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 \( \frac{d^q}{dt^q} \) takes both the fractional derivative and the fractional integral in a single expression which is defined as:

\[
\frac{d^q}{dt^q} = \begin{cases} 
\frac{d^q}{dt^q}, & q > 0, \\
1, & q = 0, \\
\int_a^t (d\tau)^q, & q < 0,
\end{cases}
\]  

(1)

where \( a \) and \( t \) represents the operation bounds whereas \( q \in \mathbb{R} \). 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:

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

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

\[
(k - L_m) D^q f(t) = h^{-q} \sum_{j=0}^{k} (-1)^j \binom{q}{j} f(t - j).
\] (3)

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

\[
c^{(q)}_0 = 1, \quad c^{(q)}_j = \left(1 - \frac{1 + q}{j}\right)c^{(q)}_{j-1}.
\] (4)

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

\[
y(t) = f(y(t), t)h^q - \sum_{j=1}^{\infty} c^{(q)}_j y(t_{j-1}).
\] (5)

3 Locust Search (LS) Algorithm

In the operation of LS [23], a population \( \mathbf{L} \) \( ((\mathbf{l}_1^i, \mathbf{l}_2^i, \ldots, \mathbf{l}_n^i)) \) of \( N \) locusts (individuals) is processed from the initial stage \((k=0)\) to a total \( \text{gen} \) number iterations \((k=\text{gen})\).

Each individual \( \mathbf{l}_i^k \) \( (i \in [1, \ldots, N]) \) symbolizes an \( n \)-dimensional vector \( \{l_{1,i}, l_{2,i}, \ldots, l_{n,i}\} \) where each dimension represents a domain variable of the optimization problem to be solved. The set of variables represents the valid search space \( S = \{l_i \in \mathbb{R}^n | lb_d \leq l_{i,d} \leq ub_d\} \), 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 \( \Delta \mathbf{l}_i \) 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 \( \mathbf{l}_i^k \) and \( \mathbf{l}_i^j \) is computed by considering the following model:

\[
s_{y_i}^n = \rho(\mathbf{l}_i^k, \mathbf{l}_i^j) \cdot s(r_{y_i}) \cdot d_{y_i} + rand(1, -1),
\] (6)

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

\[
s(r_{y_i}) = F \cdot e^{-r_{y_i}} - e^{-r_{y_i}}.
\] (7)

Here, \( r_{y_i} \) is the distance between \( \mathbf{l}_i^k \) and \( \mathbf{l}_i^j \), \( F \) 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. \( \rho(\mathbf{l}_i^k, \mathbf{l}_i^j) \) is a function that calculates the dominance value of the best element between \( \mathbf{l}_i^k \) and \( \mathbf{l}_i^j \). In order to operate \( \rho(\mathbf{l}_i^k, \mathbf{l}_i^j) \), all the individuals from \( \mathbf{L} \) \( ((\mathbf{l}_1^k, \mathbf{l}_2^k, \ldots, \mathbf{l}_n^k)) \) 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 \( \rho(\mathbf{l}_i^k, \mathbf{l}_i^j) \) is defined as follows:
where \( \text{rank}(\alpha) \) delivers the rank of the \( \alpha \)-element. According to Eq. 8, \( \rho(t^*_i,t^*_j) \) gives a value within \([0,1]\). Fig. 1 shows the behavior of \( \rho(t^*_i,t^*_j) \) considering 100 elements. In the Figure, it is assumed that \( t^*_i \) represents one of the 99 individuals with ranks among 0 and 98 whereas \( t^*_i \) is fixed to the worst individual (rank 99).

Then, the resultant force \( S^*_i \) on each element \( t^*_i \) is computed as the superposition of all of the pairwise interactions exerted on it:

\[
S^*_i = \sum_{j=1}^{n} s^*_{ij}.
\]

Finally, \( \Delta t_i \) is assumed similar to the social force experimented by \( t^*_i \) as the superposition of all of the pairwise reciprocal forces. Consequently, \( \Delta t_i \) is represented as follows:

\[
\Delta t_i = S^*_i.
\]

After calculating the new locations \( P(t_1, t_2, \ldots, t_N) \) of the population \( L^* (t^*_1, t^*_2, \ldots, t^*_N) \), the final locations \( F(t_1, t_2, \ldots, t_N) \) must be computed. This procedure can be summarized by the following formulation (in terms of a minimization problem):

\[
f_j = \begin{cases} 
  p_j & \text{if } f(p_j) < f(t^*_j), \\
  t^*_j & \text{otherwise.}
\end{cases}
\]

### 3.2 Social Operation (B)

The social operation is a discriminating operation which considers only to a subset \( E \) of the final positions \( F \) (where \( E \subseteq F \)). In the process first is necessary to order \( F \) in terms of their fitness values and collect the individuals in a temporal population \( B = \{b_1, b_2, \ldots, b_n\} \). The individuals of \( B \) are arranged so that the best element is located in the first position \( b_1 \) \( \{b_1, b_2, \ldots, b_n\} \) whereas the worst individual is situated in the last location \( b_n \). Under such conditions, \( E \) is composed by the first \( g \) position of \( B \) (the best elements). Then, a subspace \( C_j \) is defined around each selected element \( t^*_j \in E \). The size of \( C_j \) depends on the distance \( d_e \) which is determined as follows:

\[
e_d = \frac{\sum_{q=1}^{n} (u_b_q - l_b_q)}{n} \cdot \beta,
\]

where \( u_b_q \) and \( l_b_q \) 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:

\[
\text{uss}_j^q = b_{j,q} + e_d, \\
\text{lss}_j^q = b_{j,q} - e_d,
\]

where \( \text{uss}_j^q \) and \( \text{lss}_j^q \) 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 \( t^*_j \in E \), a set of \( h \) new elements \( \{M^j_1, M^j_2, \ldots, M^j_h\} \) are randomly produced within the limits defined by Eq. 13. Considering the \( h \) samples, the new
individual \( l_{k+1} \) of the next population \( L_{k+1} \) must be extracted. In order to select \( l_{k+1} \), the best element \( m_{j}^{\text{best}} \), in terms of fitness value from the \( h \) samples (where \( m_{j}^{\text{best}} \in \left[ m_{j}^{1}, m_{j}^{2}, \ldots, m_{j}^{h} \right] \)), is examined. If \( m_{j}^{\text{best}} \) is better than \( f_{i} \), according to their fitness values, \( l_{k+1} \) is updated with \( m_{j}^{\text{best}} \), otherwise the position of \( f_{i} \) is assigned to \( l_{k+1} \). The elements of \( F \) that have not been considered by the procedure \( (f_{i} \notin E) \) transport their corresponding values to \( L_{k+1} \) 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 nonlinear 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:

\[
\begin{bmatrix}
y_{1} \\
y_{2}
\end{bmatrix} = \begin{bmatrix}
0 & 1 \\
-1 & -\varepsilon(y_{1}^{2}(t) - 1)
\end{bmatrix} \begin{bmatrix}
y_{1} \\
y_{2}
\end{bmatrix},
\]

where \( \varepsilon \) 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]:

\[
\begin{align*}
a D_{t}^{q} y_{1}(t) &= y_{2}(t), \\
a D_{t}^{q} y_{2}(t) &= -y_{1}(t) - \varepsilon(y_{1}^{2}(t) - 1)y_{2}(t).
\end{align*}
\]

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

\[
\begin{align*}
y_{1}(t) &= y_{2}(t_{k})h^{q} - \sum_{j=0}^{\infty} c_{j}^{(n)}y_{1}(t_{k+j}), \\
y_{2}(t) &= (-y_{1}(t) - \varepsilon(y_{1}^{2}(t) - 1)y_{2}(t_{k})h^{q} - \sum_{j=0}^{\infty} c_{j}^{(n)}y_{2}(t_{k+j})).
\end{align*}
\]

### 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:

\[
a D_{t}^{q} Y = F(Y, \theta_{o}),
\]

where \( Y = [y_{1}, y_{2}, \ldots, y_{m}]^{T} \) denotes the state vector of the system, \( \theta_{o} \) symbolizes the initial state vector, \( \theta = \left[ \theta_{1}, \theta_{2}, \ldots, \theta_{m} \right]^{T} \) represents the original systematic parameter set, \( q_{i} = [q_{i1}, q_{i2}, \ldots, q_{im}]^{T} \) for \( 0 < q_{i1} < 1 \) \( (i \in [1, m]) \) corresponds to the fractional derivative orders and \( F \) is a generic nonlinear function. On the other hand, the estimated fractional-order chaotic system \( FOC_{E} \) can be modeled as follows:

\[
a D_{t}^{q} \hat{Y} = F(\hat{Y}, \hat{\theta}_{o}),
\]

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

\[
J(\theta, q) = \frac{1}{M} \sum_{i=1}^{M} (Y(k) - \hat{Y}(k))^2,
\]

where \( Y(k) \) and \( \hat{Y}(k) \) 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 \( \hat{Y} \) and \( \hat{q} \) within the
searching space \( \Omega \), 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 \( q = [q_1, q_2]^T \) \( (m=2) \)
and one systematic parameter \( \varepsilon \).

### 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
\( N=40 \) 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 \( (\omega) \) 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
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.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>GA</th>
<th>PSO</th>
<th>DE</th>
<th>LS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BEST</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\varepsilon$</td>
<td>0.9021</td>
<td>0.9152</td>
<td>0.9632</td>
<td>0.9978</td>
</tr>
<tr>
<td>$</td>
<td>\varepsilon - 1</td>
<td>/1$</td>
<td>0.0979</td>
<td>0.0848</td>
</tr>
<tr>
<td>$q_1$</td>
<td>1.3001</td>
<td>1.2810</td>
<td>1.2210</td>
<td>1.2005</td>
</tr>
<tr>
<td>$</td>
<td>q_1 - 1.2</td>
<td>/1.2$</td>
<td>0.0834</td>
<td>0.0675</td>
</tr>
<tr>
<td>$q_2$</td>
<td>0.8702</td>
<td>0.8871</td>
<td>0.8229</td>
<td>0.8011</td>
</tr>
<tr>
<td>$</td>
<td>q_2 - 0.8</td>
<td>/0.8$</td>
<td>0.0877</td>
<td>0.1088</td>
</tr>
<tr>
<td><strong>WORST</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\varepsilon$</td>
<td>0.1731</td>
<td>0.1176</td>
<td>0.3732</td>
<td>0.7198</td>
</tr>
<tr>
<td>$</td>
<td>\varepsilon - 1</td>
<td>/1$</td>
<td>0.8269</td>
<td>0.8824</td>
</tr>
<tr>
<td>$q_1$</td>
<td>2.1065</td>
<td>0.3643</td>
<td>1.8532</td>
<td>1.3075</td>
</tr>
<tr>
<td>$</td>
<td>q_1 - 1.2</td>
<td>/1.2$</td>
<td>0.7554</td>
<td>0.6964</td>
</tr>
<tr>
<td>$q_2$</td>
<td>0.1219</td>
<td>1.7643</td>
<td>1.2154</td>
<td>0.9101</td>
</tr>
<tr>
<td>$</td>
<td>q_2 - 0.8</td>
<td>/0.8$</td>
<td>0.8476</td>
<td>1.2053</td>
</tr>
<tr>
<td><strong>MEAN</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\varepsilon$</td>
<td>1.2131</td>
<td>1.2052</td>
<td>1.1701</td>
<td>1.0186</td>
</tr>
<tr>
<td>$</td>
<td>\varepsilon - 1</td>
<td>/1$</td>
<td>0.2131</td>
<td>0.2052</td>
</tr>
<tr>
<td>$q_1$</td>
<td>0.9032</td>
<td>1.0974</td>
<td>1.3421</td>
<td>1.2654</td>
</tr>
<tr>
<td>$</td>
<td>q_1 - 1.2</td>
<td>/1.2$</td>
<td>0.2473</td>
<td>0.0855</td>
</tr>
<tr>
<td>$q_2$</td>
<td>0.9052</td>
<td>0.7229</td>
<td>0.7832</td>
<td>0.8089</td>
</tr>
<tr>
<td>$</td>
<td>q_2 - 0.8</td>
<td>/0.8$</td>
<td>0.1315</td>
<td>0.0963</td>
</tr>
</tbody>
</table>

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

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

<table>
<thead>
<tr>
<th></th>
<th>GA</th>
<th>PSO</th>
<th>DE</th>
<th>LS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.2251</td>
<td>0.2016</td>
<td>0.0982</td>
<td>0.0126</td>
</tr>
</tbody>
</table>

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 2

<table>
<thead>
<tr>
<th>p-values</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>LS vs. GA</td>
<td>0.00021</td>
</tr>
<tr>
<td>LS vs. PSO</td>
<td>0.00098</td>
</tr>
<tr>
<td>LS vs. DE</td>
<td>0.00123</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th></th>
<th>GA</th>
<th>PSO</th>
<th>DE</th>
<th>LS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>97,378</td>
<td>95,366</td>
<td>68,446</td>
<td>55,933</td>
</tr>
</tbody>
</table>

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) 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


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