Computación y Sistemas
ISSN: 1405-5546
computacion-y-sistemas@cic.ipn.mx
Instituto Politécnico Nacional
México

Cuevas, Erik; Gálvez, Jorge; Avalos, Omar
Parameter Estimation for Chaotic Fractional Systems by Using the Locust Search Algorithm
Computación y Sistemas, vol. 21, núm. 2, 2017, pp. 369-380
Instituto Politécnico Nacional
Distrito Federal, México

ISSN 2007-9737

Parameter Estimation for Chaotic Fractional Systems by Using the Locust Search Algorithm

Erik Cuevas, Jorge Gálvez, Omar Avalos

Universidad de Guadalajara, CUCEI, Departamento de Electrónica, Mexico

erik.cuevas@cucei.udg.mx

Abstract. Due to its multiple applications, parameter identification for fractional-order chaotic systems has attracted the interests of several research communities. In the identification, the parameter estimation process is transformed into a multidimensional optimization problem where fractional orders, as well as functional parameters of the chaotic system are considered the decision variables. Under this approach, the complexity of fractional-order chaotic systems tends to produce multimodal error surfaces for which their cost functions are significantly difficult to minimize. Several algorithms based on evolutionary computation principles have been successfully applied to identify the parameters of fractional-order chaotic systems. However, most of them maintain an important limitation; they frequently obtain sub-optimal results as a consequence of an inappropriate balance between exploration and exploitation in their search strategies. This paper presents an algorithm for parameter identification of fractional-order chaotic systems. In order to determine the parameters, the proposed method uses the evolutionary method called Locust Search (LS), which is based on the behavior of swarms of locusts. Different to the most of existent evolutionary algorithms, it explicitly avoids the concentration of individuals in the best positions, eliminating critical flaws such as the premature convergence to sub-optimal solutions and the limited exploration-exploitation balance. Numerical simulations have been conducted on the fractional-Order Van der Pol oscillator to show the effectiveness of the proposed scheme.

Keywords. Locust search, fractional-order systems, evolutionary computation, parameter identification, Van der Pol oscillator.

1 Introduction

A fractional order model is a system that is characterized by a fractional differential equation containing derivatives of non-integer order. Several engineering problems, such as transmission lines [1], electrical circuits [2] and control systems [3], can be more accurately described by fractional differential equations than integer order schemes. For this reason, in the last decade, the fractional order systems [4–8] have attracted the interests of several research communities.

System identification is a practical way to model a fractional order system. However, because the mathematical interpretation of fractional calculus is lightly distinct to integer calculus, it is difficult to model real fractional order systems directly based on analytic mechanisms [9]. For classical integer order system, once the maximum order of the system has been defined, the parameters of the model can be identified directly. However, for a fractional order system, because identification requires the choice of the fractional order of the operators, and the systematic parameters, the identification process of such systems is more complex than that of the integer order models [10]. Under such conditions, most of the classical identification methods cannot directly applied to identification of a fractional order systems [11].

The problem of estimating the parameters of fractional order systems has been commonly solved through the use of deterministic methods such as non-linear optimization techniques [12], input output frequency contents [13] or operational matrix [14]. These methods have been exhaustively analyzed and represent the most consolidated available tools. The interested reader in such approaches can be referred to [15] for a recent survey on the state-of-the-art.

As an alternative to classical techniques, the problem of identification in fractional order systems has also been handled through evolutionary methods. In general, they have demonstrated, under several circumstances, to deliver better results than those based on deterministic approaches in terms of accuracy and robustness [16]. Under these methods, an individual is represented by a candidate model. Just as the evolution process unfolds, a set of evolutionary operators are applied in order to produce better individuals.

The quality of each candidate solution is evaluated through an objective function whose final result represents the affinity between the estimated model and the actual one. Some examples of these approaches used in the identification of fractional order systems involve methods such as Genetic Algorithms (GA) [17], Artificial Bee Colony (ABC) [18], Differential Evolution (DE) [19] and Particle Swarm Optimization (PSO) [20]. Although these algorithms present interesting results, they have an important limitation:

They frequently obtain sub-optimal solutions as a consequence of the limited balance between exploration and exploitation in their search strategies. This limitation is associated to their evolutionary operators employed to modify the individual positions. In such algorithms, during their operation, the position of each individual for the next iteration is updated producing an attraction towards the position of the best particle seen so-far or towards other promising individuals. Therefore, as the algorithm evolves, such behaviors cause that the entire population rapidly concentrates around the best particles, favoring the premature convergence and damaging the appropriate exploration of the search space [21, 22].

This paper presents an algorithm for parameter identification of fractional-order chaotic systems. In order to determine the parameters, the proposed method uses a novel evolutionary method called Locust Search (LS) [23, 31, 32] which is based on the behavior of swarms of locusts. In the proposed algorithm, individuals emulate a group of locusts which interact to each other based on the biological laws of the cooperative swarm.

The algorithm considers two different behaviors: solitary and social. Depending on the behavior, each individual is conducted by a set of evolutionary operators which mimics different cooperative conducts that are typically found in the swarm. Different to most of existent evolutionary algorithms, the behavioral model in the proposed approach explicitly avoids the concentration of individuals in the current best positions. Such fact allows avoiding critical flaws such as the premature convergence to sub-optimal solutions and the incorrect exploration-exploitation balance. Numerical simulations have been conducted on the fractional-Order Van der Pol oscillator to show the effectiveness of the proposed scheme.

The paper is organized as follows. In Section 2, the concepts of fractional calculus are introduced. Section 3 gives a description for the Locust Search algorithm. Section 4 gives a brief description of the fractional-order Van der Pol Oscillator. Section 5 formulates the parameter estimation problem. Section 6 shows the experimental results. Finally some conclusions are discussed in Section 7.

2 Fractional Calculus

Fractional calculus is a generalization of integration and differentiation to non-integer order fundamental operator. The differential-integral operator, denoted by ${}_aD_t^q$ takes both the fractional derivative and the fractional integral in a single expression which is defined as:

$$ {}_aD_t^q = \begin{cases} \dfrac{d^q}{dt^q}, & q > 0, \\ 1, & q = 0, \\ \displaystyle\int_a^t (d\tau)^q, & q < 0, \end{cases} \tag{1} $$

where a and t represents the operation bounds whereas $q \in \Re$. The commonly used definitions for fractional derivatives are the Grünwald-Letnikov, Riemann-Liouville [24] and Caputo [25]. According to the Grünwald-Letnikov

approximation, the fractional-order derivative of order q is defined as follows:

$$D_t^q f(t) = \lim_{h\to 0} \frac{1}{h^q} \sum_{j=0}^{\infty} (-1)^j \binom{q}{j} f(t - jh). \quad (2)$$

In the numerical calculation of fractional-order derivatives, the explicit numerical approximation of the q-th derivative at the points $kh, (k = 1, 2, ...)$ maintains the following form [26]:

$${}_{(k-L_m/h)}D_{t_k}^q f(t) \approx h^{-q} \sum_{j=0}^{k} (-1)^j \binom{q}{j} f(t_k - j). \quad (3)$$

where L_m is the memory length $t_k = kh$, h, is the time step and $(-1)^j \binom{q}{j}$ are the binomial coefficients. For their calculation we can use the following expression:

$$c_0^{(q)} = 1, \quad c_j^{(q)} = \left(1 - \frac{1+q}{j}\right) c_{j-1}^{(q)}. \quad (4)$$

Then, the general numerical solution of the fractional differential equation is defined as follows:

$$y(t_k) = f(y(t_k), t_k) h^q - \sum_{j=1}^{k} c_j^{(q)} y(t_{k-j}). \quad (5)$$

3 Locust Search (LS) Algorithm

In the operation of LS [23], a population $\mathbf{L}^k$ $(\{\mathbf{l}_1^k, \mathbf{l}_2^k, \ldots, \mathbf{l}_N^k\})$ of N locusts (individuals) is processed from the initial stage (k=0) to a total *gen* number iterations (k= *gen*). Each individual $\mathbf{l}_i^k$ ($i \in [1, \ldots, N]$) symbolizes an n-dimensional vector $\{l_{i,1}^k, l_{i,2}^k, \ldots, l_{i,n}^k\}$ where each dimension represents a domain variable of the optimization problem to be solved. The set of variables represents the valid search space $\mathbf{S} = \{\mathbf{l}_i^k \in \mathbb{R}^n \,|\, lb_d \le l_{i,d}^k \le ub_d\}$, where lb_d and ub_d represents the lower and upper bounds for the d dimension, respectively. The quality of each element $\mathbf{l}_i^k$ (candidate solution) is evaluated by using the objective function $f(\mathbf{l}_i^k)$. In LS, at each iteration consists of two operators: (A) solitary and (B) social.

3.1 Solitary Operation (A)

In the solitary operation, a new location $\mathbf{p}_i$ ($i \in [1, \ldots, N]$) is generated by modifying the current element location $\mathbf{l}_i^k$ with a change of position $\Delta\mathbf{l}_i$ $(\mathbf{p}_i = \mathbf{l}_i^k + \Delta\mathbf{l}_i)$. $\Delta\mathbf{l}_i$ is the result of the individual interactions experimented by $\mathbf{l}_i^k$ as a consequence of its biological behavior. Such interactions are pairwise computed among $\mathbf{l}_i^k$ and the other N-1 individuals in the swarm. Therefore, the final force exerted between $\mathbf{l}_j^k$ and $\mathbf{l}_i^k$ is computed by considering the following model:

$$\mathbf{s}_{ij}^m = \rho(\mathbf{l}_i^k, \mathbf{l}_j^k) \cdot s(r_{ij}) \cdot \mathbf{d}_{ij} + rand(1, -1), \quad (6)$$

where $\mathbf{d}_{ij} = (\mathbf{l}_j^k - \mathbf{l}_i^k)/r_{ij}$ is the unit-vector, pointing from $\mathbf{l}_i^k$ to $\mathbf{l}_j^k$. Furthermore, *rand*(1,-1) is an number randomly produced between 1 and -1. The factor $s(r_{ij})$ represents the social relation between $\mathbf{l}_j^k$ and $\mathbf{l}_i^k$, which is calculated as follows:

$$s(r_{ij}) = F \cdot e^{-r_{ij}/L} - e^{-r_{ij}}. \quad (7)$$

Here, r_{ij} is the distance between $\mathbf{l}_j^k$ and $\mathbf{l}_i^k$, F represents the strength of attraction whereas L is the attractive length factor. It is assumed that F<1 and L>1 so that repulsion is stronger in a shorter-scale, while attraction is applied in a weaker and longer-scale. $\rho(\mathbf{l}_i^k, \mathbf{l}_j^k)$ is a function that calculates the dominance value of the best element between $\mathbf{l}_j^k$ and $\mathbf{l}_i^k$. In order to operate $\rho(\mathbf{l}_i^k, \mathbf{l}_j^k)$, all the individuals from $\mathbf{L}^k$ $(\{\mathbf{l}_1^k, \mathbf{l}_2^k, \ldots, \mathbf{l}_N^k\})$ are arranged in terms of their fitness values. Therefore, a rank is assigned to each element, so that the best individual obtains the rank 0 (zero) whereas the worst individual receives the rank N-1. Under such conditions, the function $\rho(\mathbf{l}_i^k, \mathbf{l}_j^k)$ is defined as follows:

$$\rho(\mathbf{l}_i^k,\mathbf{l}_j^k)=\begin{cases} e^{-\left(5\cdot \mathrm{rank}(\mathbf{l}_i^k)/N\right)} & \text{if } \mathrm{rank}(\mathbf{l}_i^k)<\mathrm{rank}(\mathbf{l}_j^k), \\ e^{-\left(5\cdot \mathrm{rank}(\mathbf{l}_j^k)/N\right)} & \text{if } \mathrm{rank}(\mathbf{l}_i^k)>\mathrm{rank}(\mathbf{l}_j^k), \end{cases} \quad (8)$$

where rank(*α*) delivers the rank of the *α*-element. According to Eq. 8, $\rho(\mathbf{l}_i^k,\mathbf{l}_j^k)$ gives a value within [0,1]. Fig. 1 shows the behavior of $\rho(\mathbf{l}_i^k,\mathbf{l}_j^k)$ considering 100 elements. In the Figure, it is assumed that $\mathbf{l}_i^k$ represents one of the 99 individuals with ranks among 0 and 98 whereas $\mathbf{l}_j^k$ is fixed to the worst individual (rank 99).

Then, the resultant force $\mathbf{S}_i^m$ on each element $\mathbf{l}_i^k$ is computed as the superposition of all of the pairwise interactions exerted on it:

$$\mathbf{S}_i^m = \sum_{\substack{j=1 \\ j\neq i}}^{N} \mathbf{s}_{ij}^m . \quad (9)$$

Finally, $\Delta\mathbf{l}_i$ is assumed similar to the social force experimented by $\mathbf{l}_i^k$ as the superposition of all of the pairwise reciprocal forces. Consequently, $\Delta\mathbf{l}_i$ is represented as follows:

$$\Delta\mathbf{l}_i = \mathbf{S}_i^m . \quad (10)$$

After calculating the new locations $\mathbf{P}(\{\mathbf{p}_1,\mathbf{p}_2,\ldots,\mathbf{p}_N\})$ of the population $\mathbf{L}^k(\{\mathbf{l}_1^k,\mathbf{l}_2^k,\ldots,\mathbf{l}_N^k\})$, the final locations $\mathbf{F}(\{\mathbf{f}_1,\mathbf{f}_2,\ldots,\mathbf{f}_N\})$ must be computed. This procedure can be summarized by the following formulation (in terms of a minimization problem):

$$\mathbf{f}_i=\begin{cases} \mathbf{p}_i & \text{if } f(\mathbf{p}_i)<f(\mathbf{l}_i^k), \\ \mathbf{l}_i^k & \text{otherwise.} \end{cases} \quad (11)$$

3.2 Social Operation (B)

The social operation is a discriminating operation which considers only to a subset $\mathbf{E}$ of the final positions $\mathbf{F}$ (where $\mathbf{E}\subseteq\mathbf{F}$). In the process first is necessary to order **F** in terms of their fitness values and collect the individuals in a temporal population $\mathbf{B}=\{\mathbf{b}_1,\mathbf{b}_2,\ldots,\mathbf{b}_N\}$. The individuals of **B** are arranged so that the best element is located in the first position $\mathbf{b}_1$ $\{b_{1,1},b_{1,2},\ldots,b_{1,n}\}$ whereas the worst individual is situated in the last location $\mathbf{b}_N$. Under such conditions, $\mathbf{E}$ is composed by the first *g* position of **B** (the best elements). Then, a subspace C_j is defined around each selected element $\mathbf{f}_j\in\mathbf{E}$. The size of C_j depends on the distance e_d which is determined as follows:

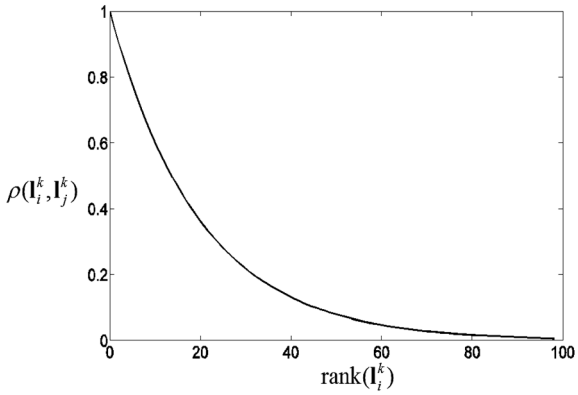

Fig. 1. Behavior of $\rho(\mathbf{l}_i^k,\mathbf{l}_j^k)$ considering 100 individuals

$$e_d=\frac{\sum_{q=1}^{n}\left(ub_q-lb_q\right)}{n}\cdot\beta, \quad (12)$$

where ub_q and lb_q are the upper and lower limits of the *q*-th dimension, n is the number of dimensions of the optimization problem, whereas $\beta\in[0,1]$ is a tuning factor. Therefore, the bounds of C_j are modeled as follows:

$$\begin{aligned} uss_j^q &= b_{j,q}+e_d, \\ lss_j^q &= b_{j,q}-e_d, \end{aligned} \quad (13)$$

where uss_j^q and lss_j^q are the upper and lower limits of the *q-th*-dimension for the subspace C_j, respectively. Once creating the subspace C_j in the neighborhood of the element $\mathbf{f}_j\in\mathbf{E}$, a set of *h* new elements $(\mathbf{M}_j^h=\{\mathbf{m}_j^1,\mathbf{m}_j^2,\ldots,\mathbf{m}_j^h\})$ are randomly produced within the limits defined by Eq. 13. Considering the *h* samples, the new

individual $\mathbf{l}_j^{k+1}$ of the next population $\mathbf{L}^{k+1}$ must be extracted. In order to select $\mathbf{l}_j^{k+1}$, the best element $\mathbf{m}_j^{best}$, in terms of fitness value from the *h* samples (where $\mathbf{m}_j^{best} \in \left[\mathbf{m}_j^1, \mathbf{m}_j^2, \ldots, \mathbf{m}_j^h\right]$), is examined. If $\mathbf{m}_j^{best}$ is better than $\mathbf{f}_j$ according to their fitness values, $\mathbf{l}_j^{k+1}$ is updated with $\mathbf{m}_j^{best}$, otherwise the position of $\mathbf{f}_j$ is assigned to $\mathbf{l}_j^{k+1}$. The elements of $\mathbf{F}$ that have not been considered by the procedure ($\mathbf{f}_w \notin \mathbf{E}$) transport their corresponding values to $\mathbf{L}^{k+1}$ without variation.

The social operation is used to exploit only favorable solutions. According to the social operation, inside each subspace C_j, *h* random samples are produced. Since the number of selected elements in each subspace is very small (typically $h < 4$), the use of this operator cannot be considered computational expensive.

4 Fractional-order Van der Pol Oscillator

The Van der Pol Oscillator model has been extensively studied as a complex example of non-linear system. It provides important models for a wide range of dynamic behaviors for several engineering applications [26, 27]. The classical integer-order Van der Pol Oscillator is described by a second-order non-linear differential equation as follows:

$$\begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & -\varepsilon(y_1^2(t)-1) \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}, \tag{14}$$

where ε is a control parameter that reflects the nonlinearity degree of the system. On the other hand, the fractional-order Van der Pol Oscillator model of order *q* is defined by the following formulation [28]:

$$\begin{aligned} {}_0D_t^{q_1} y_1(t) &= y_2(t), \\ {}_0D_t^{q_2} y_2(t) &= -y_1(t) - \varepsilon(y_1^2(t)-1) y_2(t). \end{aligned} \tag{15}$$

Considering the Grünwald-Letnikov approximation (see Eq. 5), the numerical solution for the fractional-order Van der Pol Oscillator is given by:

$$\begin{aligned} y_1(t_k) &= y_2(t_{k-1}) h^{q_1} - \sum_{j=1}^{k} c_j^{(q_1)} y_1(t_{k-j}), \\ y_2(t_k) &= (-y_1(t_k) - \varepsilon(y_1^2(t_k)-1) y_2(t_{k-1})) h^{q_2} - \sum_{j=1}^{k} c_j^{(q_2)} y_2(t_{k-j}). \end{aligned} \tag{16}$$

5 Problem Formulation

In the proposed approach, the identification process is considered as a multidimensional optimization problem. In the optimization process, the parameters of a new fractional-order chaotic system FOC_E are determined by using the LS method from the operation of the original fractional-order chaotic system FOC_O.

The idea is that FOC_E presents the best possible parametric affinity with FOC_O. Under such circumstances, the original fractional-order chaotic system FOC_O can be defined as follows:

$${}_aD_t^{\mathbf{q}} Y = F(\mathbf{Y}, \mathbf{Y_0}, \boldsymbol{\theta}), \tag{17}$$

here $\mathbf{Y} = [y_1, y_2, ..., y_m]^T$ denotes the state vector of the system, $\mathbf{Y_0}$ symbolizes the initial state vector, $\boldsymbol{\theta} = [\theta_1, \theta_2, ..., \theta_m]^T$ represents the original systematic parameter set, $\mathbf{q} = [q_1, q_2, ..., q_m]^T$ for $0 < q_i < 1 \quad (i \in [1, \ldots, m])$ corresponds to the fractional derivative orders and *F* is a generic non-linear function. On the other hand, the estimated fractional-order chaotic system FOC_E can be modeled as follows:

$${}_aD_t^{\hat{\mathbf{q}}} \hat{Y} = F(\hat{\mathbf{Y}}, \mathbf{Y}_0, \hat{\boldsymbol{\theta}}), \tag{18}$$

where $\hat{\mathbf{Y}}$, $\hat{\boldsymbol{\theta}}$ and $\hat{\mathbf{q}}$ denotes the estimated state system, the estimated systematic parameter vector and the estimated fractional orders, respectively. Since the goal is that FOC_E presents the best possible parametric affinity with FOC_O, the problem can be approached as an optimization problem described by the following formulation:

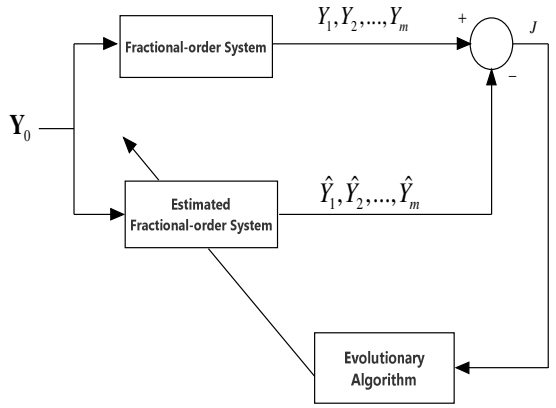

Fig. 2. Evolutionary Algorithm for Fractional-order System Parameter Estimation

$$\overline{\theta}, \overline{q} = \arg \min_{(\hat{\mathbf{Y}}, \hat{\mathbf{q}}) \in \Omega} (J(\theta, q)), \tag{19}$$

where $\overline{\theta}, \overline{q}$ denotes the best possible parametric values obtained by the optimization process, Ω symbolizes the search space admitted for parameters ($\hat{\mathbf{Y}}$ and $\hat{\mathbf{q}}$) whereas *J* represents the objective function that evaluates the parametric affinity between FOC_O and FOC_E. This affinity can be computed as follows:

$$J(\theta, q) = \frac{1}{M} \sum_{k=1}^{M} (\mathbf{Y}(k) - \hat{\mathbf{Y}}(k))^2, \tag{20}$$

where $\mathbf{Y}(k)$ and $\tilde{\mathbf{Y}}(k)$ represent the state values produced by the original and estimated systems, respectively. On the other hand, *k* denotes the sampling time point and *M* represents the length of data used for parameter estimation. According to the optimization problem formulated in Eq. 19, the parameter identification can be achieved by searching suitable values of $\hat{\mathbf{Y}}$ and $\hat{\mathbf{q}}$ within the searching space Ω, such that the objective function has been minimized.

Fig. 2 shows the graphic representation of the identification process. Since the fractional-order Van der Pol oscillator has been chosen to test the performance of the proposed approach, the fractional-order system maintain two different fractional derivative orders $\mathbf{q} = [q_1, q_2]^T$ (*m*=2) and one systematic parameter ε.

6 Experimental Results

To verify the effectiveness and robustness of the proposed approach, the fractional-order Van der Pol oscillator is chosen to test its performance. The simulations has been conducted by using MATLAB (Version 7.1, MathWorks, Natick, MA, USA) on an Intel(R) Core(TM) i7-3470 CPU, 3.2 GHz with 4 GB of RAM. In order to calculate the objective function, the number of samples is set as 300 and the step size is 0.01.

In this section, the results of the LS algorithm have been compared to those produced by the Genetic Algorithms (GA) [17], Particle Swarm Optimization (PSO) method [20], the Differential Evolution (DE) [19], and the proposed method. In all comparisons, the population has been set to 40 (*N*=40) individuals. The maximum iteration number for all functions has been set to 100. Such stop criterion has been selected to maintain compatibility to similar works reported in the literature [16].

The parameter setting for each of the algorithms in the comparison is described as follows:

1. GA: The population size has been set to 70, the crossover probability with 0.55, the mutation probability with 0.10 and number of elite individuals with 2. The roulette wheel selection and the 1-point crossover are applied.
2. PSO: In the method, $c_1 = c_2 = 2$ whereas the inertia factor (ω) is decreased linearly from 0.9 to 0.2.
3. DE: The DE/Rand/1 scheme has been employed. The parameter settings follow the instructions suggested in [30]. The crossover probability is *CR*=0.9 whereas the weighting factor is *F*=0.8.
4. In LS, *F* and L are set to 0.6 and *L*, respectively. Similarly, *g* is fixed to 20 (*N*/2), *h*=2, $\beta = 0.6$ whereas *gen* and *N* are set to 1000 and 40, respectively. Once such parameters have been experimentally determined, they are considered for all experiments in this section.

In the experiments, the fractional-order Van der Pol Oscillator to be estimated has been

Table 1. Simulation result of the algorithms GA, PSO, DE and LS

	Parameter	GA	PSO	DE	LS
BEST	ε	0.9021	0.9152	0.9632	0.9978
	$\frac{\lvert\varepsilon-1\rvert}{1}$	0.0979	0.0848	0.0368	0.0022
	q_1	1.3001	1.2810	1.2210	1.2005
	$\frac{\lvert q_1-1.2\rvert}{1.2}$	0.0834	0.0675	0.0175	0.0004
	q_2	0.8702	0.8871	0.8229	0.8011
	$\frac{\lvert q_2-0.8\rvert}{0.8}$	0.0877	0.1088	0.0286	0.0013
WORST	ε	0.1731	0.1176	0.3732	0.7198
	$\frac{\lvert\varepsilon-1\rvert}{1}$	0.8269	0.8824	0.6268	0.2802
	q_1	2.1065	0.3643	1.8532	1.3075
	$\frac{\lvert q_1-1.2\rvert}{1.2}$	0.7554	0.6964	0.5443	0.0895
	q_2	0.1219	1.7643	1.2154	0.9101
	$\frac{\lvert q_2-0.8\rvert}{0.8}$	0.8476	1.2053	0.5192	0.1376
MEAN	ε	1.2131	1.2052	1.1701	1.0186
	$\frac{\lvert\varepsilon-1\rvert}{1}$	0.2131	0.2052	0.1701	0.0186
	q_1	0.9032	1.0974	1.3421	1.2654
	$\frac{\lvert q_1-1.2\rvert}{1.2}$	0.2473	0.0855	0.1186	0.0545
	q_2	0.9052	0.7229	0.7832	0.8089
	$\frac{\lvert q_2-0.8\rvert}{0.8}$	0.1315	0.0963	0.0210	0.0111

configured such that $q_1 = 1.2$, $q_2 = 0.8$ and $\varepsilon = 1$. Similarly, the initial state has been set to [0.02- 0.2].

The statistical results of the best, the mean and the worst estimated parameters with the corresponding relative error values over 100 independent runs are shown in Table 1. From Table 1, it can be easily seen that the estimated values generated by the proposed LS algorithm are closer to the actual parameter values, which means that it is more accurate than the standard GA, PSO and DE algorithms.

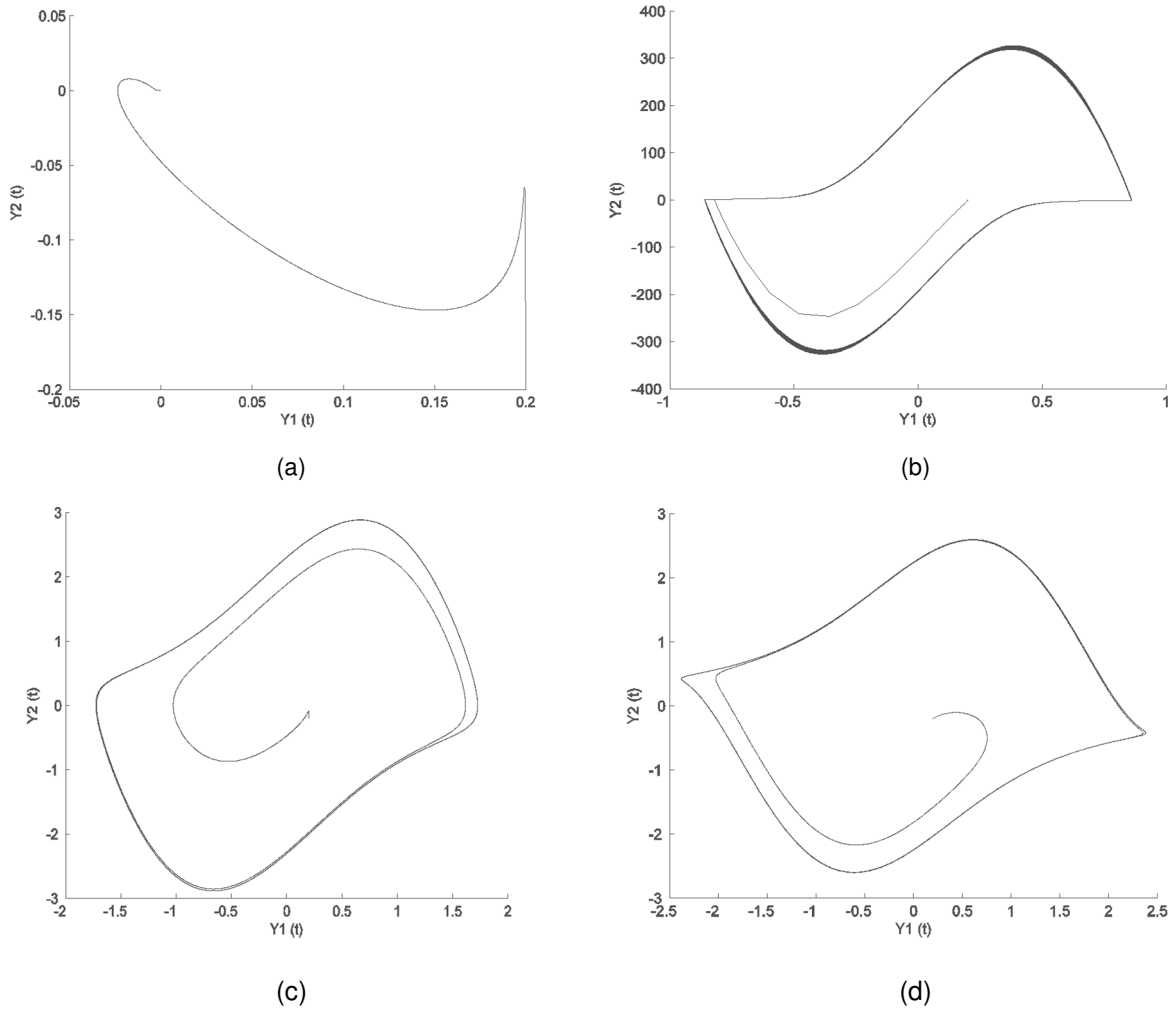

Fig. 3. Phase diagrams of the Van der Pol Oscillator by using the mean estimated parameters for (a) GA, (b) PSO, (c) DE and (d) the proposed approach

Likewise, it can also be clearly found that the relative error values obtained by the LS algorithm are all smaller than those of the standard GA, PSO and DE algorithms, which can also prove that the LS algorithm has a higher performance in terms of accuracy. Therefore, the estimated parameters can be closer to the true values than the GA, PSO and DE algorithms.

With this evidence, it can be concluded that the LS algorithm can more efficiently identify a fractional-order systems than the other algorithms used in the comparisons. Therefore, the estimated parameters can be closer to the true values than the GA, PSO and DE algorithms.

With this evidence, it can be concluded that the LS algorithm can more efficiently identify a fractional-order systems than the other algorithms used in the comparisons. In order to show the proficiency, of the proposed approach, Figure 3 presents the phase diagrams of the Van der Pol Oscillator by using the mean estimated parameters for each method.

The convergence curves of the parameters and fitness values estimated by the set of

Table 2. Average best solution obtained by each algorithm GA, PSO, DE and LS

GA	PSO	DE	LS
0.2251	0.2016	0.0982	0.0126

Table 3. *p*-values produced by Wilcoxon's test that compares LS vs GA, LS vs PSO and DE over the "average best-solution" values from Table 3

	***p*-values**
LS vs. GA	0.00021
LS vs. PSO	0.00098
LS vs. DE	0.00123

Table 4. Average Number of Function Evaluations (NFE) obtained by each algorithm GA, PSO, DE and LS

GA	PSO	DE	LS
97,378	95,366	68,446	55,933

algorithms are shown in Figures 4-6 in a single execution. From Figures 4-6, it can be clearly observed that convergence processes of the parameters and fitness values of LS algorithm are better than other algorithms. Additionally, the estimated parameter values obtained by the LS algorithm fall faster than the other algorithms.

Furthermore, Table 2 shows the average best solution obtained by each algorithm. The average best solution (ABS) expresses the average value of the best function evaluations that have been obtained from 100 independent executions. A non-parametric statistical significance test known as the Wilcoxon's rank sum test for independent samples [30, 31] has been conducted with an 5% significance level, over the "average best-solution" data of Table 2.

Table 3 reports the *p*-values produced by Wilcoxon's test for the pair-wise comparison of the "average best-solution" of two groups. Such groups are formed by LS vs. GA, LS vs. PSO and LS vs. DE. As a null hypothesis, it is assumed that there is no significant difference between mean values of the two algorithms.

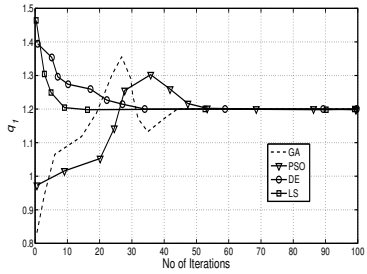

Fig. 4. Estimated parameter q_1 (fractional order).

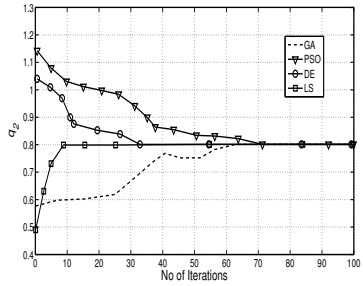

Fig. 5. Estimated parameter q_2 (fractional order).

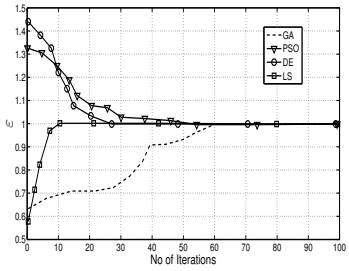

Fig. 6. Estimated systematic parameter ε .

The alternative hypothesis considers a significant difference between the "average best-solution" values of both approaches. All *p*-values reported in the table are less than 0.05 (5% significance level) which is a strong evidence against the null hypothesis, indicating that the LS results are statistically significant and that it has not occurred by coincidence (i.e. due to the normal noise contained in the process).

On the other hand, the time spent by all methods in the estimation of the parameter set is evaluated. Evolutionary methods are, in general, complex pieces of software with several operators and stochastic branches. Therefore, it is difficult to conduct a complexity analysis from a

deterministic perspective. Therefore, the number of function evaluations (*NFE*) is commonly used in order to evaluate the computational effort. Table 4 presents the *NFE* values obtained by each approach. The Table reports the averaged value considering 100 different executions. From the table, it can be seen that the proposed LS method present the best performance.

7 Conclusions

Due to its multiple applications, parameter identification for fractional-order chaotic systems has attracted the interests of several research communities. In the identification, the parameter estimation process is transformed into a multidimensional optimization problem where fractional orders, as well as functional parameters of the chaotic system are considered the decision variables. Under this approach, the complexity of fractional-order chaotic systems tends to produce multimodal error surfaces for which their cost functions are significantly difficult to minimize. Several algorithms based on evolutionary computation principles have been successfully applied to identify the parameters of fractional-order chaotic systems. However, most of them maintain an important limitation, they frequently obtain sub-optimal results as a consequence of an inappropriate balance between exploration and exploitation in their search strategies.

In this paper, an algorithm for parameter identification of fractional-order chaotic systems has been presented. In order to determine the parameters, the proposed method uses a novel evolutionary method called Locust Search (LS) [R1] which is based on the behavior of swarms of locusts. In the proposed algorithm, individuals emulate a group of locusts which interact to each other based on the biological laws of the cooperative swarm.

The algorithm considers two different behaviors: solitary and social. Depending on the behavior, each individual is conducted by a set of evolutionary operators which mimics different cooperative conducts that are typically found in the swarm. Different to most of existent evolutionary algorithms, the behavioral model in the proposed approach explicitly avoids the concentration of individuals in the current best positions. Such fact allows to avoid critical flaws such as the premature convergence to sub-optimal solutions and the incorrect exploration-exploitation balance.

In order to test the proficiency and robustness of the presented method, it has been compared to other algorithms based on evolutionary principles such as GA, PSO and DE. The comparison examines the identification of the fractional Van der Pol Oscillator. The results show a high performance of the proposed descriptor in terms of precision and robustness.

References

1. **Das, S. (2011).** *Observation of fractional calculus in physical system description*. New York: Springer; pp. 101–56. DOI:10.1007/978-3-642-20545-3_3.
2. **Arena, P., Caponetto, R., Fortuna, L., & Porto, D. (2000).** *Nonlinear Order Circuits and Systems An Introduction*. World Scientific, Singapore, Singapore.
3. **Rivero, M., Rogosin, S.V., Tenreiro-Machado, J.A., & Trujillo, J.J. (2013).** Stability of fractional order systems. *Math. Problem Eng.* DOI:/10.1155/2013/356215.
4. **Diethelm, K. (2011).** An efficient parallel algorithm for the numerical solution of fractional differential equations. *Fract. Calc. Appl. Anal.,* Vol. 14, No. 3, pp. 475–490. DOI: https://doi.org/10.2478/s13540-011-0029-1.
5. **Diethelm, K. & Ford, N.J. (2002).** Analysis of fractional differential equations. *J. Math. Anal. Appl.,* Vol. 265, No. 2, pp. 229–248. DOI: 10.1006/jmaa.2000.7194.
6. **Kilbas, A.A.A., Srivastava, H.M., & Trujillo, J.J. (2006).** *Theory and Applications of Fractional Differential Equations*. Elsevier Science, Netherlands.
7. **Podlubny, I. (1998).** *Fractional Differential Equations*. Academic Press, USA.
8. **Miller, K.S. & Ross, B. (1993).** *An Introduction to the Fractional Calculus and Fractional Differential Equations*. Wiley, New York.
9. **Wei Hu, Yongguang Yu, & Shuo Zhang (2015).** A hybrid artificial bee colony algorithm for parameter identification of uncertain fractional-order chaotic systems. *Nonlinear Dynamic*,

Vol. 82, No. 3, pp 1441–1456. DOI: 10.1007/ s11071-015-2251-6.

10. **Yongguang Yu, Han-Xiong Li, Sha Wang, & Junzhi Yu (2009).** Dynamic analysis of a fractional-order Lorenz chaotic system. *Chaos, Solitons and Fractals,* Vol. 42, No. 2, pp. 1181–1189. DOI: 10.1016/j.chaos.2009. 03.016.
11. **Poinot, T. & Trigeassou, J.C. (2004).** Identification of fractional systems using an output error technique. *Nonlinear Dyn*, Vol. 38, No. 1, pp. 133–54. DOI: 10.1007/s11071-004-3751-y.
12. **Nazarian, P., Haeri, M., & Tavazoei, M.S. (2010).** Identifiability of fractional order systems using input output frequency contents. *ISA Trans,* Vol. 49, No. 2, pp. 207–14. DOI: 10.1016/j.isatra. 2009.11.007.
13. **Saha-Ray, S. (2012).** On Haar wavelet operational matrix of general order and its application for the numerical solution of fractional Bagley Torvik equation. *App. Math Comp*, Vol. 218, No. 9, pp. 5239–5248. DOI: 10.1016/ j.amc.2011.11.007.
14. **Kerschen, G., Worden, K., Vakakis, A.F., & Golinval, J.C. (2006).** Past, present and future of nonlinear system identification in structural dynamics. *Mechanical Systems and Signal Processing,* Vol. 20, No. 3, pp. 505–592. DOI: 10.1016/j.ymssp.2005.04.008.
15. **Quaranta, G., Monti, G., & Marano, C. (2010).** Parameters identification of Van der Pol–Duffing oscillators via particle swarm optimization and differential evolution. *Mechanical Systems and Signal Processing,* Vol. 24, No. 7, pp. 2076–2095. DOI: 10.1016/j.ymssp.2010.04.006.
16. **Shengxi Zhou, Junyi Cao, & Yangquan Chen (2013).** Genetic Algorithm-Based Identification of Fractional-Order Systems. *Entropy,* Vol. 15, No. 5, pp. 1624–1642. DOI:10.3390/e15051624.
17. **Wei Hu, Yongguang Yu, & Sha Wang (2015).** Parameters Estimation of Uncertain Fractional-Order Chaotic Systems via a Modified Artificial Bee Colony Algorithm. *Entropy,* Vol. 17, pp. 692–709; DOI:10.3390/e17020692.
18. **Fei Gao, Xue-jing Lee, Feng-xia Fei, Heng-qing Tong, Yan-fang Deng, & Hua-ling Zhao (2014).** Identification time-delayed fractional order chaos with functional extrema model via differential evolution. *Expert Systems with Applications*, Vol. 41, No. 4, pp. 1601–1608. DOI: 10.1016/ j.eswa.2013.08.057.
19. **Defeng Wu, Zi Ma, Aiguo Li, & Quanmin Zhu (2011).** Identification for fractional order rational models based on particle swarm optimization. *J. of Computer Applications in Technology*, Vol. 41, No. 1/2, pp.53–59. DOI:10.1504/ IJCAT.2011. 042232.
20. **Tan, K.C., Chiam, S.C., Mamun, A.A., & Goh, C.K. (2009).** Balancing exploration and exploitation with adaptive variation for evolutionary multi-objective optimization. *European Journal of Operational Research*, Vol. 197, No. 2, pp. 701–713. DOI:10.1016/j.ejor.2008.07.025.
21. **Chen, G., Low, C.P., & Yang, Z. (2009).** Preserving and exploiting genetic diversity in evolutionary programming algorithms. *IEEE Transactions on Evolutionary Computation,* Vol. 13, No. 3, pp. 661–673. DOI: 10.1109/ TEVC.2008.2011742.
22. **Cuevas, E., González, A., Fausto, F., Zaldívar, D., & Pérez-Cisneros, M. (2015).** Multithreshold Segmentation by Using an Algorithm Based on the Behavior of Locust Swarms. *Mathematical Problems in Engineering*, DOI:10.1155/2015/ 805357.
23. **Podlubny, I. (1998).** *Fractional Differential Equations.* Acad. Press.
24. **Miller, K.S. & Ross, B. (1993).** *An Introduction to the Fractional Calculus and Fractional Differential Equations.* Wiley.
25. **Dorkal, L. (1994).** Numerical Models for the Simulation of the Fractional-Order Control Systems. *Num. Model. Simul. Fractional-Order Control Syst*.
26. **Barbosa, R.S., Machado, J.T., Vinagre, B.M., & Calderon, J. (2007)**. Analysis of the Van der Pol Oscillator Containing Derivatives of Fractional Order. *J. Vib. Control*, Vol. 13, No. 9-10, pp. 1291–1301.
27. **Cartwright, J., Eguiluz, V., Hernandez-Garcia E., & Piro, O. (1999)**. Dynamics of elastic excitable media. *J. Bifur. Chaos Appl. Sci. Eng.*, pp. 2197–2202. DOI:10.1142/S0218127499 001620.
28. **Cuevas, E., Zaldivar, D., Pérez-Cisneros, M., & Ramírez-Ortegón, M. (2011)**. Circle detection using discrete differential evolution optimization. *Pattern Analysis and Applications*, Vol. 14, No. 1, pp. 93–107. DOI: 10.1007/s10044-010-0183-9.
29. **Wilcoxon, F. (1945).** Individual comparisons by ranking methods. *Biometrics*, Vol. 1, No. 6, pp. 80–83. DOI: 10.2307/3001968.
30. **Garcia, S., Molina, D., Lozano, M., & Herrera, F. (2008).** A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behavior: a case study on the CEC'2005 Special session on real parameter optimization. *J Heurist.* DOI: 10.1007/s10732-008-9080-4.

31. Cuevas, E., Zaldivar, D., & Perez, M. (2016). Automatic Segmentation by Using an Algorithm Based on the Behavior of Locust Swarms. *Applications of Evolutionary Computation in Image Processing and Pattern Recognition*, Vol. 100 of the series Intelligent Systems Reference Library, pp. 229–269. DOI: 10.1007/978-3-319-26462-2_10.

32. Cuevas, E., González, A., Zaldívar, D., & Pérez-Cisneros, M. (2015). An optimization algorithm based on the behavior of locust swarms. *International Journal of Bio-Inspired Computation*, Vol. 7, No. 6, pp. 402–407. DOI: 10.1504/IJBIC.2015.073178.

Article received on 02/02/2016; accepted on 07/02/2017. Corresponding author is Erik Cuevas.