Targeting and Control of Chaos

Eric J. Kostelich
Department of Mathematics, Box 871804,
Arizona State University, Tempe, Arizona 85287, USA

Ernest Barreto
Department of Physics,
University of Maryland, College Park, Maryland 20742, USA

Abstract

The "control of chaos" refers to a procedure in which a saddle fixed point in a chaotic attractor is stabilized by means of small time dependent perturbations. Control may be switched between different saddle periodic orbits, but it is necessary to wait for the trajectory to enter a small neighborhood of the saddle point before the control algorithm can be applied.

This paper describes an extension of the control idea, called "targeting." By targeting, we mean a process in which a typical initial condition can be steered to a prespecified point on a chaotic attractor using a sequence of small, time dependent changes to a convenient parameter. We show, using a 4-dimensional mapping describing a kicked double rotor, that points on a chaotic attractor with two positive Lyapunov exponents can be steered between typical saddle periodic points extremely rapidly—in as little 12 iterations on the average. Without targeting, typical trajectories require 10,000 or more iterations to reach a small neighborhood of saddle periodic points of interest.

1 Introduction

A chaotic process has sensitive dependence on initial conditions that prevents long-term predictions of the state of the system. Chaotic dynamical processes typically exhibit highly irregular behavior and can be represented mathematically by so-called "strange attractors" whose geometry is very complex. Despite the complexities of chaotic behavior, the sensitive dependence on initial conditions can be exploited to maintain the system about some desired final state (like a saddle periodic orbit embedded in the attractor) by a carefully chosen sequence of small perturbations to a control parameter. This is the basic idea behind the so-called "control of chaos," wherein small perturbations can be used to formulate a feedback stabilization of one of the infinite number of unstable periodic orbits that naturally occur in a chaotic attractor. (See Ref. [1] and the paper by Prof. Ott elsewhere in this volume.) The method relies in part on suitable linear approximations of the stable and unstable manifolds associated with the saddle periodic orbit.

One of the first laboratory experiments to demonstrate the feasibility of this feedback stabilization consisted of a driven, flexible beam whose dynamical behavior was well approximated by a two dimensional map [2]. Although the uncontrolled

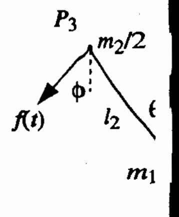


Figure 1: T

process was chaotic, the control algorith
point that was embedded in the attract
to maintain the control. The applicabili
in a variety of laboratory experiments.

A natural extension of this idea is
stated simply as follows: Given a typical
sequence of perturbations that directs t
some prespecified point on the chaotic
the inherent exponential sensitivity of
expects that a suitable alteration of th
small controlling adjustments of one or

An initial demonstration of targeti
sidered some numerical experiments us
targeting procedure was successfully us
dynamics were approximately describab
per, we describe an alternative approac
to systems of higher dimensionality tha

One potential application of the tar
trajectories to a neighborhood of a pr
show, the targeting algorithm is particu
dimension.

We focus attention on the double
a sequence of impulse kicks on two thi
Fig. 1. This idealized mechanical syste
its attractor is a subset of $R^2 \times S^2$.
here, the Lyapunov dimension [7] of th
two positive Lyapunov exponents.

Romeiras *et al.* [6] have demonstr
some of the saddle periodic points embe
showed that control can be achieved by
often is necessary to wait several thousa
a sufficiently small neighborhood of the
linearizations of the dynamical system

Figure 2 shows the results of a ty

ontrol of Chaos

ostelich
natics, Box 871804,
mpe, Arizona 85287, USA

arreto
of Physics,
Park, Maryland 20742, USA

act

edure in which a saddle fixed point in a
f small time dependent perturbations.
ent saddle periodic orbits, but it is
er a small neighborhood of the saddle
applied.
ne control idea, called "targeting." By
ypical initial condition can be steered
actor using a sequence of small, time
eter. We show, using a 4-dimensional
that points on a chaotic attractor with
teered between typical saddle periodic
iterations on the average. Without
00 or more iterations to reach a small
interest.

on initial conditions that prevents long-
. Chaotic dynamical processes typically
represented mathematically by so-called
y complex. Despite the complexities of
on initial conditions can be exploited to
al state (like a saddle periodic orbit em-
sen sequence of small perturbations to a
behind the so-called "control of chaos,"
formulate a feedback stabilization of one
orbits that naturally occur in a chaotic
'rof. Ott elsewhere in this volume.) The
proximations of the stable and unstable
ic orbit.
;s to demonstrate the feasibility of this
flexible beam whose dynamical behavior
nal map [2]. Although the uncontrolled

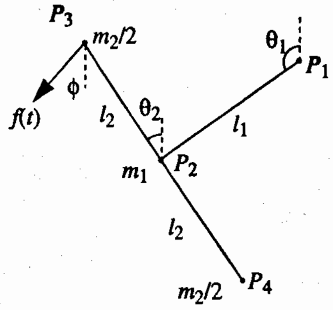


Figure 1: The double rotor.

process was chaotic, the control algorithm maintained the beam about a saddle fixed point that was embedded in the attractor, and only small perturbations were needed to maintain the control. The applicability of the method has since been demonstrated in a variety of laboratory experiments.

A natural extension of this idea is the notion of *targeting*. The problem can be stated simply as follows: Given a typical initial condition on the attractor, determine a sequence of perturbations that directs the resulting trajectory to a small region about some prespecified point on the chaotic attractor as rapidly as possible. Because of the inherent exponential sensitivity of chaotic time evolutions to perturbations, one expects that a suitable alteration of the trajectory can be accomplished using only *small* controlling adjustments of one or more available system parameters.

An initial demonstration of targeting was given by Shinbrot *et al.* [3], who considered some numerical experiments using a two dimensional map. In addition, a targeting procedure was successfully used in a laboratory experiment for which the dynamics were approximately describable by a one dimensional map [4]. In this paper, we describe an alternative approach of the targeting problem that is applicable to systems of higher dimensionality than previously considered.

One potential application of the targeting algorithm is to steer otherwise chaotic trajectories to a neighborhood of a prespecified saddle periodic point. As we will show, the targeting algorithm is particularly effective for systems of moderately high dimension.

We focus attention on the double rotor map [5], which describes the effect of a sequence of impulse kicks on two thin, massless rods connected as illustrated in Fig. 1. This idealized mechanical system exhibits complex dynamical behavior, and its attractor is a subset of $R^2 \times S^2$. For the values of the parameters considered here, the Lyapunov dimension [7] of the attractor is approximately 2.8, and it has two positive Lyapunov exponents.

Romeiras *et al.* [6] have demonstrated a control algorithm that can stabilize some of the saddle periodic points embedded within the double rotor attractor. They showed that control can be achieved by using only one control parameter. However, it often is necessary to wait several thousand iterations before a given trajectory falls in a sufficiently small neighborhood of the desired saddle fixed point so that the required linearizations of the dynamical system are valid.

Figure 2 shows the results of a typical numerical experiment. The plot shows

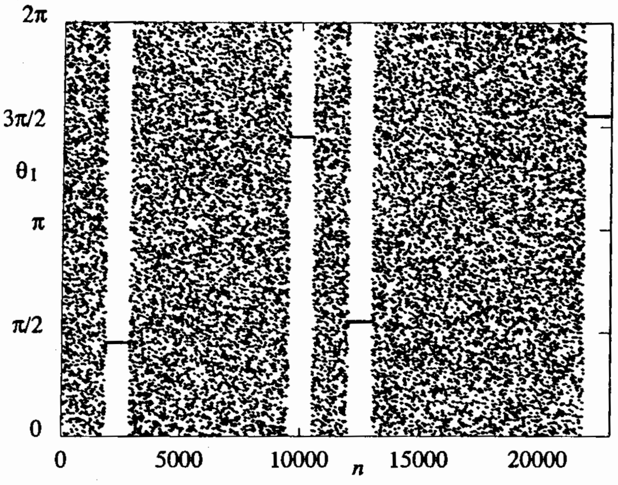


Figure 2: Waiting times for the control of chaos algorithm.

the position θ_1 in radians of one of the rotors as a function of time n. Before the control is applied (for example, during the first 2000 or so iterations of the map), the rotors move chaotically, and the position x_1 assumes values throughout the interval $(0, 2\pi)$. Eventually, one of the iterates lands sufficiently close to a prespecified period-1 saddle point A that the control algorithm can be applied. As long as the control is on, the iterates remain in a small neighborhood of A (for instance, for n approximately between 2000 and 3000). To switch to a different period-1 saddle fixed point B, the control is turned off, allowing the orbit to move away from A and resume its chaotic motion. Eventually, the orbit enters a suitably small neighborhood of B, the control is turned on, the point remains near B until the control is turned off, etc.

This example illustrates an application wherein one wants to switch between different periodic saddle orbits. A difficulty arises in that the waiting times can be quite long. (Although they are not shown in the figure, waiting times of 150,000 or more iterates are not uncommon in this numerical experiment.) In general, one expects that the average waiting time before a typical orbit approaches a given saddle periodic point is proportional to the dimension of the attractor. In this case, the dimension is approximately 2.8, so the average distance between nearest neighbors in a subset of N points on the attractor scales as $N^{-1/2.8}$ [8]. In other words, if the orbit must fall within 10^{-2} of the saddle point for the control algorithm to work, then the waiting time is on the order of $10^{5.6}$ iterations. The observed waiting times in the numerical experiment are consistent with this rough estimate.

The targeting problem has a natural application here, because it can reduce the waiting time by orders of magnitude. In this example, the objective is to choose target points that lie in small neighborhoods of the saddle periodic points of interest. In order to switch between periodic points, one applies the targeting algorithm to steer the trajectory to a small neighborhood of one of the periodic points, then turns on the control algorithm to maintain the orbit near the point for as long as desired.

After the control is turned off, tl
trajectory to a neighborhood of a
The targeting problem also is
sional loss of control, even when t
such cases, one wants to return to

2 The double rotor

In this section, we outline the bas
first with a brief outline of the dou
slightly different version of the ma
The first rod, of length ℓ_1, piv
of length $2\ell_2$, pivots about P_2 (w
position of the two rods at time t
masses $m_2/2$ are attached to each
at P_1 (with coefficient ν_1) slows t
velocity $\dot\theta_1(t)$; friction at P_2 slows t
first rod) at a rate proportional to
P_3 receives impulse kicks at times t
with strength ρ. Gravity and air r
The double rotor map is the fo

$$x_{n+1} = \left(\begin{array}{c} \Theta_{n+1} \\ \dot\Theta_{n+1} \end{array} \right.$$

Here Θ_n and $\dot\Theta_n$ are 2-vectors,

$$\Theta_n = \left(\begin{array}{c} \theta_1^{(n} \\ \theta_2^{(n} \end{array} \right.$$

that give the angular positions and
$\theta_i^{(n)} = \theta_i(nT)$ and $\dot\theta_i^{(n)} = \dot\theta_i(nT^+)$.
Also,

$$G(\Theta$$

and L and M are constant 2×2 m
$m_2\ell_2^2 \equiv I$. Then

$$L = \sum_{i=1}^{2} W_i e^{\lambda_i T},$$

with

$$W_1 = \left(\begin{array}{cc} a & b \\ b & d \end{array} \right.$$

where $a = \frac{1}{2}(1 + \nu_1/\Delta)$, $d = \frac{1}{2}(1$
$-\frac{1}{2}(\nu_1 + 2\nu_2 \pm \Delta)$, and $c_{1,2} = \rho\ell_{1,}$

After the control is turned off, the targeting algorithm can be applied to steer the trajectory to a neighborhood of another prespecified periodic point, and so on.

The targeting problem also is applicable when noise in the system causes occasional loss of control, even when the orbit initially is near the desired fixed point. In such cases, one wants to return to the periodic point as quickly as possible.

2 The double rotor map

In this section, we outline the basic ideas behind the targeting procedure. We begin first with a brief outline of the double rotor map. A derivation can be found in [5]; a slightly different version of the map (that is used here) is described in [6].

The first rod, of length ℓ_1, pivots about P_1 (which is fixed), and the second rod, of length $2\ell_2$, pivots about P_2 (which moves). The angles $\theta_1(t)$, $\theta_2(t)$ measure the position of the two rods at time t. A point mass m_1 is attached at P_2, and point masses $m_2/2$ are attached to each end of the second rod (at P_3 and P_4). Friction at P_1 (with coefficient ν_1) slows the first rod at a rate proportional to its angular velocity $\dot{\theta}_1(t)$; friction at P_2 slows the second rod (and simultaneously accelerates the first rod) at a rate proportional to $\dot{\theta}_2(t) - \dot{\theta}_1(t)$. The end of the second rod marked P_3 receives impulse kicks at times $t = T, 2T, \ldots$, always from the same direction and with strength ρ. Gravity and air resistance are absent.

The double rotor map is the four dimensional map $x_{n+1} = F(x_n)$, defined by

$$x_{n+1} = \begin{pmatrix} \Theta_{n+1} \\ \dot{\Theta}_{n+1} \end{pmatrix} = \begin{pmatrix} (M\dot{\Theta}_n + \Theta_n) \bmod 2\pi \\ L\dot{\Theta}_n + G(\Theta_{n+1}) \end{pmatrix}. \tag{1}$$

Here Θ_n and $\dot{\Theta}_n$ are 2-vectors,

$$\Theta_n = \begin{pmatrix} \theta_1^{(n)} \\ \theta_2^{(n)} \end{pmatrix}, \qquad \dot{\Theta}_n = \begin{pmatrix} \dot{\theta}_1^{(n)} \\ \dot{\theta}_2^{(n)} \end{pmatrix},$$

that give the angular positions and velocities of the rods after the nth kick. That is, $\theta_i^{(n)} = \theta_i(nT)$ and $\dot{\theta}_i^{(n)} = \dot{\theta}_i(nT^+)$. The angles θ_1 and θ_2 are taken to lie in $[0, 2\pi)$. Also,

$$G(\Theta) = \begin{pmatrix} c_1 \sin\theta_1 \\ c_2 \sin\theta_2 \end{pmatrix},$$

and L and M are constant 2×2 matrices. For simplicity, we assume $(m_1 + m_2)\ell_1^2 = m_2\ell_2^2 \equiv I$. Then

$$L = \sum_{i=1}^{2} W_i e^{\lambda_i T}, \qquad M = \sum_{i=1}^{2} W_i (e^{\lambda_i T} - 1)/\lambda_i$$

with

$$W_1 = \begin{pmatrix} a & b \\ b & d \end{pmatrix}, \qquad W_2 = \begin{pmatrix} d & -b \\ -b & a \end{pmatrix}$$

where $a = \frac{1}{2}(1 + \nu_1/\Delta)$, $d = \frac{1}{2}(1 - \nu_1/\Delta)$, $b = -\nu_2/\Delta$, $\Delta = \sqrt{\nu_1^2 + 4\nu_2^2}$, $\lambda_{1,2} = -\frac{1}{2}(\nu_1 + 2\nu_2 \pm \Delta)$, and $c_{1,2} = \rho\ell_{1,2}/I$. In all the numerical work described in this

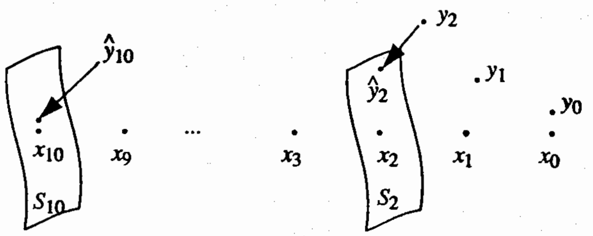


Figure 3: A path of points, consisting of x_{10} and its preimages x_9, x_8,ldots, x_0. The uncontrolled trajectory starting from y_0 rapidly moves away from the target trajectory. The targeting procedure determines perturbations that move the point y_2 to a new point $\hat{y}_2$ on the stable manifold S_2 of x_2. The trajectory starting from $\hat{y}_2$ rapidly approaches the target trajectory, as represented by the point $\hat{y}_{10}$. The surfaces labeled S_2 and S_{10} are representations of the stable manifolds associated with the points x_2 and x_{10}, respectively.

paper, we fix the values of the parameters

$$\nu = T = I = m_1 = m_2 = \ell_2 = 1, \qquad \ell_1 = \frac{1}{\sqrt{2}}$$

and use the force ρ as the control parameter, taking as the nominal value $\rho = \bar{\rho} = 9$.

We write $x_n = F^n(x_0)$ to mean the n times iterated point x_0, *i.e.*, the point obtained by iterating the map n times starting from x_0. The double rotor map is invertible, so $F^{-n}(x_0)$ refers to the nth iterate of x under the inverse map. The notation $F(x)$ means that the map is applied with the kick set to its nominal value (here $\bar{\rho} = 9$); the notation $F(x, \rho)$ means the map applied to x with the kick set to ρ. We let $DF(x, \rho)$ denote the corresponding Jacobian matrix of partial derivatives of F with respect to θ_i and $\dot{\theta}_i$.

3 The targeting procedure

Let T be typical point on the attractor and suppose that T is the target point. A *path* is a set of points consisting of T and a sequence of its preimages. The basic idea is illustrated in Fig. 3. The target point is labeled as x_{10} to emphasize the idea that this path shows the target and ten of its preimages.

Suppose that, as the map is iterated, we find a point y_0 that falls near x_0. We wish to find a sequence of perturbations to some available parameter such that the orbit starting at y_0 approaches the orbit starting at x_0. (As the dynamics are chaotic, the orbits diverge rapidly without targeting.) We now outline the basic targeting algorithm in the case where one parameter (ρ in the case of the double rotor map) is available for control.

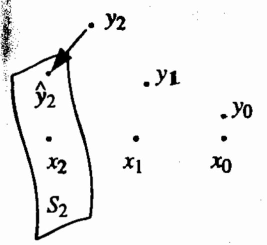
If the problem were linear, then in principle we could compute four successive perturbations $\delta\rho_0$, $\delta\rho_1$, $\delta\rho_2$ and $\delta\rho_3$ that allow us to hit the point x_4 starting from y_0, because in general the gradient vectors $\partial F/\partial\rho_0, \ldots, \partial F/\partial\rho_3$ are linearly independent.

In practice, the problem is highly nonlinear, and it is not possible to find a sufficiently accurate linearization so that Newton's method can be applied to determine

the perturbations needed to hit x_4
adopt an alternative approach. (Fo
and the effects of numerical roundo
For the parameter values given
has two positive and two negative L
Lyapunov exponents of the double ro
of Benettin *et al.* [9]) As a result, as
2-dimensional unstable manifold an
the stable manifold associated with
then the orbit starting from s appr
iterated. In general, the rate of ap
exponents associated with the attra
of the targeting algorithm is to try
to move the orbit starting from y_0
the x's. If it is successful, then the
In the case where one parameter
tions $\delta\rho_0$ and $\delta\rho_1$ so that the orbit st
fold of x_2. (Recall that S_2 is a 2-dime
and $\partial F(F(x_0, \rho_0), \rho_1)/\partial\rho_1$ typically
in R^4, the 2-plane spanned by the
denoted by $\hat{y}_2$ in Fig. 3.)
A basic difficulty arises in appro
linear approximation of S_2 is inadeq
can be found by calculating the inv
path. For instance, a reasonable line
in a small neighborhood of x_{10}, can
associated with the matrix $DF(x_2, \rho$
vectors that span this stable eigensp
$\sigma_1 s_1$, where $|\sigma_i|$ is small. Although
$F^{-1}(z)$, $F^{-2}(z)$, ... rapidly approa
S_{10} is an expanding set under the in
contract.
Thus, in the case of one paramet
problem consists of determining two
with values for σ_0 and σ_1, such that

$F^{-8}(x_{10} + \sigma_0 s_0 +$

Equation (2) can be solved numerica
guarantee that Newton's method wi
termined two successive perturbatio
of y_0 onto the stable manifold of x
merical roundoff, the dynamics brin
exponential rate determined by the
There is nothing special about
iterations to estimate S_2. For exam
timate S_2 by the fourth inverse iter
is farther away, say at x_{12}, then we
x_{10} or x_{11} instead. Going further d



o and its preimages x_9, x_8,ldots, x_0.
rapidly moves away from the target
es perturbations that move the point
$\hat{y}_2$ of x_2. The trajectory starting from
is represented by the point $\hat{y}_{10}$. The
ns of the stable manifolds associated

$= 1, \qquad \ell_1 = \frac{1}{\sqrt{2}}$

aking as the nominal value $\rho = \bar{\rho} = 9$.
mes iterated point x_0, *i.e.*, the point
ng from x_0. The double rotor map is
ite of x under the inverse map. The
with the kick set to its nominal value
nap applied to x with the kick set to ρ.
cobian matrix of partial derivatives of

uppose that T is the target point. A
uence of its preimages. The basic idea
eled as x_{10} to emphasize the idea that
nages.
d a point y_0 that falls near x_0. We wish
vailable parameter such that the orbit
at x_0. (As the dynamics are chaotic,
We now outline the basic targeting
in the case of the double rotor map) is

iple we could compute four successive
us to hit the point x_4 starting from y_0,
$_1, \ldots, \partial F/\partial\rho_3$ are linearly independent.
r, and it is not possible to find a suffi-
s method can be applied to determine

the perturbations needed to hit x_4 unless y_0 is extremely close to x_0. Instead, we adopt an alternative approach. (For the moment, let us ignore the presence of noise and the effects of numerical roundoff errors.)

For the parameter values given above, numerical results show that the attractor has two positive and two negative Lyapunov exponents. (A numerical estimate of the Lyapunov exponents of the double rotor attractor can be obtained using the algorithm of Benettin *et al.* [9]) As a result, associated with a typical point on the attractor is a 2-dimensional unstable manifold and a 2-dimensional stable manifold. For example, the stable manifold associated with the point x_2 in Fig. 3 is labeled S_2. If $s \in S_2$, then the orbit starting from s approaches the orbit starting from x_2 as the map is iterated. In general, the rate of approach is rapid because the negative Lyapunov exponents associated with the attractor are fairly large in absolute value. The goal of the targeting algorithm is to try to determine perturbations to the parameters to move the orbit starting from y_0 onto the stable manifold associated with one of the x's. If it is successful, then the dynamics draw the orbits closer together.

In the case where one parameter can be varied, we attempt to find two perturbations $\delta\rho_0$ and $\delta\rho_1$ so that the orbit starting from y_0 is moved onto S_2, the stable manifold of x_2. (Recall that S_2 is a 2-dimensional sheet. The vectors $\partial F(F(x_0, \rho_0), \rho_1)/\partial\rho_0$ and $\partial F(F(x_0, \rho_0), \rho_1)/\partial\rho_1$ typically are linearly independent. Since we are working in R^4, the 2-plane spanned by the gradient vectors intersects S_2 in a unique point, denoted by $\hat{y}_2$ in Fig. 3.)

A basic difficulty arises in approximating S_2. In general, $\hat{y}_2$ is far from x_2, and a linear approximation of S_2 is inadequate. In practice, a better approximation of S_2 can be found by calculating the inverse images of suitable points further down the path. For instance, a reasonable linear approximation of the stable manifold S_{10}, valid in a small neighborhood of x_{10}, can be determined by finding the stable eigenspace associated with the matrix $DF(x_2, \rho)DF(x_3, \rho) \cdots DF(x_{10}, \rho)$. Let s_0 and s_1 denote vectors that span this stable eigenspace at x_{10}. Let z be the point $z = x_{10} + \sigma_0 s_0 + \sigma_1 s_1$, where $|\sigma_i|$ is small. Although z may not lie exactly on S_{10}, the inverse images $F^{-1}(z)$, $F^{-2}(z)$, ... rapidly approach the corresponding sets S_9, S_8, ..., because S_{10} is an expanding set under the inverse map, and components perpendicular to S_{10} contract.

Thus, in the case of one parameter targeting, a successful solution to the steering problem consists of determining two parameter perturbations $\delta\rho_0$ and $\delta\rho_1$, together with values for σ_0 and σ_1, such that

$$F^{-8}(x_{10} + \sigma_0 s_0 + \sigma_1 s_1) = F(F(y_0, \rho_0), \rho_1) = \hat{y}_2. \tag{2}$$

Equation (2) can be solved numerically using Newton's method. There is no *a priori* guarantee that Newton's method will converge; however, when it does, we have determined two successive perturbations to the control parameter that steer the orbit of y_0 onto the stable manifold of x_2. Thereafter, in the absence of noise and numerical roundoff, the dynamics bring the two orbits closer together, typically at an exponential rate determined by the negative Lyapunov exponents.

There is nothing special about the choice of x_{10} and the use of eight inverse iterations to estimate S_2. For example, if the target point is x_6, then we can estimate S_2 by the fourth inverse iterate of a point close to x_6. If the target point is farther away, say at x_{12}, then we can look at the inverse images of a point near x_{10} or x_{11} instead. Going further down the path in this manner typically yields a point whose inverse image is a better approximation to the stable manifold S_2. However, the numerical solution of Eq. (2) is more ill-conditioned. For example, if we look at the inverse images of S_8, then it is necessary to evaluate the matrix product $DF^{-1}(x_8)DF^{-1}(x_7)\cdots DF^{-1}(x_3)$. If we look at S_{10} instead, then we must evaluate the matrix product $DF^{-1}(x_{10})DF^{-1}(x_9)\cdots DF^{-1}(x_3)$. These matrix products become more singular as more terms are added. Thus there is a tradeoff between numerical precision and approximation errors arising from the dynamics. For the parameters of the double rotor map used in this investigation, we have found that six inverse iterations is a good compromise. References [10] and [11] discuss some of the numerical issues in more detail.

If two parameters are available for control, then only one perturbation step is necessary. This is because typically there is a 2-plane through y_1, spanned by $g_p = \partial F(y_0,p,q)/\partial p$ and $g_q = \partial F(y_0,p,q)/\partial q$, that intersects the stable manifold S_1 of x_1. The procedure outlined above can be easily extended to other maps in different phase space dimensions and with different numbers of positive Lyapunov exponents. For example, if the attractor sits in a 6-dimensional space and has three positive Lyapunov exponents, then the procedure requires three successive changes to a single parameter to hit the 3-dimensional stable manifold of the appropriate point in the path. If three parameters can be varied independently, then one tries to hit the stable manifold of x_1, and so on.

Because of small errors in the initial approximation of S_8 and numerical roundoff errors, the control described above must be repeated periodically in order to keep the new trajectory close to the path leading to the target. We recalculate the perturbations at each iteration along the path where possible in order to allow for the presence of noise and/or roundoff error. (Recalculation of the control is not possible, for example, at the point just before the end of a path when only one control parameter is being used.)

4 Trees

The procedure described in the previous section works well, but the map must be iterated a large number of times before reaching a neighborhood of one of the points in the path leading to the target. A long path increases the likelihood that a given iterate lies near a point on the path, but the time required to reach the target also increases. Our objective is to steer a typical iterate to the target point in as few steps as possible.

One refinement is to build a hierarchy or "tree" of paths leading to the target. Let a target point x_T be given, together with a "root" path $x_0, x_1, \ldots, x_{T-1}$ leading to it. (In the work described in Refs. [10] and [11], each path typically has about 20 points.) The map is iterated (say from an arbitrary initial condition in the basin of attraction) until a point z_n is found that lies in a suitably small neighborhood of one of the points in the target path. The path leading to z_n (that is, z_n and 20 preimages) is stored in the tree; this path leads to the root path, and forms part of the first level of the tree. A path leading to a neighborhood of one of the preimages of z_n would be in the second level of the tree, and so on. Figure 4 gives a schematic illustration of the procedure.

By storing the paths in a tree, we increase the probability that a given iterate on the attractor lies near a path that can be steered to the target point. For example, if

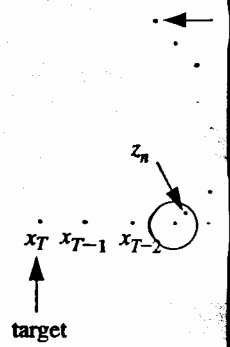


Figure 4: Schematic illustration of t

an iterate lies on the path leading to
z_n as the target point. Once a sma
algorithm is reapplied to steer the tra
which is the ultimate target.

A basic advantage of the tree s
needed to reach x_T grows linearly
number of points that can be stored
It is possible to construct "leafy" tre
With a suitable tree, one does not
near a path in the tree. Similarly, if
instance, if Newton's method fails to
of the trajectory. However, it is no
falls near some other point in the
algorithm once again. Reference [11
used to construct an "optimal" tree.

In the results described below, w
stored in paths to a depth of three
no more than 60 steps are required
constructed for each target point of

Once the tree is built, it is possi
follows. Let z_0 be a point on the a
in the path tree, then we create a r
perturbations to the kick. Here $A =$
η_i is a random variable in a small i
uniform distribution in the interval [

We now check whether any of t
tree. If so, then the targeting proced
steered the point z_0 to the target in
the random kick, followed by no mo
of the points in A can be iterated (
them can be steered successfully to t

Trees can be useful in other cont
certain regions of the attractor. (Per

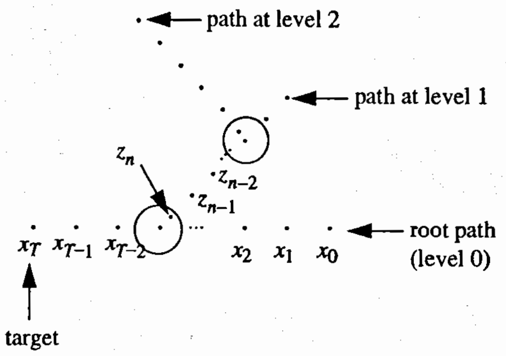


Figure 4: Schematic illustration of the hierarchy of paths leading to the target point.

an iterate lies on the path leading to z_n, the targeting algorithm can be applied with z_n as the target point. Once a small neighborhood of z_n is reached, the targeting algorithm is reapplied to steer the trajectory to a small neighborhood of the point x_T, which is the ultimate target.

A basic advantage of the tree structure is that the maximum amount of time needed to reach x_T grows linearly with the number of levels in the tree, but the number of points that can be stored grows exponentially with the number of levels. It is possible to construct "leafy" trees that reach into many regions of the attractor. With a suitable tree, one does not need to wait very long before some iterate lies near a path in the tree. Similarly, if the targeting algorithm fails at some point (for instance, if Newton's method fails to find a solution of Eq. (2)), then we lose control of the trajectory. However, it is not long before the uncontrolled trajectory again falls near some other point in the tree, whereupon we can attempt the targeting algorithm once again. Reference [11] describes some of the procedures that can be used to construct an "optimal" tree.

In the results described below, we use trees that typically consist of 10,000 points stored in paths to a depth of three levels. (Thus, if each path has length 20, then no more than 60 steps are required to reach the target point.) A separate tree is constructed for each target point of interest.

Once the tree is built, it is possible to steer points to the target very quickly, as follows. Let z_0 be a point on the attractor. If z_0 is not close to any of the points in the path tree, then we create a new set of points A by making n small random perturbations to the kick. Here $A = \{z_1^{(i)} : z_1^{(i)} = F(z_0, \rho_0 + \eta_i), 1 \leq i \leq n\}$ where η_i is a random variable in a small interval around 0. Typically we take η_i from a uniform distribution in the interval $[-0.05, 0.05]$.

We now check whether any of the points in A lies near any of the points in the tree. If so, then the targeting procedure is attempted. If it is successful, then we have steered the point z_0 to the target in no more than 61 steps (the first step consists of the random kick, followed by no more than 60 steps of the control procedure). Each of the points in A can be iterated (using the nominal value of the kick) until one of them can be steered successfully to the target.

Trees can be useful in other contexts. Suppose for instance that we want to avoid certain regions of the attractor. (Perhaps some regions of the phase space correspond

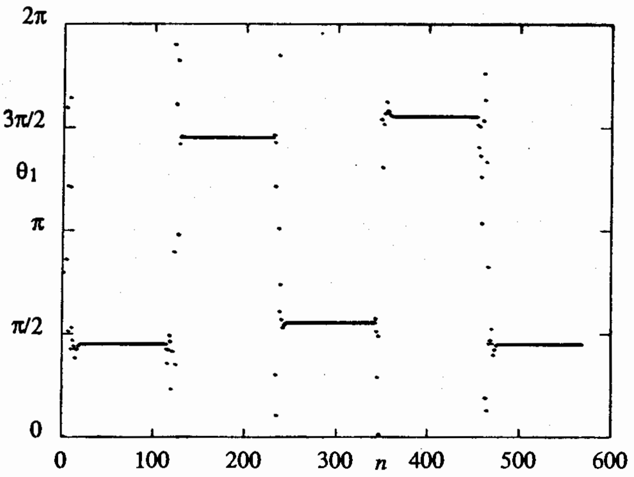


Figure 5: Control of stable periodic points with targeting.

to an undesirable operating regime of the system.) It may be possible to construct trees whose paths lie outside such a region, thus keeping the controlled system within an acceptable operating regime.

Similarly, it is possible in principle to associate a cost function with each of the paths in the tree. In cases where there are two or more sets of paths that lead to the target, one can select the set with the least total cost.

5 Results

Figure 5 shows the results of using the targeting algorithm together with control of saddle periodic points. We have selected four saddle fixed points on the attractor. We then pick four different target points, each of which is in a small neighborhood of the fixed points. A tree of paths leading to each target point is constructed as described above. The uncontrolled process quickly leads to a point that lies near some point in the tree. We apply the targeting algorithm to steer the trajectory to a neighborhood of the saddle fixed point. A control algorithm (see [11]) is then used to stabilize the trajectory near the fixed point. After the control is turned off, the trajectory quickly wanders away from the fixed point, but it approaches another point on the path tree, so the trajectory can be targeted to a neighborhood of the next saddle fixed point. In this way, we can rapidly switch between different saddle fixed points. In contrast to the case described in Fig. 2, where wait times of several thousand iterates are common between controlled states, targeting allows us to reduce the waiting time to as little as 12–16 iterates. Thus, the targeting algorithm can reduce the waiting time by two orders of magnitude.

Of course, there can be significant computational costs associated with building the path trees. In the results described here, 1–10 million iterates are required to

build the trees leading to the sadd
will dictate in practice whether the
procedures outweighs the effort neec

Even so, the necessary comput
modern workstation (less than five r
Graphics Indy). Memory requireme
the trees. Obviously, the computatic
the nature of the dynamical system,
of the phase space.

The effect of noise on the targ
case of the double rotor map, the
in targeting are quickly amplified b
somewhat sensitive to noise. Howev
do not appear to present serious diff
of two or more control parameters i
one control parameter. See Ref. [11]

6 Conclusions

We conclude with some brief remar
papers presented at the conference.
sensitive dependence on initial cond
dependence makes long-term predict
presence of noise or measurement un
Someday, chaos may be seen as a de
system can be switched from one st

Professor Judd has suggested th
be operated in a "pseudo-periodic"
are directed from one small region of
bations. (The trajectories do not n
trees as described in this article can
collection of targets that forms a ps
can be applied to each target in the

Acknowledgments

This work is supported by the Depa
E. K. is supported in part by the De
Many of the computations were don
the University of Maryland.

[1] E. Ott, C. Grebogi and J. A. Y

[2] W. L. Ditto, S. N. Rauseo and

[3] T. Shinbrot, E. Ott, C. Grebo
3250.

build the trees leading to the saddle fixed points. The nature of the application will dictate in practice whether the time savings from repeated use of the targeting procedures outweighs the effort needed to build the trees.

Even so, the necessary computations to build the trees are easily done on a modern workstation (less than five minutes of CPU time on a DEC Alpha or Silicon Graphics Indy). Memory requirements are also modest, generally 4–5 megabytes for the trees. Obviously, the computational and memory requirements depend heavily on the nature of the dynamical system, the dimension of the attractor, and the dimension of the phase space.

The effect of noise on the targeting algorithms remains under study. In the case of the double rotor map, the Lyapunov exponents are large, and small errors in targeting are quickly amplified by the dynamics. Thus the targeting algorithm is somewhat sensitive to noise. However, noise levels on the order of 1 percent or less do not appear to present serious difficulties. Preliminary results suggest that the use of two or more control parameters is more robust in the presence of noise than just one control parameter. See Ref. [11] for more details.

6 Conclusions

We conclude with some brief remarks on the relationship of this work to the other papers presented at the conference. One of the basic themes of the conference is the sensitive dependence on initial conditions of many nonlinear processes. While such dependence makes long-term predictions of such systems difficult or impossible in the presence of noise or measurement uncertainties, it can be made to work in one's favor. Someday, chaos may be seen as a desirable design feature because the behavior of the system can be switched from one state to another with only small perturbations.

Professor Judd has suggested that chaotic dynamics may allow some systems to be operated in a "pseudo-periodic" fashion, wherein trajectories within an attractor are directed from one small region of the attractor to another using only small perturbations. (The trajectories do not necessarily constitute a periodic orbit.) The path trees as described in this article can be adapted for this purpose. All one needs is a collection of targets that forms a pseudo-periodic orbit, and the targeting algorithm can be applied to each target in the sequence.

Acknowledgments

This work is supported by the Department of Energy Office of Scientific Computing. E. K. is supported in part by the Dept. of Energy Program in Applied Mathematics. Many of the computations were done at the Keck Chaos Visualization Laboratory at the University of Maryland.

References

[1] E. Ott, C. Grebogi and J. A. Yorke, *Phys. Rev. Lett.* **64** (1990), 1196.

[2] W. L. Ditto, S. N. Rauseo and M. L. Spano, *Phys. Rev. Lett.* **65** (1990), 3211.

[3] T. Shinbrot, E. Ott, C. Grebogi and J. A. Yorke, *Phys. Rev. Lett.* **65** (1990), 3250.

[4] T. Shinbrot, W. Ditto, C. Grebogi, E. Ott, M. Spano and J. A. Yorke, *Phys. Rev. Lett.* **68** (1992), 2863.

[5] E. J. Kostelich, C. Grebogi, E. Ott and J. A. Yorke, *Physica D* **25** (1987), 347; *Phys. Lett. A* **118** (1986), 448; erratum **120A** (1987), 497.

[6] F. J. Romeiras, C. Grebogi, E. Ott and W. P. Dayawansa, *Physica D* **58** (1992), 165.

[7] J. L. Kaplan and J. A. Yorke, in *Functional Differential Equations and Approximation of Fixed Points*, ed. by H.-O. Peitgen and H.-O. Walter. Berlin: Springer-Verlag, Lecture Notes in Mathematics, Vol. 730, p. 204.

[8] J. D. Farmer, E. Ott and J. A. Yorke, *Physica D* **7** (1983), 153.

[9] G. Benettin, L. Galgani, A. Giogilli, and J.-M. Strelcyn, *Meccanica* **15** (1980), 9, 21.

[10] E. J. Kostelich, C. Grebogi, E. Ott and J. A. Yorke, *Phys. Rev. E* **47** (1993), 305.

[11] E. Barreto, C. Grebogi, E. Ott, and J. A. Yorke, *Phys. Rev. E* **51** (1995), 4169.

Commentary by K. Glass

This paper looks at the control problem of targeting points in a chaotic system. The introduction of the paper outlines the targeting problem and stresses the value of using targeting when stabilizing periodic points. Stabilization algorithms are discussed in the paper in this volume by E. Ott et al.

The third section of the paper outlines in detail the procedure used to target a point in a chaotic system. This procedure involves finding the stable manifold of some point on the trajectory leading to the target point and then attempting to direct the system onto this manifold. A tree is then built up of sections of trajectories leading to these manifolds. Although creating the tree of paths will require some off-line computation, the resulting algorithm will be highly efficient. Not only will the target point be reached after only a few steps, the perturbation required to target the system may be calculated quickly making the method effective for on-line control.

The concept of storing sections of trajectories in a tree of paths would be highly useful in the later sections of the paper: "Creating and Targeting Periodic Orbits". I feel also that it may be possible to use the idea of creating trajectories discussed here to extend the tree of paths used in the above algorithm. By creating trajectories using small perturbations, we might be able to construct paths in less accessible areas of the system.

Commentary by I. Mareels

The contribution by Kostelich and Barreto discusses in the context of a double rotor example a method to achieve fast transitions between orbits on a strange attractor by the aid of small amplitude control.

Under the generic condition that
attractor the minimal transition time
beat control action (to use control eng
is proposed. (The minimum time op
optimal control problem.)

The authors propose a numerica
actions. Essentially a tree structure
is oncstructed off line.

Questions to authors

1. could you do a small calculati
tree would roughly give you a transiti
the fractal dimension it seems to com

2. Why are most paths without c
action? CPU time savings? Could yo

)tt, M. Spano and J. A. Yorke, *Phys.*

. A. Yorke, *Physica D* **25** (1987), 347;
120A (1987), 497.

V. P. Dayawansa, *Physica D* **58** (1992),

tional Differential Equations and Ap-
-O. Peitgen and H.-O. Walter. Berlin:
matics, Vol. 730 , p. 204.

hysica D **7** (1983), 153.

l J.-M. Strelcyn, *Meccanica* **15** (1980),

J. A. Yorke, *Phys. Rev. E* **47** (1993),

. Yorke, *Phys. Rev. E* **51** (1995), 4169.

rgeting points in a chaotic system. The
; problem and stresses the value of using
abilization algorithms are discussed in

1 detail the procedure used to target a
lves finding the stable manifold of some
)oint and then attempting to direct the
lt up of sections of trajectories leading
ree of paths will require some off-line
ighly efficient. Not only will the target
turbation required to target the system
d effective for on-line control.
ories in a tree of paths would be highly
eating and Targeting Periodic Orbits".
idea of creating trajectories discussed
ove algorithm. By creating trajectories
) construct paths in less accessible areas

scusses in the context of a double rotor
; between orbits on a strange attractor

Under the generic condition that the map is controllable at each point on the attractor the minimal transition time control problem is well posed. A repeated dead beat control action (to use control engineering lingo) with amplitude bounded control is proposed. (The minimum time optimal control problem is considered a difficult optimal control problem.)

The authors propose a numerical method to generate the appropriate control actions. Essentially a tree structure of paths which covers adequately the attractor is oncstructed off line.

Questions to authors

1. could you do a small calculation as in the intro to why 10,000 points in the tree would roughly give you a transition time of 20 (I think it is illuminating) Using the fractal dimension it seems to come out quite nicely.

2. Why are most paths without control, and only the root path contains control action? CPU time savings? Could you comment on this?