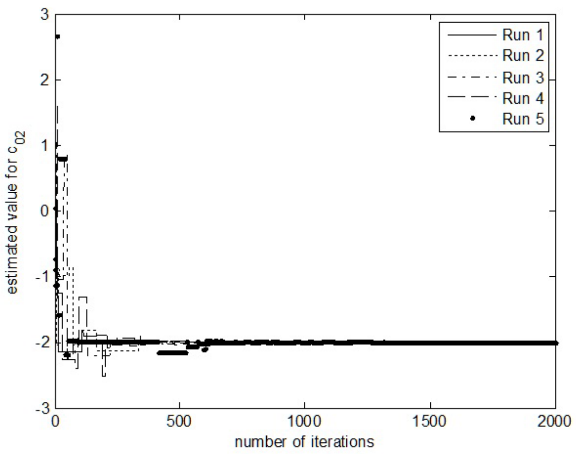

FIGURE 6. Convergence trajectories of estimated parameters c_{02} for Run 1 $\sim$ Run 5

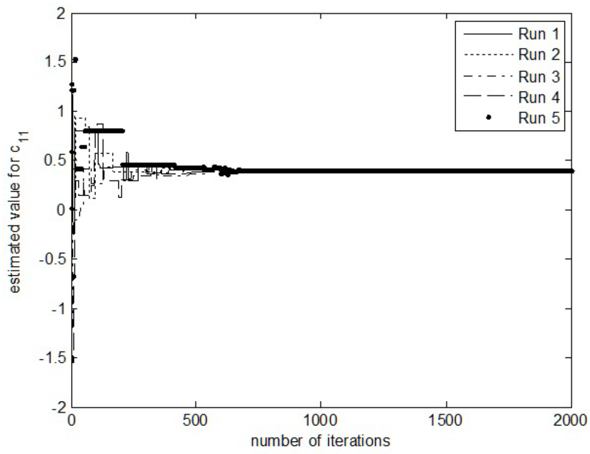

FIGURE 7. Convergence trajectories of estimated parameters c_{11} for Run 1 $\sim$ Run 5

Simulation results are displayed in Figures 2-7. Figure 2 shows all convergence trajectories of cost functions for Run 1 $\sim$ Run 5 with respect to the number of iterations. It is clearly seen from Figure 2 that they eventually approximate to zero after about 1000 iterations. In addition, the convergence trajectories of all filter coefficients estimated are also shown in Figures 3-7, respectively, for Run 1 $\sim$ Run 5. As can be seen from these figures, the actual filter coefficients can be accurately solved for any of independent runs.

5. **Conclusions and Future Work.** This paper has proposed a new parameter estimation scheme for the bilinear polynomial filter by the differential evolution algorithm. For the use of the algorithm, all filter coefficients estimated are collected to form a parameter vector. By executing three main evolutionary operations on all parameter vectors inside the population, the actual filter parameters can be correctly solved. Furthermore, the robustness of the proposed method is also guaranteed by examining five independent runs with different sets of initial conditions. Simulation results sufficiently show the applicability of the proposed method on the coefficient estimation of the bilinear polynomial filter. In our future study, other new evolutionary computations including the frog leap

algorithm, particle swarm optimization, and artificial bee colony may be considered to tackle the same problem, and some comprehensive comparisons are further made.

Acknowledgment. This work was supported by the Ministry of Science and Technology of Taiwan under the grant MOST 104-2221-E-366-006.

REFERENCES

[1] R. Storn and K. Price, Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces, *Journal of Global Optimization*, vol.11, pp.341-359, 1997.

[2] W. D. Chang, Differential evolution-based nonlinear system modeling using a bilinear series model, *Applied Soft Computing*, vol.12, pp.3401-3407, 2012.

[3] C. M. C. Roque and P. A. L. S. Martins, Differential evolution optimization for the analysis of composite plates with radial basis collocation meshless method, *Composite Structures*, vol.124, pp.317-326, 2015.

[4] I. Poikolainen, F. Neri and F. Caraffini, Cluster-based population initialization for differential evolution frameworks, *Information Science*, vol.297, pp.216-235, 2015.

[5] S. P. Torres and C. A. Castro, Specialized differential evolution technique to solve the alternating current model based transmission expansion planning problem, *Electrical Power and Energy Systems*, vol.68, pp.243-251, 2015.

[6] M. Atif and F. A. Al-Sulaiman, Optimization of heliostat field layout in solar central receiver systems on annual basis using differential evolution algorithm, *Energy Conversion and Management*, vol.95, pp.1-9, 2015.

[7] J. Tvrdik and I. Krivy, Hybrid differential evolution algorithm for optimal clustering, *Applied Soft Computing*, vol.35, pp.502-512, 2015.

[8] M. Basu, Improved differential evolution for economic dispatch, *Electrical Power and Energy Systems*, vol.63, pp.855-861, 2014.

[9] Y. Bi, D. Srinivasan, X. Lu, Z. Sun and W. Zeng, Type-2 fuzzy multi-intersection traffic signal control with different evolution optimization, *Expert Systems with Applications*, vol.41, pp.7338-7349, 2014.

[10] H. Modares, A. Alfi and M. B. N. Sistani, Parameter estimation of bilinear systems based on an adaptive particle swarm optimization, *Engineering Applications of Artificial Intelligence*, vol.23, pp.1105-1111, 2010.

[11] S. K. Singh, A. K. Goswami and N. Sinha, Power system harmonic parameter estimation using bilinear recursive least square (BRLS) algorithm, *Electrical Power and Energy Systems*, vol.67, pp.1-10, 2015.

[12] P. L. dos Santos, J. A. Ramos and J. L. Martins de Carvalho, Identification of a benchmark Wiener-Hammerstein: A bilinear and Hammerstein-bilinear model approach, *Control Engineering Practice*, vol.20, pp.1156-1164, 2012.

[13] Th. S. S. Singh and A. Chatterjee, A comparative study of adaptation algorithms for nonlinear system identification based on second order Volterra and bilinear polynomial filters, *Measurement*, vol.44, pp.1915-1923, 2011.

[14] J. Zhang and H. Zhao, A novel adaptive bilinear filter based on pipelined architecture, *Digital Signal Processing*, vol.20, pp.23-38, 2010.

[15] H. Zhao, X. Zeng, Z. He, T. Li and W. Jin, Nonlinear adaptive filter-based simplified bilinear model for multichannel active control of nonlinear noise processes, *Applied Acoustic*, vol.74, pp.1414-1421, 2013.