# Controlling chaos in unidimensional maps using macroevolutionary algorithms

Jesús Marín[1,2,3] and Ricard V. Solé[2,4]
[1]E. U. d'Enginyers Tècnics Industrials de Barcelona, Department of Automatic Control, Technical University of Catalonia, Compte d'Urgell, 187, 08036 Barcelona, Spain
[2]Complex Systems Research Group, Department of Physics, Technical University of Catalonia, Campus Nord B4, 08034 Barcelona, Spain
[3]Department of Software, Technical University of Catalonia, Campus Nord C5, 08034 Barcelona, Spain
[4]Santa Fe Institute, Hyde Park Road 1399, New Mexico 87501

(Received 31 January 2001; revised manuscript received 4 October 2001; published 18 January 2002)

We introduce a simple search algorithm that explores the parameter of periodically perturbed discrete maps in order to find desired orbits through chaos control. The method has been applied to one-dimensional maps but is easily extendable to higher-dimensional systems. Here, we consider two types of chaos control involving proportional pulses in the system variables [Phys. Rev. Lett. **72**, 1455 (1994)] and constant feedback [Phys. Rev. E **51**, 6239 (1995)], the first case being presented in detail. It is shown that our method allows a rapid exploration of parameter space and the finding of high-fitness (i.e., controlled) solutions close to the target orbits, even when high periodicities are required.

DOI: 10.1103/PhysRevE.65.026207 PACS number(s): 05.45.−a, 07.05.Mh

## I. INTRODUCTION

Chaotic behavior in nonlinear dynamical systems is known to be suppressed under some given conditions by using periodic perturbations [1,2]. These techniques of chaos control, particularly the method discovered by Ott, Grebogi, and Yorke (OGY) [1] have been shown to work particularly well on systems involving low-dimensional dynamical systems of many types. Further modifications of the OGY method have been able to stabilize high-periodic orbits [3]. Some of these methods have been successfully applied to experimental systems. The implications of these results are deep, since chaos control can be used to manipulate dynamical patterns in real systems and can also be used as a source of stabilization in population dynamics or neural networks [2,4]. Extensions to spatial systems have also been developed, with potential implications in therapy [5].

There are several known types of chaos control and two well-known ways of performing such control: (a) by manipulating the system so as to modify its behavior from chaos to a given desired periodic orbit and (b) by stabilizing unstable periodic orbits embedded into the strange attractor. Two simple techniques of chaos control, the so-called proportional [6] and constant feedback [7] methods, have been successfully applied to low-dimensional systems of different types and, more recently, to the control of spatial chaos [8].

These techniques can be used to control both discrete and continuous systems and have the advantage of not needing a detailed knowledge of the system's orbits. Two simple methods have been recently proposed: the Güémez-Matías method [7] and the Parthasarathy-Sinha method [6].

We are interested in controlling $p$-periodic orbits in one-dimensional discrete maps of the form

$$x_{n+1}=f_{\mu}(x_n), \tag{1.1}$$

where $f_{\mu}(x)$ is a single-parameter quadratic map and $x$ belong to space state $\Omega$. In our paper, we have used the logistic map defined by

$$x_{n+1}=4\mu x_n(1-x_n), \tag{1.2}$$

where $\mu\in[0,1]$ and $\Omega=[0,1]$. For $\mu>\mu_c=0.892\,47,\ldots$, chaotic dynamics starts to appear. The underlying idea of chaos control is that the chaotic solutions at this domain of parameter space contain an infinite number of periodic, unstable solutions

$$O_{\mu}^{(p)}=\{x_1^*,\ldots,x_p^*;x_i^*=f_{\mu}^p(x_i^*),\forall i\}, \tag{1.3}$$

which can be stabilized under appropriate periodic perturbations. The presence of these periodic orbits has important consequences. Chaos control has been successfully applied to several biological systems such as heart [9] and brain [10] dynamics and has been suggested to be feasible in controlling experimental ecological systems [11].

An obvious problem emerging from the study of chaos control is the choice of the appropriate parameters together with the potential high periodicity of the desired orbits to be stabilized. When a good knowledge of the system is available, some methods (such as OGY) allow to define a clear criterion of control by applying small feedback to one of the accessible system parameters. These methods have been analyzed in depth. However, in many cases, such *a priori* knowledge of the dynamics of the system (such as the location of stable fixed points) is not available. Besides, successful control is often limited to low-periodic orbits.

Other methods have been used involving rather nonspecific changes in the system variables. One of them [6] applies—in an additive way—a constant feedback ($\gamma$), i.e.,

$$x_{n+1}=f_{\mu}(x_n)+\gamma, \tag{1.4}$$

where $\gamma\in[-1,1]$ gives the strength of the pulse (and will be used as relevant parameter in the search algorithm, see below).

Another control algorithm [7] has been shown to stabilize chaotic systems by applying proportional pulses to the system variables. In this method, every $p$ iterations, a pulse of strength $\gamma\in[-1,1]$ is applied as follows:

 

$$x_{n+1}=f_\mu(x_n)(1+\gamma) \quad \text{for} \quad (n \;\mathbf{mod}\; p)=0,$$
$$x_{n+1}=f_\mu(x_n) \quad \text{otherwise}. \tag{1.5}$$

Most previous studies involving these nonparametric perturbation methods considered specific values of the control parameters but no general method for finding them was studied. The problem considered here is: given a target orbit of period $p$

$$T^{(p)}=\{x_1',x_2',\ldots,x_p';x_i'\in\Omega\}, \tag{1.6}$$

is it possible to find the set of conditions allowing the stabilization of $O_\mu$ toward $T^{(p)}$ for a given $\mu$?

Since there is no analytic treatment to this problem, an alternative method is a search in parameter space through some suitable algorithm. In a previous study by Weeks and Burguess [12], it was shown that a neural network-training method based on genetic algorithm evolution was able to control chaotic maps in one and two dimensions. More recently [13], genetic algorithms have been used in local control to stabilize equilibrium points. Other techniques [14,15] are based on a reinforcement learning procedure in order to find the best control strategy to be applied from a set with prefixed control values.

Such methods required no previous knowledge about the system to be controlled, and all of them have been shown to work when fixed points or low-periodic orbits are the target to be reached. However, high-periodic orbits are seldom considered. Following the same lines (i.e., by exploring the parameter space through evolutionary search), we consider a simple evolutionary algorithm globally searching those control parameter from a continuous interval of values such that the controlled orbit is closest to the desired target $T^{(p)}$, which can have high periodicity.

The paper is organized as follows: in Sec. II, the method is presented. In Sec. III, the results of the stabilization for several orbits are analyzed. In Sec. IV, our main results and possible extensions are discussed.

## II. SEARCH METHOD

Our search method is based on a modification of a previous study of evolutionary optimization on fitness landscapes [16,17]. A first step in our application requires an appropriate definition of a fitness function or objective function $F$ to be maximized, comparing our given target orbit $T^{(p)}$ with the candidate orbit

$$O^{(q_i)}_{\mu,\gamma}=\{x_1^*,\ldots,x_{q_i}^*\}. \tag{2.1}$$

Here $q_i$ indicates the periodicity of this orbit (which may or not be equal to $p$). $O^{(q_i)}_{\mu,\gamma}$ will be generated fixing $\mu$ and applying a perturbation $\gamma$ with periodicity $p$ using control defined in Eqs. (1.5) or (1.4) over system.

The proposed fitness function uses a periodicity similarity measure between $q_i$ and $p(\sigma_{per})$, and a similarity measure between points in orbits $(\sigma_{orb})$, so as follows:

$$F(\mathbf{u})=\sigma_{per}\sigma_{orb}, \tag{2.2}$$

where $\mathbf{u}$ is a vector within parameters to optimize. In our case, $\mathbf{u}=\{\gamma\}$, although this could also include $\mu$.

These are the criteria followed in order to define similarity measures:

(1) The difference between the periodicity of the $i$th candidate orbit $(q_i)$ and the target orbit $(p)$ must to be minimal. For this reason we define periodicity similarity measure $(0 \leqslant \sigma_{per} \leqslant 1)$ as

$$\sigma_{per}=1-\epsilon_{per}|q_i-p|, \tag{2.3}$$

where $\epsilon_{per}$ is the precision in the points that define the target orbit (here we use $\epsilon_{orb}=10^{-3}$). That means that state space $\Omega$ has been discretized into $(1/\epsilon_{per})$-sized bins in order to estimate $q_i$.

(2) An error measure $\delta$ between the points belonging to the target orbit $(T^{(p)})$ and the points belonging to the candidate orbit $(O^{(q)}_{\mu,\gamma})$ is defined as

$$\delta=\frac{1}{k}\sum_{j=1}^{k}\left|T^{(p)}[1+(j \;\mathbf{mod}\; p)] - O^{(q_i)}_{\mu,\gamma}[1+(j \;\mathbf{mod}\; q_i)]\right|^2, \tag{2.4}$$

where $k$ is the minimum common multiple of $\{p,q_i\}$ in order to guarantee all points of both orbits will be considered. We define the similarity measure between both orbits $\sigma_{orb}\in[0,1]$, normalizing in power scale

$$\sigma_{orb}=0 \quad \text{if} \quad \exists j\in[1,q_i]:O^{(q_i)}_{\mu,\gamma}(j)\notin\Omega,$$
$$\sigma_{orb}=2^{\log_2(\epsilon_{orb})\sqrt{\delta}} \quad \text{otherwise} \tag{2.5}$$

where $\epsilon_{orb}$ is the accuracy used (in our simulations, $\epsilon_{orb}=1/256$). The choice of the logarithm scaling has been made in order to enhance the differences in the underlying fitness landscape and thus shorten the search time.

We have defined the problem in terms of an optimization of the fitness function $F(\mathbf{u})$. We can apply some optimization method in order to find the strength of pulse $(\gamma\in[-1,1])$ that maximizes $F(\{\gamma\})$. Some methods, such as genetic algorithms [18–20], random search [21], and macroevolutionary algorithms (MA) [16,17] have been tested, being the last one the more efficient and simple in implementation.

The macroevolutionary algorithm was developed as a simple alternative to genetic algorithms. As these, MA optimizes a fitness function through a parallel search using a constant number $N$ of possible candidate solutions (population). The population $M$ evolves at each time step (generation) by applying operators acting over each solution candidate. However, MA has monotonous convergence doing easier tunning algorithm parameters, and allows us to find better solutions using under less computationally costly conditions.

These specific operators for MA can be summarized as follows:

(1) *Selection operator*: it allows us to decide which are candidate solutions surviving in the next generation. The state $S_i$ of a given candidate $\mathbf{u}_i\in M$ will be given by

$$S_i(t+1) = \text{alive} \quad \text{for} \quad F(\mathbf{u}_i) \geq \sum_{j=1}^{N} \frac{F(\mathbf{u}_j)}{N},$$

$$S_i(t+1) = \text{extinct} \quad \text{otherwise}, \tag{2.6}$$

where $t$ is generation number.

(2) *Colonization operator*: it allows us to fill extinct candidates in two ways. With a probability $\tau$, a totally random solution $\mathbf{u}_r$ will be generated. In our case, the random solution is simply a random number $\gamma \in [-1,1]$ with uniform distribution. Otherwise, exploitation of surviving solutions takes place through colonization: the extinct solution will be attracted towards the best-fitness solution of population $(\mathbf{u}_b)$. Mathematically, this reads as

$$\mathbf{u}_i(t+1) = \mathbf{u}_b(t) + \rho\lambda[\mathbf{u}_b(t) - \mathbf{u}_i(t)] \quad \text{for} \quad \xi > \tau,$$

$$\mathbf{u}_i(t+1) = \mathbf{u}_r \quad \text{for} \quad \xi \leq \tau, \tag{2.7}$$

where $\xi \in [0,1]$ is a random number, $\lambda \in [-1,+1]$ (both with uniform distribution) and $\rho$ and $\tau$ are given constants of our algorithm. So, we can see that $\rho$ describes a maximum radius around surviving solutions and $\tau$ acts as a temperature parameter.

In our case, a candidate solution $\mathbf{u}$ consists of a strength of pulse $(\gamma)$ with a predefined parameter range $(\gamma \in [-1, +1])$. All simulations has been performed using a population of $N=100$ candidates along $G=100$ generations, with algorithm parameters $\rho=0.5$ and $\tau$ varying (such as in simulated annealing [22]) using a sigmoidal function defined by

$$\tau(t) = \frac{1}{1+e^{\psi(t-G/2)}} \tag{2.8}$$

with a slope $\psi=0.2$.

The complete algorithm is presented in Fig. 1.

**Parameters:**
```
  Parameter to optimize: u_i = {γ}
  Algorithm parameters: {N, G, ρ}
  Map parameters: {μ, transient size, T^(p)}
```

**Algorithm description:**
```
  Choose a random M // initialization
  For each u_i ∈ M, apply fitness function F(u_i)
  u_b ← Get best solution of M so as ∀u_i ∈M : F(u_i) ≤F(u_b)
  For generation t from 1 to G do
     τ ← (1 + e^{ψ(t−G/2)})^{−1}
     For each u_i ∈ M do
            // selection
          If F(u_i) < Σ_{j=1}^N F(u_j)/N then
               // Colonization
             ξ ← random(0,1)
             If ξ ≤ τ then
                 u_i ← random  // random solution, that is, random γ
             Else
                   // mixing features between best and extinct
                 λ ← random(-1,1)
                 u_i ← u_b + ρλ(u_b − u_i)
             End
             Apply fitness function F(u_i)
          End
     End
     u_b ← Get best solution of M so as ∀u_i ∈M : F(u_i) ≤F(u_b)
  End
  Shows u_b as the best found solution
```

FIG. 1. Summary of optimization process to control unidimensional maps using macroevolutionary algorithms.

### III. RESULTS

The target orbits have been chosen from the bifurcation scenarios generated by using $\mu$ parameters in the chaotic domain of the logistic map. These scenarios have been obtained by using different values of $p$ and $\gamma$. Once a given $p$ has been chosen, we have numerically simulated the dynamics of the perturbed map by applying different strengths. This leads to a rich variety of orbits of many different periodicities. From these orbits we chose a few as target solutions (listed in Table I). Here, low $p=2$ but also high $p=13$ periodicities have been used.

Let us first consider the observed structure of the fitness landscapes. In Fig. 2(a), the Lyapunov exponent corresponding to a logistic map with periodic perturbations each four iterations is shown. In Figs. 2(b) and 2(c), we have plotted the fitness landscapes $F(\{\gamma,\mu\})$ of different orbits by varying $\mu$ and $\gamma$ (here, darker points mean higher fitness values). Orbits #2 and #6 have been used (see Table I.) Not surprisingly, the function $F(\{\gamma,\mu\})$ in Fig. 2(b) shows a close resemblance with the Lyapunov plot shown in Fig. 2(a). This is due to the underlying relationship between orbit complexity and fitness. The Lyapunov map structure shown in the first plot has been in fact deeply analyzed by M. Markus and co-workers [23]. For this particular case of low periodicity, the landscape appears rather smooth (with a few, localized exceptions) and one can imagine that a hill climbing by a standard evolutionary search algorithm will work well. But we can see that the landscape associated to the second orbit is much more rugged, with sharp, thin highly fit domains (as the small horned structure on the upper-left part). The situation gets worse as periodicity increases.

As an example, in Fig. 3(a) we show the bifurcation diagram that includes orbit #4 (with $\mu=0.99$) as a thin periodic window. Here, the control parameter is the strength $\gamma \in [-1,+1]$ (see Tables I and II). For positive values of the pulse strength no bounded orbit is found. The periodic windows displayed by this diagram are all very localized and our target is shown in the inset plot (for $-0.08 < \gamma < -0.07$).

TABLE I. Table with the target orbits that have been used in the simulations.

| # orbit | $T^{(p)}$, $O^{(p)}_{\mu,\gamma}$ |
|---|---|
| 1 | 0.710 0.236 |
| 2 | 0.841 0.481 0.899 0.372 |
| 3 | 0.344 0.848 0.485 0.898 |
| 4 | 0.316 0.856 0.490 0.913 |
| 5 | 0.788 0.601 0.863 0.425 0.324 |
| 6 | 0.231 0.676 0.834 0.527 0.131 0.434 0.935 |
| 7 | 0.900 0.338 0.806 0.564 0.885 0.365 0.835 0.497 |
| 8 | 0.687 0.769 0.635 0.830 0.506 0.895 0.337 0.800 0.573 0.876 0.390 0.079 0.259 |

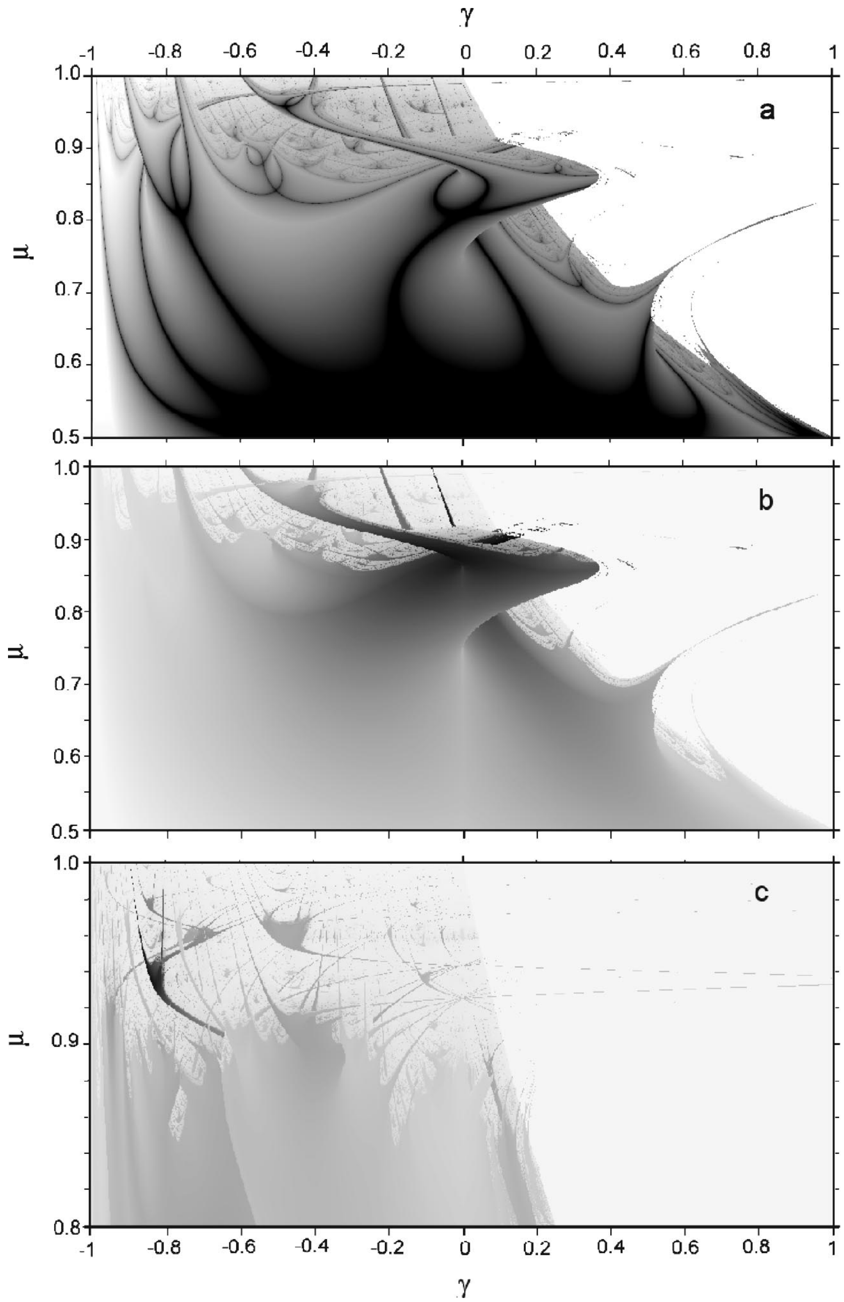

FIG. 2. Varying $\mu$ and strength of pulse ($\gamma$). In (a), Lyapunov exponent for orbit #2 is represented ($p=4$). In (b), fitness function $F(\{\gamma,\mu\})$ (high values in black) for orbit #2. In (c), fitness function $F(\{\gamma,\mu\})$ for orbit #6 ($p=7$).

The corresponding fitness, multipeaked landscape is shown in Fig. 3(b). The same plots for the orbit #5 are shown in Fig. 4. In this case, the bifurcation diagram is much more simple [Fig. 4(a)] and the desired orbit is also indicated in the enlarged domain shown by the inset. This time the fitness landscape [Fig. 4(c)] is strongly correlated and almost single peaked (some low peaks are observable but once the domain $\gamma \in (-0.7,-0.1)$ is reached by some candidate solution, a quick climbing should be expected.

Actually, the fitness landscapes explored in our paper show a number of interesting traits. For small $p$ and/or small $\mu$ values (over $\mu_c$) reveal highly correlated landscapes such as the one shown in Fig. 4(b). On the other hand, for higher $\mu$ values and/or higher periodicities, the fitness function $F(\{\gamma\})$ shows typically a very rugged, multipeaked pattern. However, even in the most rugged landscapes, correlations between peak sizes have been observed through our paper. This corresponds to bifurcation diagrams where the target orbit is confined to a thin, sharp periodic window—see Figs. 3(a) and 3(b). This trends is exploited by the search algorithm and in all cases considered, the target solutions are found.

In Figs. 3(c) and 4(c), the fitness increase of the best candidate solution is shown. The algorithm easily finds high-fit solutions and these are rapidly improved. After 100 generations, we always found a controlled orbit with the desired periodicity. Our results are summarized in Table II where we can see that the algorithm always reaches a fitness $F(\{\gamma\}) >0.99$ in most cases. Only combinations of high $\mu$ or high periodicity (orbits #2, #3, and #4) give lower values of fitness. In our simulations, target orbit points belong to existing orbits, rounded to three digits of precision, showing the good

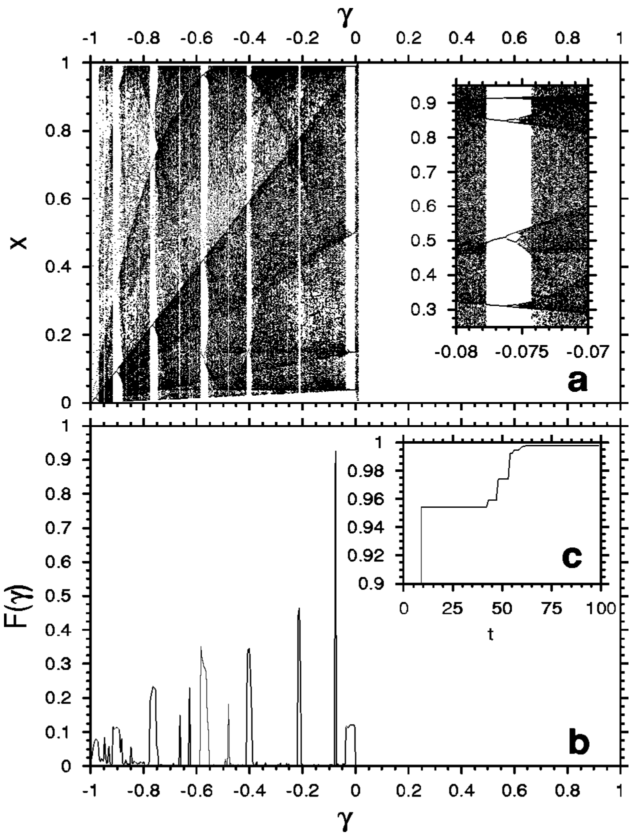

FIG. 3. Fixing $p=4$, and $\mu=0.99$ (orbit #4, see tables): (a) Orbit reached for each strength of pulse ($\gamma$). (b) Fitness value for each strength of pulse ($\gamma$). (c) Fitness reached by MA in a run along 100 generations.

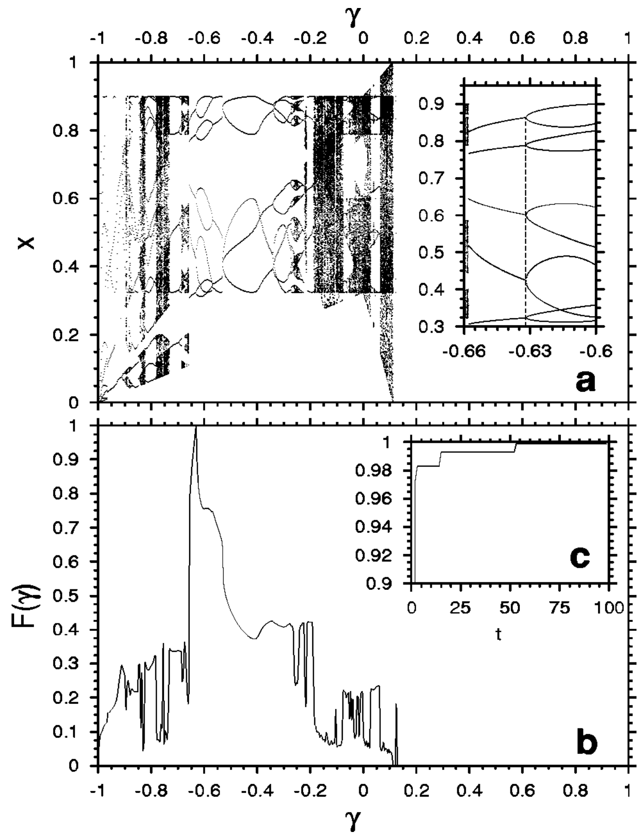

FIG. 4. Fixing $p=5$ and $\mu=0.9$ (orbit #5, see tables): (a) Orbit reached for each strength of pulse ($\gamma$). (b) Fitness value for each strength of pulse ($\gamma$). (c) Fitness reached by MA in a run along 100 generations.

performance and optimality of our procedure. The same results are obtained for the constant feedback system and extension to higher-dimensional maps is straightforward (although we have only explored two-dimensional systems, with very good results).

## IV. DISCUSSION

The problem of how to control unstable orbits in chaotic dynamical systems is one of the most interesting applied topics of nonlinear dynamics. The presence of chaos in physical, biological, and chemical systems has been demonstrated in a large number of cases and is acknowledged to be a common phenomenon.

Different techniques of chaos control have been presented over the last decade, but a systematic method to find the appropriate parameters to stabilize a desired orbit is usually a nontrivial problem. Such difficulties are increased with the system's dimensionality, desired periodicity, and degree of instability (as measured, for example, in terms of the largest Lyapunov exponent).

In this paper, we have introduced a technique able to stabilize chaotic dynamical systems into desired periodic orbits. These orbits were previously chosen from previously generated bifurcation scenarios, but in fact, the method also works

TABLE II. Macroevolutionary algorithm's ability to find existing orbits. For each orbit, 15 runs has been performed.

| $p$ | $\mu$ | # orbit | Fitness reached | $\gamma$ Found | # evaluations | % runs $q_i=p$ |
|---|---|---|---|---|---|---|
| 2 | 0.984 | #1 | $0.999348\pm0$ | $-0.708762$ | $2780\pm94$ | 100% |
| 4 | 0.9 | #2 | $0.998951\pm0.00012$ | 0.135524 | $1956\pm112$ | 100% |
| 4 | 0.94 | #3 | $0.998356\pm0$ | $-0.043487$ | $2669\pm127$ | 100% |
| 4 | 0.99 | #4 | $0.997800\pm0$ | $-0.077732$ | $3529\pm91$ | 100% |
| 5 | 0.9 | #5 | $0.999042\pm0$ | $-0.632015$ | $2661\pm72$ | 100% |
| 7 | 0.952 | #6 | $0.953930\pm0.044318$ | $-0.861962$ | $2985\pm195$ | 100% |
| 8 | 0.9 | #7 | $0.998449\pm0$ | 0.043226 | $2646\pm75$ | 100% |
| 13 | 0.895 | #8 | $0.987704\pm0.020688$ | $-0.907755$ | $3268\pm93$ | 100% |

for arbitrarily chosen orbits, although the found solutions are limited by the constraints imposed by the underlying equations. Since the desired orbits can actually be impossible to find, the algorithm is able to obtain a close solution and thus a fitness lower than one.

Our technique involves an evolutionary search method based on a simple algorithm. The current candidates are selected by their position with respect to the average fitness. This allows us to generate extinctions of many possible sizes, thus enhancing the opportunities for further exploration when good solutions are found. Since the fitness landscapes are correlated (at different levels), the system is able to exploit these correlations and find high-fit, optimal solutions.

The method works well in the two cases considered and extensions to higher dimensions (such as the as Hénon system or a coupled map lattice [8,24]) seem to behave in a rather efficient way, although systematic analysis of such higher-dimensional systems will be presented elsewhere. It is interesting to note, however, that our system succeeds when long-periodic orbits are the targets of the search algorithm. Most systematic approaches (with few exceptions [3]) fail to find the right parameter combinations to reach control when periodic orbits with $p>4$ are used. Although longer transients are observed before the optimum is found, the correlation in the fitness landscape can be exploited to reach a good solution. Future work should be done to study the exploratory capacities of our paper for a larger variety of dynamical systems. In this context, further developments should include approaches able to perform the search without a predefined knowledge of the parameter ranges to be explored.

## ACKNOWLEDGMENTS

We thank Javier Gamarra for useful discussions. This work has been supported by CICYT Grant No. PB97-0693 and by the Santa Fe Institute (R.V.S.).

[1] E. Ott, C. Grebogi, and J. A. Yorke, Phys. Rev. Lett. **64**, 1196 (1990).
[2] T. Shinbrot, C. Grebogi, E. Ott, and J. A. Yorke, Nature (London) **363**, 411 (1993).
[3] E. R. Hunt, Phys. Rev. Lett. **67**, 1953 (1991).
[4] R. V. Solé and L. Menéndez de la Prida, Phys. Lett. A **199**, 65 (1995).
[5] S. Sinha, A. Pande, and R. Pandit, Phys. Rev. Lett. **86**, 3678 (2001).
[6] S. Parthasarathy and S. Sinha, Phys. Rev. E **51**, 6239 (1995).
[7] J. Güémez and M. A. Matías, Phys. Lett. A **181**, 29 (1993).
[8] N. Parekh, S. Parthasarathy, and S. Sinha, Phys. Rev. Lett. **81**, 1401 (1998).
[9] A. M. Garfinkel, W. L. Spano, J. N. Ditto, and J. Weiss, Science **257**, 1230 (1992).
[10] S. J. Schiff *et al.*, Nature (London) **370**, 615 (1994).
[11] R. V. Solé, J. B. P. Gamarra, M. Ginovart, and D. López, Bull. Math. Biol. **61**, 1187 (1999).
[12] E. R. Weeks and J. M. Burgess, Phys. Rev. E **56**, 1531 (1997).
[13] H. Richter and K. J. Reinschke, Physica D **144**, 309 (2000).
[14] S. Gadaleta and G. Dangelmayr, Chaos **9**, 775 (1999).
[15] S. Gadaleta and G. Dangelmayr, Phys. Rev. E **63**, 036217 (2001).
[16] J. Marín and R. V. Solé, IEEE Trans. Evol. Comp. **3**, 272 (1999).
[17] J. Marín and R. V. Solé, in *GECCO-99*, Vol. 2, edited by Banzhaf (San Francisco, Morgan Kaufmann, 1999), pp. 1344–1349.
[18] J. H. Holland, *Adaptation in Natural and Artificial Systems* (MIT, Cambridge, MA, 1992).
[19] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning* (Addison-Wesley, Reading, MA, 1989).
[20] M. Mitchell, *An Introduction to Genetic Algorithms* (MIT, Cambridge, MA, 1996).
[21] W. H. Press, Brian P. Flannery *et al.*, *Numerical Recipes in C* (Cambridge University, Cambridge, England, 1991).
[22] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, Science **220**, 671 (1983).
[23] M. Markus, Comput. Phys. **12**, 481 (1990).
[24] V. V. Astakhov, V. S. Anishchenko, and A. V. Shabunin, IEEE Trans. Circuits Syst. **42**, 352 (1995).