# Using Chaos to Direct Trajectories to Targets

Troy Shinbrot,[a] Edward Ott,[b],[c] Celso Grebogi,[b],[d] and James A. Yorke[d]
*University of Maryland, College Park, Maryland 20742*

(Received 1 June 1990)

A method is developed which uses the exponential sensitivity of a chaotic system to tiny perturbations to direct the system to a desired accessible state in a short time. This is done by applying a small, judiciously chosen, perturbation to an available system parameter. An expression for the time required to reach an accessible state by applying such a perturbation is derived and confirmed by numerical experiment. The method introduced is shown to be effective even in the presence of small-amplitude noise or small modeling errors.

PACS numbers: 05.45.+b

Chaotic systems exhibit extreme sensitivity to initial conditions. This characteristic is often regarded as an annoyance, yet it provides us with an extremely useful capability without a counterpart in nonchaotic systems. In particular, the future state of a chaotic system can be substantially altered by a tiny perturbation. If we can accurately sense the state of the system and intelligently perturb it, this presents us with the possibility of rapidly directing the system to a desired state. As we will show below for a particular example, an initial condition for a chaotic system which, if left on its own, would require more than 6000 time steps to reach a small target region can be directed with small perturbations to the desired region in only 12 time steps. In this paper, we report a method for achieving such targeting, and we show that the method can be effective even in the presence of small-amplitude noise and small modeling errors.

For simplicity, we consider a three-dimensional continuous-time dynamical system, $d\mathbf{X}/dt=\mathbf{F}(\mathbf{X})$. The extension to higher dimensions is given at the end of the paper. We assume that model equations describing the system are known, although they need not be exact. We now employ a surface of section, and denote the coordinates in the surface of section by $\xi$, and the Poincaré map of $\xi$ by

$$\xi_{n+1}=\mathbf{f}(\xi_n,\alpha)\,, \tag{1}$$

where the map $\mathbf{f}$ is necessarily invertible and $\alpha$ is a system parameter.[1] Suppose that we wish to go from a source point $\mathbf{X}_s$ to a small region about a target point, $\mathbf{X}_t$. Following the trajectory from $\mathbf{X}_s$ forward in time, we find its first intersection with the surface of section and denote this point $\xi_s$. Following the trajectory through $\mathbf{X}_t$ backward in time, we similarly determine its first intersection with the surface of section and denote this point $\xi_t$. Thus we have reduced our problem to that of a two-dimensional map in which we desire to go from $\xi_s$ to the vicinity of $\xi_t$. We assume that the system parameter $\alpha$ is available for adjustment at each iterate. Thus we can replace $\alpha$ in the map (1) by $\alpha_n$. However, we also suppose that only small adjustments of $\alpha$ are allowed. That is, $\alpha_n=\bar{\alpha}+\delta_n$, where $\bar{\alpha}$ is a nominal value of $\alpha$ and the deviation from $\bar{\alpha}$, denoted $\delta_n$, is restricted to be small.

If the process under study is ergodic, then in the absence of perturbations (i.e., for $\alpha_n=\bar{\alpha}$) the time required to travel from a source point $\xi_s$ to a small neighborhood $\bar{\varepsilon}_t$ of linear size $\varepsilon_t$ about a target point $\xi_t$ in the ergodic set is typically $\tau_0 \sim 1/\mu(\bar{\varepsilon}_t)$, where $\mu$ denotes the natural probability measure of the chaotic set. The measure $\mu(\bar{\varepsilon}_t)$ typically scales with the information dimension,[2] $D$, so the time required obeys

$$\tau_0 \sim (1/\varepsilon_t)^D\,, \tag{2}$$

for small $\varepsilon_t$. Thus in the absence of perturbations, the amount of time required to reach a desired target increases according to a power law as the size of the target region decreases. We will see that this power law can be converted to a much weaker logarithmic increase by applying a carefully chosen small perturbation.

Now suppose that perturbations of $\alpha$ can be applied. After one iteration of the return map, the change of the state, $\delta\xi$, relative to the point $\mathbf{f}(\xi_s,\bar{\alpha})$, due to a small perturbation, $\delta_1$, is given by the Taylor expansion

$$\delta\xi \cong \left.\frac{\partial \mathbf{f}(\xi_s,\alpha)}{\partial\alpha}\right|_{\bar{\alpha}} \delta_1\,. \tag{3}$$

Letting $\delta_1$ vary through a small interval, $|\delta_1|<\delta_*$, Eq. (3) defines a small line segment through the point $\mathbf{f}(\xi_s,\bar{\alpha})$. We denote this line segment $\overline{\delta\xi}$, and we denote its length $\delta\xi$. Since our system is chaotic, the length of the image of this line segment will grow roughly geometrically with each successive iteration of the map $\mathbf{f}(\xi,\bar{\alpha})$. Let $n_1$ denote the number of iterates required for the small line segment to stretch to a length of order 1. This typically happens when $\delta\xi\exp(n_1\lambda_1)\sim 1$ if $\delta\xi$ is small, where $\lambda_1$ is the positive Lyapunov exponent obtained for typical initial conditions on the attractor. Defining $\tau_1=\lambda_1^{-1}\ln(1/\delta\xi)$, the length of the line segment becomes of order 1 after about $\tau_1$ iterates (i.e., $n_1\sim\tau_1$) if $\delta\xi$ is small. Without loss of generality, we take the size of the relevant ergodic region to be of the order of 1 so that $\tau_1$ is approximately the number of iterates required for the line segment to span the ergodic region. Likewise, if we map the region $\bar{\varepsilon}_t$ backward in time, we find that its preimage spans the ergodic region after a number of preiterates which is typically of the order of $\tau_2 = |\lambda_2|^{-1} \ln(1/\varepsilon_t)$ if $\varepsilon_t$ is small, where $\lambda_2$ is the negative Lyapunov exponent for the map **f** for typical initial conditions on the attractor.

Thus we adopt the following procedure. We iterate the segment $\overline{\delta\xi}$ forward using $a_n = \bar{a}$ for $n_1$ iterates until its length becomes of order unity. We then iterate the region $\bar{\varepsilon}_t$ backward for $n_2$ iterates until it first intersects the $n_1$ forward iteration of the line segment $\overline{\delta\xi}$. Typically for small $\delta\xi$ and $\varepsilon_t$ we have $n_1 \cong \tau_1$ and $n_2 \cong \tau_2$. Iterating a point in the middle of this intersection backward $n_1$ times, we find a point on the line segment $\overline{\delta\xi}$ which is mapped to the target region $\bar{\varepsilon}_t$ in $n_1 + n_2$ iterates. Knowing this point we can then determine the required perturbation $\delta_1$ to be applied on the first iterate from $\boldsymbol{\xi}_s$ by using Eq. (3). Note that, for the situation considered so far, we assume no noise and no modeling error, and consequently we can achieve targeting with $\delta_n = 0$ for $n \geq 2$. For small $\delta\xi$ and $\varepsilon_t$ the total time to go from $\boldsymbol{\xi}_s$ to $\bar{\varepsilon}_t$ by this method scales as

$$\tau = \tau_1 + \tau_2 = \lambda_1^{-1} \ln(1/\delta\xi) + |\lambda_2|^{-1} \ln(1/\varepsilon_t) . \tag{4}$$

For example, for $\delta\xi \sim \varepsilon_t$, we have $\tau \sim \ln(1/\varepsilon_t)$ in contrast to the power law (2).

In practice, we cannot actually iterate either the line $\overline{\delta\xi}$ or the region $\bar{\varepsilon}_t$. Rather, we iterate discrete approximations to them and make successive refinements until a sufficiently accurate intersection is obtained. We do this by putting a fixed number $\hat{N}_s \gg 1$ of equally spaced points on $\overline{\delta\xi}$, iterating these points, and joining their images with straight-line segments. Similarly, we iterate $\hat{N}_t \gg 1$ points on the perimeter of $\bar{\varepsilon}_t$ backward in time and join their images with straight-line segments. Once an intersection is detected, we refine its accuracy by repeatedly halving the intersecting forward and backward line segments and determining which of the halves actually contain the intersection. We must achieve accuracy sufficient to strike the $n_2$ backward iterate of the target, which is a long, thin region with width of order $\varepsilon_t \exp(-\lambda_1 n_2)$. Since we have normalized the size of the attractor to be of order 1, the curvature of the line $\overline{\delta\xi}(n_1)$ and the long sides of the $n_2$ backward iterate of $\bar{\varepsilon}_t$ are also of order 1. Thus, taking account of the curvature, to resolve the intersection within a distance of $\varepsilon_t \exp(-\lambda_1 n_2)$, we require that the distance between points on $\overline{\delta\xi}(n_1)$ and on the $n_2$ backward iterate of $\bar{\varepsilon}_t$ be of order $[\varepsilon_t \exp(-\lambda_1 n_2)]^{1/2}$. The square root results because the maximum distance between the curve and its discrete straight-line approximation is quadratic in the length of the straight-line segment. Each time we halve our line segments, we increase the resolution by a factor of 2 at the expense of including three additional points [one on $\overline{\delta\xi}(n_1)$ and one each on the two segments bounding the backward iterate of $\bar{\varepsilon}_t$ near the intersection]. Thus to resolve the intersection, we require a number of points $N'$ additional to the original $\hat{N} = \hat{N}_s + \hat{N}_t$ points, where $N'$ obeys

$$[\varepsilon_t \exp(-\lambda_1 n_2)]^{1/2} > 2^{-N'/3} . \tag{5}$$

Using the relation $\varepsilon \exp(|\lambda_2| n_2) \sim 1$, this can be rewritten as

$$N' \gtrsim \tfrac{3}{2} D \ln(1/\varepsilon_t) , \tag{6}$$

where $D = 1 + \lambda_1/|\lambda_2|$ is the Lyapunov dimension of the attractor. We stress that $\hat{N}$ is fixed (typically we took $\hat{N} \sim 100$) and does not depend on $\varepsilon_t$ or $\delta\xi$. Consequently, as $\varepsilon_t$ is reduced, the required computational effort increases logarithmically in $\varepsilon_t$ as shown in Eq. (6).

In order to show why our method of using forward and backward iterations was employed, we now contrast it with another conceivable procedure. If one iterated the line segment $\overline{\delta\xi}$ forward until it first intersected the region $\bar{\varepsilon}_t$, it would do so on iterate $n_1 + n_2$. One could then choose a point in this intersection, iterate the point backward $n_1 + n_2$ steps to find the corresponding point on the original line segment $\overline{\delta\xi}$, and then determine $\delta_1$ from Eq. (3). While this works in principle, the numerical requirements of this pure forward scheme are needlessly more severe than when we determine an intersection by iterating $\overline{\delta\xi}$ forward $n_1$ steps and $\bar{\varepsilon}_t$ backward $n_2$ steps. In the pure forward method, to detect an intersection between the target and the $n_1 + n_2$ iterate of the source, we require that the approximation of $\overline{\delta\xi}(n_1 + n_2)$ obtained by joining the $N_s$ points with straight-line segments intersect the region $\bar{\varepsilon}_t$. Since the curvature of $\overline{\delta\xi}(n_1 + n_2)$ is typically of order 1, we thus require

$$\delta\xi(n_1 + n_2)/N_s < \varepsilon_t^{1/2} . \tag{7}$$

The source line will have length unity after $n_1$ iterates, and will then expand by roughly $\exp(n_2 \lambda)$ during the next $n_2$ iterates, where $\lambda$ is the topological entropy.[3] So we require

$$N_s \gtrsim \varepsilon_t^{-1/2} \exp(n_2 \lambda) \tag{8}$$

or

$$N_s \gtrsim (1/\varepsilon_t)^{\lambda/|\lambda_2| + 1/2} . \tag{9}$$

Thus the number of points required by the pure forward method increases exponentially with $1/\varepsilon_t$, but only increases logarithmically with $1/\varepsilon_t$ in the forward-backward method.[4]

We now illustrate the method with a specific chaotic system. In particular, we deal with the Hénon map[5] in the form $x_{n+1} = a + 0.3 y_n - x_n^2$ and $y_{n+1} = x_n$, with $\bar{a} = 1.4$. As an initial example, we choose the target region to be a small square centered on $\boldsymbol{\xi}_t$ with edge length $\varepsilon_t = 0.0038$. We find that for a representative pair of source and target points, say, $\boldsymbol{\xi}_s = (0.4772, -1.188)$ and $\boldsymbol{\xi}_t = (0.1371, -1.328)$, without applying a perturbation, 6062 iterations are required before the orbit from $\boldsymbol{\xi}_s$

strikes within the target neighborhood, $\bar{\varepsilon}_t$. However, if we are permitted to vary $a$ by up to 1 part in 1000 about its nominal value, we find that our targeting method directs the trajectory to the target neighborhood in only twelve iterations.

To confirm the predicted logarithmic behavior in (4), the following numerical experiment was performed. Source and target locations were chosen at random with respect to the natural measure on the Hénon attractor. Then we fix a target size $\varepsilon_t$, and for each pair of source and target points, our targeting algorithm, described above, was applied. The total number of iterations required to go from $\xi_s$ to $\bar{\varepsilon}_t$ was determined for each pair, and the results were then averaged over many source-target pairs. This process was repeated for several values of $\varepsilon_t$. The neighborhood $\bar{\varepsilon}_t$ was chosen to be a small square of edge length $\varepsilon_t$ centered on $\xi_t$. The result of this experiment is shown in Fig. 1. The solid line of slope $\lambda_1^{-1}+|\lambda_2|^{-1}$, predicted by Eq. (4), is consistent with the data. Also shown as a dashed line is the power-law dependence expected without control from Eq. (2) with $D \cong 1.26$ (the information dimension for the Hénon attractor). The logarithmic dependence of the time to reach the target on $\varepsilon_t$ with control shows dramatic improvement over the power-law dependence without control.

The preceding discussion demonstrates that targeting can be achieved for chaotic systems using only small controls. It remains to be shown, however, that the method discussed can be effective in the presence of noise or modeling errors. Thus we suppose that the real system obeys $\xi_{n+1}=\mathbf{g}(\xi_n,a)+\Delta_n$. Here we imagine that our model map $\mathbf{f}(\xi_n,a)$ is slightly in error, and that, unknown to us, the correct form is $\mathbf{g}(\xi_n,a)$. We, furthermore, allow small-amplitude random noise to disturb the system at each iteration as indicated by the term $\mathbf{\Delta}_n$.

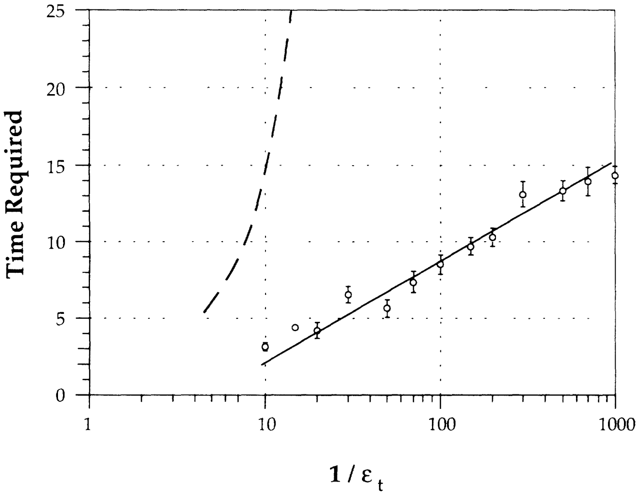

FIG. 1. Average time required to reach typical target neighborhood from typical source with control vs neighborhood size, $\varepsilon_t$ ($\delta\xi=\varepsilon_t$). Solid line has slope predicted from Eq. (4); dashed line indicates expected behavior without control. Error bars are standard error for 25-point means.

To investigate the effect of noise alone, we take $\mathbf{f}=\mathbf{g}$. The following test was performed. Source and target locations were chosen at random on the Hénon attractor, and a trajectory between the source and the target neighborhood was found for the case without noise as previously described. As an example, the neighborhood size $\varepsilon_t$ was chosen to be 0.01 and the time required to hit the target in the absence of noise and with only $\delta_1 \neq 0$ was ten iterations. Then for each of the ten iterations, a random amount of noise was applied with $\mathbf{\Delta}_n$ distributed uniformly in the interval $|\mathbf{\Delta}_n|<\Delta_*$. As shown in Fig. 2 for the case $\Delta_*=0.01$, the noise displaced the tenth iteration to a point (denoted $\xi_{10}$ in the figure) far away from $\xi_t$. Since the noise was applied at every iteration, we next compensated by recomputing the trajectory at every iteration and adjusting the applied perturbation correspondingly. That is, at *each* iterate we used the map $\mathbf{f}$ to determine $\delta_n$ by calculating the intersection of the forward iteration of the line determined from $\delta\xi_{n+1}=\delta_{n+1}\partial\mathbf{f}/\partial a$ with the backward iteration of the region $\bar{\varepsilon}_t$. The result of this procedure is shown in the inset of Fig. 2. The tenth iteration (denoted $\hat{\xi}_{10}$ in the figure) now lies within the target region. Thus we have shown that our method can be effective in the presence of small-amplitude noise provided that we apply a correction $\delta_n$ at each iterate.

It can similarly be shown that targeting can also be achieved even when the system is imperfectly modeled, i.e., when $\mathbf{f}$ differs slightly from the true map, $\mathbf{g}$. After each iteration, we have to compensate for the difference $\mathbf{g}-\mathbf{f}$. For example, let us consider the Hénon map for the case without noise, where $\mathbf{f}$ is the Hénon map with $\bar{a}=1.4$, and $\mathbf{g}-\mathbf{f}=0.014$; $\varepsilon_t$ is still 0.01, and we use the same source and target as in our noise example (cf. Fig. 2). If we apply our procedure only on the first iterate

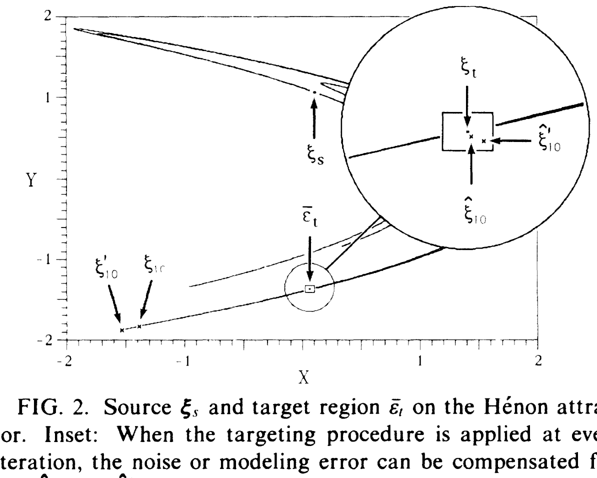

FIG. 2. Source $\xi_s$ and target region $\bar{\varepsilon}_t$ on the Hénon attractor. Inset: When the targeting procedure is applied at every iteration, the noise or modeling error can be compensated for, and $\hat{\xi}_{10}$ and $\hat{\xi}'_{10}$ both lie within the target region.

from $\xi_s$ (as we would if $\mathbf{f}=\mathbf{g}$), then the trajectory again ends at a point (denoted $\xi'_{10}$ in Fig. 2) far from the target. As before, however, if our targeting algorithm is applied at every iteration, the tenth iterate (denoted $\hat{\xi}'_{10}$ in Fig. 2) arrives in the target neighborhood despite the modeling error.

To generalize our method to higher dimensions, consider that the map $\mathbf{f}$ is $N$ dimensional and the attractor has $k$ expanding and $N-k$ contracting directions at typical points. We note that when a $k$-dimensional surface and an $(N-k)$-dimensional surface intersect, generically they do so at isolated points, and small smooth perturbations will not destroy these intersections or create new ones. Thus, for a typical point and a typical small $k$-dimensional disk $D^k$ centered at this point, the $n$th iterate of the disk $\mathbf{f}^n(D^k)$ will be a $k$-dimensional surface and its $k$-dimensional area will increase with $n$. Similarly, if we take an $(N-k)$-dimensional disk, $D^{N-k}$, centered at a typical point, then $\mathbf{f}^{-n}(D^{N-k})$ will be an $(N-k)$-dimensional surface whose area will increase with $n$. As these areas increase, typically they will intersect. We emphasize that targeting can typically be achieved with any dimensionality $N$ even if we only have one available adjustable scalar parameter $\alpha$. To see this we note the following. Consider a trajectory $\xi_i = \mathbf{f}^i(\xi_0, \alpha)$. If we perturb $\alpha$ from $\bar{\alpha}$ by an infinitesimal amount $\delta_i$ at time $i$, then at time $m > i$, a perturbation of $\xi_m$ given by $\mathbf{v}_{i,m}\delta_i$ results, where $\mathbf{v}_{i,m}$ is an $N$-dimensional vector which is determined by the partial derivatives of the map along the trajectory. For typical $\xi_0$ and $\mathbf{f}$, the vectors $\mathbf{v}_{0,k}, \mathbf{v}_{1,k}, \dots, \mathbf{v}_{k-1,k}$ are linearly independent and thus can be used to create the $k$ disk $D^k$.

In conclusion, we have demonstrated that it is possible to rapidly reach a small, accessible target region in a chaotic system by applying small perturbations to an available parameter. The method used is robust against small-amplitude noise and small modeling errors, making it especially suited to practical applications. We emphasize that the problem addressed in this Letter is a very general one and can be expected to arise often.[6]

This research was supported by the U.S. Department of Energy (Scientific Computing Staff Office of Energy Research). The computation was done at the National Energy Research Supercomputer Center.

[a] Department of Physics.

[b] Laboratory for Plasma Research.

[c] Departments of Electrical Engineering and of Physics.

[d] Institute for Physical Science and Technology and Department of Mathematics.

[1] Alternatively, if model equations are not available, the map $\mathbf{f}$ might be derived from experimental data by use of the delay coordinate embedding technique [F. Takens, in *Dynamical Systems and Turbulence,* edited by D. Rand and L. S. Young (Springer-Verlag, Berlin, 1981), p. 230; N. H. Packard *et al.*, Phys. Rev. Lett. **45**, 712 (1980); J.-P. Eckmann and D. Ruelle, Rev. Mod. Phys. **57**, 617 (1985)].

[2] See, for example, J. D. Farmer, E. Ott, and J. A. Yorke, Physica (Amsterdam) **7D**, 153 (1983).

[3] S. Newhouse, in *Physics of Phase Space,* edited by Y. S. Kim and W. W. Zachary (Springer-Verlag, Berlin, 1987), p. 2; Y. Yomdin, Isr. J. Math. **57**, 285 (1987). The length of a *small* line segment typically grows at the exponential rate $\lambda_1$. After the length becomes of order 1, however, it grows at the exponential rate $\lambda$.

[4] We note that some improvement can be obtained by using higher-order fitting (e.g., parabolic rather than linear) of the curves to the iterated points. The exponential and logarithmic dependences of the pure forward and the forward-backward methods remain, however.

[5] Similar results have been obtained for the area-preserving "standard map."

[6] For example, see E. Ott, C. Grebogi, and J. A. Yorke, Phys. Rev. Lett. **64**, 1196 (1990); in *Chaos,* edited by D. K. Campbell (American Institute of Physics, New York, 1990), pp. 153-172.