Exploring sensitive dependence and transitivity to optimize travel time in chaotic systems

S H S Santana¹, E E N Macau¹, O C Winter² and L F A França¹

¹Instituto Nacional de Pesquisas Espaciais - INPE, Avenida dos Astronautas, 1758 - CEP 12227-010, São José dos Campos, SP, Brazil
²Universidade Estadual Paulista - UNESP, Grupo de Dinâmica Orbital e Planetologia, Avenida Abílio Sampaio, 333 - CEP 12516-410, Guaratinguetá, SP, Brazil
E-mail: saymonhss@gmail.com

Abstract. Transitivity and sensitive dependence on initial conditions are the main characteristics of chaotic behavior. The latter one can be exploited so that small controlled perturbations in system parameters may imply a faster transfer in time from a desired start point to a neighborhood of a desired final state. In this study three targeting approaches are evaluated: The first one uses a geometric approach to find the proper perturbation which allows a faster transfer between two desired points; The second, an evolutionary algorithm called GEO (Generalized External Optimization), is adapted to search for optimized orbits; The third one, uses successive perturbations along the path in order to direct the orbits to the final desired point in a short time interval. These three methods are evaluated regarding performance and implementation complexity.

1. Introduction

Since the 18th century science has expended efforts to understand the behavior of nonlinear systems. Several systems in nature, such as the n-body problem in celestial mechanics, exhibit chaotic behavior.

Although notable scientists have worked to understand the behavior of systems of celestial mechanics, only in 1890, Poincaré showed that in the presence of homoclinic intersections, there is a sensitive dependence on initial conditions.

Chaotic systems are sensitive to initial conditions. This property, may be used in order to optimize the transport between two desired points of the system. Once the future state of a chaotic system may be substantially changed by small perturbations, as will be shown in our results, is possible to reduce the travel time of a path if a suitable perturbation is applied. This procedure, often called targeting, consists of finding a nearby short path using only available dynamics. In references [6] [4] [5] one may see the applicability of this method in orbital dynamics.

Our study focuses on trajectories in a Hénon Map [3], which takes a point \((x_n, y_n)\) in the plane and maps it to a new point \((x_{n+1}, y_{n+1})\).

\[
x_{n+1} = 1 - \alpha x_n^2 + y_n \quad (1)
\]
\[
y_{n+1} = \beta x_n \quad (2)
\]
Figure 1. Surface of section from a three dimensional flow

For $\alpha = 1.4$ and $\beta = 0.3$ the map presents a chaotic behavior.

In this work, we analyze trajectories starting from an initial state $(x_I, y_I)$ (called source-point) to a final state $(x_F, y_F)$ (target point). In order to reduce the time spent in that path, three targeting approaches were implemented. The first one [7] uses a geometric approach to find the proper perturbation which allows a quick transfer between two desired points; The second, [2] an evolutionary algorithm called GEO (Generalized External Optimization), is adapted to search for optimized orbits; The third one [1] uses successive perturbations along the path in order to direct the orbits to their final desired point in a short time interval. Each one of these methods are explained in the following sections.

2. Geometric Approach

Consider a three-dimensional continuous-time dynamical system $\frac{dX}{dt} = F(X)$ and employing a surface of section whose coordinates are denoted by $\xi$ and the Poincaré map $\xi_{n+1} = f(\xi_n, \alpha)$, where $f$ is an invertible map and $\alpha$ a system parameter.

Suppose one wish to go from a source point $X_S$ to a small region around a target point $X_T$ (as shown in figure 1).

We assume that the system’s parameter $\alpha$ is available for adjustment at each iterate, such that $\alpha_n = \bar{\alpha} + \delta_n$, where $\bar{\alpha}$ is a nominal value of $\alpha$ and $\delta_n$ denotes a small deviation from $\bar{\alpha}$.

After one iteration of the return map, the change of state $\delta_1$ due to a small perturbation $\delta_1$ relative to the point $f(\xi_s, \bar{\alpha})$ is given by:

$$\delta_1 \approx \delta_1 \left. \frac{\partial f(\xi, \alpha)}{\partial \alpha} \right|_{\bar{\alpha}}$$

where $\delta_1$ varies through a small interval $|\delta_1| < \delta_*$.  

The method consists in iterating the segment $\delta_1$ and the region $\epsilon_T$ around the target, forward and backward respectively, using $\alpha_n = \bar{\alpha}$, until it is detected the intersection of the images of $\delta_1$ and $\epsilon_T$. Thus, a point in $\delta_1$, near to the target point, which is mapped to the target region $\epsilon_T$ in $n_1 + n_2$ iterations, is found, where $n_1$ ($n_2$) is the number of iterations of the segment $\delta_1$ (region $\epsilon_T$).

In pratice, it is not possible to iterate either the segment $\delta_1$ or the region $\epsilon_T$. Actually, we iterate discrete approximations to them (See figure 2) and make successive refinements, joining their images with straight-line segments, until a sufficiently accurate solution has been obtained.
Figure 2. A trajectory in Hénon Map for $\alpha = 1.4$ and $\beta = 0.3$ is shown in (a), the blue points represent the initial segment $\delta \xi$ and the black ones the target region $\epsilon T$. In (b) and (c) zoomed images of $\delta \xi$ and $\epsilon T$ are shown, respectively.

3. Generalized Extremal Optimization

GEO is an evolutionary algorithm in which, unlike other bio-inspired algorithms, there is not a population of strings (or solutions), but a population of bits represented by a single string as shown below. Moreover, there is no crossing between individuals. Instead, each bit in GEO (species) is “forced” to mutate with a probability proportional to its assigned fitness.

Figure 3. Design variables $X_1$ and $X_2$ encoded in a binary string

In this work GEO was used in order to find a point (called optimal point) near to the source, from which the number of iterations needed to reach a region in the target’s neighborhood is minimized.

The practical implementation of GEO algorithm is written as

(i) Initialize randomly a binary string of length $L$ which encodes $N$ design variables of bit length $l_j (j = 1, ..., N)$. For the initial configuration $C$ of bits, calculate the objective function value $V$ and set $C_{\text{best}} = C$ and $V_{\text{best}} = V$. 

3
(ii) For each bit \( i \) of the string, at a given iteration:

(a) Flip the bit and calculate the objective function value \( V_i \) of the string configuration \( C_i \).

(b) Set the bit fitness as \( \Delta V_i = (V_i - V_{best}) \). It indicates the relative gain (or loss) which one has when the bit is mutated, compared to the best objective function value found so far.

(c) Return the bit to its original value.

(iii) Rank the bits according to their fitness values, from \( k = 1 \) for the least adapted bit to \( k = L \) for the best adapted. In a minimization problem, higher values of \( V_i \) will have higher ranking, whereas for maximization problems is the opposite. If two or more bits have the same fitness, rank them randomly with uniform distribution.

(iv) Choose with equal probability a candidate bit \( i \) to mutate. Generate a random number \( RAN \) with uniform distribution in the range \([0,1] \). If the probability of mutation \( P_i(k) = k - \tau \) (where \( \tau \) is an adjustable positive parameter) of the chosen bit is equal or greater than \( RAN \) the bit is confirmed to mutate. Otherwise, the process is repeated until a bit is confirmed to mutate.

(v) For the mutated bit set \( C = C_i \) and \( V = V_i \).

(vi) If \( V < V_{best} \) (\( V > V_{best} \), for a maximization problem) then set \( V_{best} = V \) and \( C_{best} = C \).

(vii) Repeat steps 2 to 6 until a given stopping criterion is fulfilled.

(viii) Return \( C_{best} \) and \( V_{best} \).

4. Successive Perturbations

Consider a map, given by \( Z_{n+1} = F(b, Z_n) \), where \( b \) is a parameter which can be changed by \( \pm \delta \), so that it can assume three possible values: \( b + \delta \), \( b \), \( b - \delta \). The method uses the three possible parameter values to direct a trajectory from a starting point \( P_i \) to a neighborhood of target \( P_f \).

Initially, the map \( F \) is applied to the point \( P_i \), using three possible values of \( b \). The first iteration generates, then, three new points: \( Z_{1,1} = F(b + \delta, Z_0) \), \( Z_{1,2} = F(b, Z_0) \) and \( Z_{1,3} = F(b - \delta, Z_0) \).

This procedure is kept with \( Z_{1,1}, Z_{1,2} \) e \( Z_{1,3} \). Thus, after the second iteration, using the possible values of \( b \), nine points are generated. The process is repeated until one of points \( Z_{i,j} \) reaches the neighborhood of the final desired neighborhood.

In the following results the successive perturbations are represented, respectively, by a sequence specifying which disturbance was applied at each iteration, such that: \( -\delta \) indicates \( \alpha = \bar{\alpha} + \delta \), \( 0 \) indicates \( \alpha = \bar{\alpha} \), \( +\delta \) indicates \( \alpha = \bar{\alpha} - \delta \).

The possibles paths from a point \( Z_0 \) are shown in figure 4.

![Possible paths to reach the target](image_url)
5. Results
In all trajectory shown below, the original orbit reaches the target’s neighborhood after $10^6$ iterations.

![Graph 1](image1)

**Figure 5. Geometric approach.** Optimal orbit obtained with 37 iterations. Required perturbation $\delta(x, y) = (-3.109 \times 10^{-2}, 5.195 \times 10^{-4})$

![Graph 2](image2)

**Figure 6. Generalized Extremal Optimization.** Optimal orbit obtained with 36 iterations. Required perturbation $\delta(x, y) = (-1.416 \times 10^{-2}, -6.069 \times 10^{-2})$

![Graph 3](image3)

**Figure 7. Successive Perturbations.** Optimal orbit obtained with 12 iterations. Sequence of perturbations: $-\delta, 0, +\delta, 0, -\delta, -\delta, 0, +\delta, -\delta, -\delta, -\delta, 0$
6. Final Remarks
As GEO ignores the dynamics of the system, unknowing the feasibles regions, although the number of iterations is (on average) less than the found for the geometric approach, in some cases the optimal point may be located away from basin of attraction, thus requiring a larger perturbation.

Geometric approach, on the other hand, found, on average, the optimal trajectory within a higher number of iterations. It was able, however, to find points quite close to the source point, requiring a perturbation in order one or two times lower than the one found in GEO or successive perturbations.

The method of successive perturbations may be adjusted so that tiny values of $\delta$ are used, however, at each level $N$, $3^N$ analyses are performed, hence the required memory may be computationally impracticable.

In the presented results it is possible to note the feasibility of using sensitive dependence to guide a trajectory to a final desired state quickly. The implemented methods may also be applied to higher dimension systems.

References
[1] Baptista, M. S.; Caldas, I. L. - Easy-to-Implemente Method to Target Nonlinear Systems - CHAOS. 1998. v 8, p 290.
[2] De Sousa, F. L.; Ramos, F. M. - New Stochastic Algorithm for Design Optimization - AIAA Journal. 2003. v 41, p 1808.
[3] Hénon, M. - A Two-Dimensional Mapping with a Strange Attractor - Communications in Mathematical-Physics. 1976. v 50, p 69.
[4] Macau, E. E. N.; Caldas, I. L. - Driving trajectories in chaotic scattering - Physical Review E. 2002. v 65, p 2621.
[5] Macau, E. E. N.; De Sousa, F. L.; Ramos, F. M. - Using Chaos to Guide a Spacecraft to the moon - Acta Astronautica. 2005. v 47, p 871.
[6] Macau, E. E. N.; Grebogi C. - Control of chaos and its relevancy to spacecraft steering - Philosophical Transactions. Royal Society. Mathematical, Physical and Engineering Sciences. 2006. v 364, p 2463.
[7] Shinbrot, T.; Ott, E.; Grebogi, C.; Yorke, J. A. - Using Chaos to Direct Trajectories to Targets - Physical Review Letters. 1990. v 65, p 3215.