Automated adaptive recursive control of unstable orbits in high-dimensional chaotic systems

M. A. Rhode, J. Thomas, and R. W. Rollins
Department of Physics and Astronomy, Condensed Matter and Surface Sciences Program, Ohio University, Athens, Ohio 45701-2979

A. J. Markworth
Department of Materials Science and Engineering, Ohio State University, Columbus, Ohio 43210-1179
(Received 29 July 1996)

We develop and demonstrate an automated control strategy using an adaptive learning algorithm that can control and track periodic orbits even if they are completely unstable, i.e., have no stable manifolds. The control system is designed to operate in real time, taking time series measurements of a single variable as input and providing as output the control parameter value required to stabilize the desired unstable periodic orbit (UPO). The control scheme directs the system to the fixed point itself rather than a stable manifold and works when the the unstable Lyapunov multipliers are relatively large (≈ 6). The learning and control algorithm uses a time delay embedding with the full state vector collected within one period of the controlled orbit. Control is achieved by small perturbations of a single control parameter once each cycle using a control algorithm with one recursive term. A simulation is used to study the application of the control algorithm to the hyperchaotic Rössler system. The simulation demonstrates both control of a highly unstable UPO and tracking the UPO as system parameters slowly drift over a wide range. The difficulties encountered in tracking with recursive control are discussed. [S1063-651X(96)14911-4]

PACS number(s): 05.45.+b, 87.10.+e

I. INTRODUCTION

Perhaps one of the most promising applications of the recent advances in nonlinear dynamics is in the area of system control. Nonlinear dissipative systems typically possess dynamics that have a sensitive dependence on initial conditions. These systems settle into a deterministic ergodic wandering motion on a chaotic attractor in state space and the attractor usually has an infinite set of unstable periodic orbits (UPOs) embedded within it. A small change in a system parameter shifts slightly the position of the attractor in the state space and, as the system attempts to follow the attractor, the sensitive nature of the dynamics in the chaotic regime can cause large changes in its subsequent motion. Many feedback strategies based on this general idea use small perturbations in a control parameter to manipulate the behavior of chaotic systems. In 1990, Ott, Grebogi, and Yorke (OGY) [1] described a closed loop feedback control scheme that stabilizes an UPO in a low-dimensional system with one unstable direction. The work of OGY spawned much research (both experiment and theory) on the control of chaotic systems. The recent review by Ott and Spano [2] provides a good general overview of the field and there are several more in-depth reviews [3]. Because nonlinear systems are ubiquitous, various control methods have been applied to systems found in diverse scientific fields and there are several general reviews written from the point of view of particular disciplines [4].

Until recently, most of the experimental work on controlling chaos was done in systems that were highly dissipative, low-dimensional attractors and the stabilized UPO was a simple saddle orbit with a single unstable direction. Recently So and Ott [5] and Petrov *et al.* [6] developed new feedback control schemes, derived from a delay coordinate reconstruction of the state space, to stabilize an UPO with more than one unstable direction. The use of delay coordinates leads to recursive control equations, i.e., the control used during the next cycle depends not only on the deviation of the system from the desired fixed point in state space but also on the control parameter deviations used on one or more previous cycles. Building on this work, Ding *et al.* [7] developed a high-dimensional control method designed for relatively easy experimental implementation and applied it to control a driven magnetoelastic ribbon with one moderately unstable direction and two stable directions. The simulations by both Petrov *et al.* and Ding *et al.* stabilized orbits with more than one unstable direction where the largest Lyapunov multiplier was usually <2. Generally, these implementations of high-dimensional control use precontrol experiments to determine the control parameters and the control is applied with system parameters fixed at their nominal values at the operating point. Petrov *et al.* also demonstrated the ability to track and control through slow changes in a system parameter but in a situation where a delay coordinate embedding, and hence a recursive control equation, was avoided.

In this paper, we develop a tailored control scheme that extends the high-dimensional control capability in several ways. Larger unstable Lyapunov multipliers (≈ 6) can be accomodated and, more importantly, an automated tracking procedure is developed for use with high-dimensional recursive control schemes. The adaptive nature of the control resulting from the algorithm can be very important in controlling high-dimensional systems as we describe below. The control system includes a learning algorithm that automatically probes the system dynamics using time-delay embedding of a single scalar variable to reconstruct a state space in the neighborhood of the UPO, implements a control scheme stabilizing the UPO, and continually adapts to hold the control and track the UPO while slow system drifting takes place.

Both the automation and the ability to adapt and track the UPO are very useful, or even necessary, properties of any control system that is to be successful in a real application where the desired operating point is an UPO with many unstable directions. This is not solely because there are uncontrollable drifts present in most experimental situations. A system, wandering on a high-dimensional chaotic attractor, may seldom come close to the desired UPO and the time required to learn the local dynamics from these visitations could be prohibitive. This problem was recognized by So and Ott [5] and they suggest that some kind of targeting techniques [8,9] may be essential in controlling high-dimensional systems. Targeting techniques require a more global understanding of the system dynamics and would be generally quite difficult to obtain in most high-dimensional systems. We suggest that the most practical solution to this problem is to start the system in a region of parameter space where the attractor has lower dimension and the desired UPO has fewer unstable directions, or is even stable. If it is possible to find such a region in parameter space, then it is easy to learn the local dynamics and stabilize the UPO there. Once stabilized, we can slowly, and purposefully, change the parameters to move the system to the desired operating point where the attractor is high-dimensional (or may not exist at all) yet maintaining our automated adaptive control on the UPO all the way to the operating point. Our proposed method relies only on learning local dynamics and does not require any knowledge of the global dynamics of the system.

The next section provides a brief review of relevant recent research to provide a context for our work. In the subsequent sections we develop the control scheme, describe our adaptive learning strategy, and present the results of a simulation using the Rössler hyperchaotic model as an example test system.

A. Background

The original OGY method stabilized unstable periodic saddle orbits with one unstable direction in a low-dimensional state space by directing the system to the stable manifold of the desired UPO. The information necessary for control was obtained by linearizing the dynamics of the system in the neighborhood of the UPO in a true state space of the system. The method was generalized to high-dimensional state spaces by Romeiras *et al.* [10] and by Ott *et al.* [11]. Often it is not possible to obtain measurements of enough independent variables to use the true state space to determine the necessary dynamical information for feedback control of a chaotic system. Dressler and Nitsche [12,13] and Auerbach *et al.* [14] showed that the necessary dynamical information can be obtained from a time-delay embedding using time series measurements of a single scalar variable. However, the use of time-delay coordinates adds one or more recursive terms to the feedback control equation. When the system is highly dissipative the control equation can be written in the general form [15] $\delta p_n = K\delta x_n + R\delta p_{n-1}$, where δx_n is the deviation of the measured scalar variable from the desired fixed point at the beginning of the nth Poincaré cycle and δp_n is the deviation of the control parameter from its nominal value during the nth cycle, and K and R are scalar constants that depend on the system dynamics in the neighborhood of the fixed point of the Poincaré map for the UPO. This simple control scheme was referred to as recursive proportional feedback (RPF).

Recent work [5–7] generalized the feedback control scheme based on delay coordinate measurements to stabilize an UPO with more than one unstable direction embedded in a high-dimensional state space. So and Ott give a very general treatment where the system is directed to the stable manifold of the UPO and the delay time for the embedding is such that the state vector stretches back in time over r cycles. Their result is a control scheme that will bring the system to the stable manifold of the UPO in u cycles, where u is the number of unstable directions for the UPO. More precisely, u is the number of Lyapunov multipliers with magnitude greater than unity for the Poincaré map evaluated at the fixed point of the UPO. Each of the u control equations determining the control parameter for the next u iterates depends on the d-dimensional delay coordinate state vector at the start of the first control cycle and the past r values of the control parameter.

The algorithms proposed by Petrov *et al.* and Ding *et al.* are similar to those of So and Ott if we take the delay time equal to the period of the UPO so that one component of the time delay coordinate vector is obtained each period. Thus, it will take d periods to collect a complete state vector in this case and there will be d recursive terms in the u step feedback control. The dimension d of the delay coordinate vector required to effect control is hardly ever known *a priori* for a given experimental system. It is interesting to note that Petrov *et al.* demonstrate that one can ''experimentally'' determine d by starting with a low value of d and increasing it until control can be achieved. It is also clear that if one of the Lyapunov multipliers, say Λ_m, for the UPO is large, then waiting d periods to collect a state vector may not provide a good description of the state of the system in the presence of noise. Furthermore, assuming the system must remain within Δx of the UPO in order for the linearized dynamics to be valid, the system would have to start well within a distance $\Lambda_m^{-d}\Delta x$ of the fixed point to obtain a valid state vector in delay-coordinate space evolving according to the linearized dynamics. For similar reasons it would not be advisable to calculate the next u control perturbations and then actually use these perturbations for the next u iterations. In fact, the method recommended is to use the first control equation to calculate the δp to be applied for the first of the u iterates and then, after taking that one step, repeat the first step calculation using the updated information. This process is repeated over and over to help eliminate the adverse amplification of system noise and measurement errors on the control. We show below that the repeated application of the first control equation in the absence of noise is exactly equivalent to the complete u step control.

In the sections below we first develop a control scheme that is similar to a special case of the So and Ott methods. We aim to simplify the control scheme and to make it less sensitive to large Lyapunov multipliers. We then describe an automated adaptive learning algorithm that can be continuously applied to a system using a measured time series as input and supplying the appropriate control parameter to be used over the next control period in order to successfully stabilize the desired UPO. Finally, we describe a simulation using the learning algorithm to control a test system with dynamics governed by Rössler's four-dimensional system that has a hyperchaotic attractor.

II. DEVELOPMENT OF d-DIMENSIONAL RPF CONTROL

A. Introduction

Consider a continuous time dynamical system that lives in a $(d+1)$-dimensional state space. To reduce the continuous-time dynamical system to discrete time system we choose a Poincaré surface and measure a single scalar variable $x(t)$ whenever the trajectory crosses the Poincaré section. Let t_n be the time at the nth crossing and $\boldsymbol{u}_n$ represent the position in phase space at the nth crossing of the Poincaré section. Since the system is deterministic, the state at time t_{n+1} is uniquely determined by the state at time t_n. This gives the d-dimensional Poincaré map $\boldsymbol{G}$ relating the states at successive iterative crossings as

$$\boldsymbol{u}_{n+1}=\boldsymbol{G}(\boldsymbol{u}_n,p_n), \tag{1}$$

where p_n is the value of the parameter p during the cycle starting at state $\boldsymbol{u}_n$.

Let $x_n=x(t_n)$ be the value of a scalar variable x observed at the start of the nth Poincaré cycle. We reconstruct the system dynamics at the Poincaré section from this scalar by replacing the vector $\boldsymbol{u}_n$ with the delay-coordinate vector $\boldsymbol{x}_n$

$$\boldsymbol{x}_n=\begin{pmatrix} x(t_n) \\ x(t_n-\tau) \\ x(t_n-2\tau) \\ \vdots \\ x[t_n-(d-1)\tau] \end{pmatrix}, \tag{2}$$

where the number of delay coordinates should be large enough to reproduce the important dynamics of the system. We intend to change a single scalar control parameter once each Poincaré cycle to control the system. As pointed out by Dressler and Nitsche [12,13], the use of delay coordinates increases the effective dimension of the state space because the evolution of the system in the delay coordinate space not only depends on the delayed coordinate itself but also depends on the value of the control parameter when that coordinate was measured. To simplify this situation, we assume that $|t_n-t_{n-1}|>(d-1)\tau$ and thereby limit the number of recursive control parameter values needed to determine the future of the system to one. In this case, the Poincaré map in the time-delayed space is of the form

$$\boldsymbol{x}_{n+1}=\boldsymbol{F}(\boldsymbol{x}_n,p_{n-1},p_n). \tag{3}$$

We now assume that the system has a fixed point $\boldsymbol{x}=\boldsymbol{x}_{fp}$ for $p_{n-1}=p_n=p_0$ and consider small deviations from the fixed point by defining $\delta\boldsymbol{x}_n\equiv(\boldsymbol{x}_n-\boldsymbol{x}_{fp})$ and $\delta p_n\equiv(p_n-p_0)$. Expanding Eq. (3) to first order in $\delta\boldsymbol{x}_n$, δp_n, and δp_{n-1}, the system dynamics near the fixed point is described by

$$\delta\boldsymbol{x}_{n+1}=\hat{\mathbf{M}}\delta\boldsymbol{x}_n+\boldsymbol{v}\delta p_{n-1}+\boldsymbol{w}\delta p_n, \tag{4}$$

where $\hat{\mathbf{M}}$ is a $(d\times d)$ Jacobian matrix evaluated at the fixed point, $\boldsymbol{v}$ and $\boldsymbol{w}$ are d-dimensional vectors given by

$$\boldsymbol{v}=\frac{d\boldsymbol{F}}{dp_{n-1}}\Big|_{\delta x,\delta p=0},\quad \boldsymbol{w}=\frac{d\boldsymbol{F}}{dp_n}\Big|_{\delta x,\delta p=0}. \tag{5}$$

In what follows we present a d-dimensional recursive proportional feedback algorithm and show that by repeatedly applying the next control parameter perturbation we can stabilize the system to its fixed point. We further develop an automated adaptive learning algorithm to obtain the information required for controlling the system on the unstable fixed point from a scalar measurement of time series. Finally, the adaptive recursive control algorithm is applied in a simulation using Rössler's model of a hyperchaotic system.

B. Control strategy

Given $\delta\boldsymbol{x}_n$ and δp_{n-1} we intend to show that the system can be stabilized to the fixed point $\boldsymbol{x}_{fp}$ in $d+1$ steps. Iterating the map, given by Eq. (4), $(d+1)$ times we obtain

$$\delta\boldsymbol{x}_{n+(d+1)}=\hat{\mathbf{M}}^d(\hat{\mathbf{M}}\delta\boldsymbol{x}_n+\boldsymbol{v}\delta p_{n-1})+\hat{\mathbf{M}}^{d-1}(\hat{\mathbf{M}}\boldsymbol{w}+\boldsymbol{v})\delta p_n + \cdots +(\hat{\mathbf{M}}\boldsymbol{w}+\boldsymbol{v})\delta p_{n+d-1}+\boldsymbol{w}\delta p_{n+d}. \tag{6}$$

Equation (6) can be written in a more compact form,

$$\delta\boldsymbol{x}_{n+(d+1)}=\hat{\mathbf{M}}^d(\hat{\mathbf{M}}\delta\boldsymbol{x}_n+\boldsymbol{v}\delta p_{n-1})+\hat{\mathbf{D}}\boldsymbol{\delta}\mathbf{p}_n+\boldsymbol{w}\delta p_{n+d}, \tag{7}$$

where $\hat{\mathbf{D}}$ is a $d\times d$ matrix, consisting of column vectors $\hat{\mathbf{M}}^i(\hat{\mathbf{M}}\boldsymbol{w}+\boldsymbol{v})$, for $i=d-1,\ldots,0$:

$$\hat{\mathbf{D}}=[\hat{\mathbf{M}}^{d-1}(\hat{\mathbf{M}}\boldsymbol{w}+\boldsymbol{v}):\cdots:(\hat{\mathbf{M}}\boldsymbol{w}+\boldsymbol{v})] \tag{8}$$

and

$$\delta\boldsymbol{p}_n=\begin{pmatrix} \delta p_n \\ \delta p_{n+1} \\ \vdots \\ \delta p_{n+d-1} \end{pmatrix}. \tag{9}$$

Now, we obtain a set of d linear algebraic equations for the control parameters $\delta\boldsymbol{p}_n$ by requiring $\delta\boldsymbol{x}_{n+(d+1)}=0$ and $\delta p_{n+d}=0$. The control equation obtained is

$$\delta\boldsymbol{p}_n=\hat{\mathbf{K}}\delta\boldsymbol{x}_n+\mathbf{R}\delta p_{n-1}, \tag{10}$$

where $\hat{\mathbf{K}}=-\hat{\mathbf{D}}^{-1}\hat{\mathbf{M}}^{d+1}$ and $\mathbf{R}=-\hat{\mathbf{D}}^{-1}\mathbf{M}^d\boldsymbol{v}$. This result is a generalization of the previous recursive proportional feedback control equation [15] to the high-dimensional case.

Thus, given any $\delta\boldsymbol{x}_n$, δp_{n-1} at the nth step, we can bring the system to the fixed point in $(d+1)$ steps by using control parameter perturbations δp_n, $\delta p_{n+1},\ldots,\delta p_{n+d-1}$ given by the general control equation (10), for the first d steps and finally using $\delta p_{n+d}=0$ on step $(d+1)$. Furthermore, since the control parameter perturbation used on the last step was zero ($\delta p_{n+d}=0$), the system would stay, in the absence of noise, perfectly balanced on the fixed point with all subsequent control perturbations set to zero. Note also that this procedure will bring the system to the fixed point in $(d+1)$ steps even if there was no stable manifold. Of course, noise will be present in any real system and the above procedure could be applied over and over to stabilize the system on the fixed point.

Strict application of the above strategy would suggest we must calculate ahead $(d+1)$ steps in order to bring the system to the fixed point. If one or more of the unstable multipliers is large in magnitude, or if the noise is not small, it may not be possible to determine the system dynamics with enough precision to calculate $(d+1)$ steps forward. We show in the next section that a procedure that recalculates after each step and always applies the control perturbation for the first step in each $(d+1)$ step sequence is completely *equivalent*, in the absence of noise, to using the original $(d+1)$ step sequence. Thus, the latter procedure is preferable in the presence of noise.

C. Equivalence of first-step control

We now show that it is only necessary to find the control equation for the next step, that is,

$$\delta p_n = \boldsymbol{k}_1^T \delta \boldsymbol{x}_n + R_1 \delta p_{n-1}, \tag{11}$$

where $\boldsymbol{k}_1^T$ is a d-dimensional vector consisting of the first row of $\hat{\mathbf{K}}$ and R_1 is the first component of $\mathbf{R}$ in Eq. (10). Repeated application of control perturbations calculated using Eq. (11) will bring the system to the fixed point in $d+1$ steps.

We assume that neither $\boldsymbol{w}$ nor $\boldsymbol{v}$ is equal to an eigenvector of the matrix $\hat{\mathbf{M}}$ and that $\hat{\mathbf{M}}$ does not have degenerate eigenvalues. We define a set of basis vectors (not necessarily orthogonal)

$$\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_d \tag{12}$$

that span the d-dimensional space, where

$$\boldsymbol{y}_i = \hat{\mathbf{M}}^{-1} \boldsymbol{y}_{i-1}, \quad \text{and } \boldsymbol{y}_0 = \hat{\mathbf{M}} \boldsymbol{w} + \boldsymbol{v}, \tag{13}$$

for $i=1, \ldots, d$. Thus, for any $\delta \boldsymbol{x}_n$ and δp_{n-1}, we can write

$$\hat{\mathbf{M}} \delta \boldsymbol{x}_n + \boldsymbol{v} \delta p_{n-1} = -(a_1 \boldsymbol{y}_1 + a_2 \boldsymbol{y}_2 + \ldots + a_d \boldsymbol{y}_d), \tag{14}$$

where $a_1, \ldots, a_d$ are real numbers. Using Eq. (13), the general control equation (10) becomes

$$\hat{\mathbf{M}}^{d-1} \boldsymbol{y}_0 (\delta p_n - a_1) + \hat{\mathbf{M}}^{d-2} \boldsymbol{y}_0 (\delta p_{n+1} - a_2) + \ldots + \boldsymbol{y}_0 (\delta p_{n+d-1} - a_d) = 0, \tag{15}$$

which has the unique solution

$$\delta p_{n+i-1} = a_i \tag{16}$$

for $i= 1, \ldots, d$. Thus, given $(\delta \boldsymbol{x}_n, \ \delta p_{n-1})$, there exists a unique set of coefficients $(a_1, \ldots, a_d)$ such that Eq. (14) is satisfied. And if we iterate the system $(d+1)$ times using the sequence of control perturbations $\delta p_{n+i-1} = a_i$, $i=1, \ldots, d$, for the first d steps and take $\delta p_{n+d}=0$ for the last step, then the system is brought to the fixed point, $\delta \boldsymbol{x}_{n+d+1} = 0$.

Now we show explicitly that repeated application of the first step of the control strategy gives the exact same sequence of control perturbations bringing the system to the fixed point in the same $(d+1)$ steps. We start with $(\delta \boldsymbol{x}_n, \delta p_{n-1})$ and the coefficients $(a_1, \ldots, a_d)$ such that Eq. (14) is satisfied. Applying $\delta p_n = a_1$ and iterating one step using Eq. (4) gives

$$\delta \boldsymbol{x}_{n+1} = \hat{\mathbf{M}} \delta \boldsymbol{x}_n + \delta p_{n-1} + \boldsymbol{w} a_1. \tag{17}$$

Using Eqs. (13) and (14) and remembering $\delta p_n = a_1$, we obtain from Eq. (17) the result

$$\hat{\mathbf{M}} \delta \boldsymbol{x}_{n+1} + \boldsymbol{v} \delta p_n = -a_2 \boldsymbol{y}_1 - a_3 \boldsymbol{y}_2 - \ldots - a_d \boldsymbol{y}_{d-1}. \tag{18}$$

Applying the same control strategy to the $(\delta \boldsymbol{x}_{n+1}, \delta p_n)$ starting point gives the next δp as the coefficient of y_1, or a_2 in this case, which is the same value obtained originally for the control perturbation on the second step. Thus, repeated application of the control strategy on each successive iteration and using the coefficient of y_1 each time as the next δp reproduces the same sequence of control perturbations as originally calculated. We have explicitly shown that the repeated application of the first step of the $(d+1)$ step control is equivalent to the $(d+1)$ step control in the absence of noise. We would expect the repeated application of the first step would be better in the presence of noise because corrections are recalculated much more often.

III. ADAPTIVE LEARNING ALGORITHM

The learning algorithm described below is a generalization of a recent strategy [16,17] based on simple neural networks. We must determine the linearized system dynamics in the neighborhood of the desired fixed point in order to implement the above control strategy. It is particularly important for high-dimensional systems that an automatic learning algorithm be used to obtain the required information about the system dynamics because an uncontrolled high-dimensional system will rarely visit the state space close to the desired orbit. An automated learning algorithm can follow changes in the system dynamics (''adapt'') as a system parameter is slowly changed. This capability can be used to greatly increase the feasibility of control on an unstable orbit in regions of parameter space where very few, or even none, of the directions in state space near the orbit are stable. This can be done by changing the system parameters until we find a region of parameter space where the desired orbit is more stable. Once the system is on the stable orbit, we slowly change a system parameter in order to bring the system to a desired operating state. While changing the system parameter the adaptive nature of the learning algorithm enables us to follow the changes in the system dynamics and thus maintain control. Using this strategy, the desired operating state may be an unstable fixed point that is not part of any attractor.

When applying our algorithm to a particular system, we assume that we have gained enough experience with the dynamics of the system under study to choose an appropriate Poincaré section and a desirable periodic orbit for control. The dimension d of the embedding space should be chosen as small as possible and yet remove false nearest neighbors in the neighborhood of the fixed point corresponding to the desired periodic orbit on the Poincaré section. The choice of the delay time τ is not critical so long as the condition $(t_n - t_{n-1}) > (d-1)\tau$ is satisfied. In practice, it is best to choose τ such that the delay-coordinate vector $\boldsymbol{x}_n$ has components distributed nearly uniformly throughout most of the time interval between Poincaré section crossings for the controlled periodic orbit. In general, the value of τ would not be changed by our adaptive learning algorithm, but in particular cases, the dynamics of the orbit may be such that τ can be chosen based on a particular signature of the orbit itself. The latter method has the advantage that it will automatically adapt with the control orbit and this is the method used in our simulation.

We assume the dynamics around the fixed point in linear order to be described by

$$\boldsymbol{x}_{n+1} = \hat{\mathbf{M}}\boldsymbol{x}_n + \boldsymbol{b} + \boldsymbol{v}\,\delta p_{n-1} + \boldsymbol{w}\,\delta p_n\,, \tag{19}$$

where the offset vector $\boldsymbol{b}$ must be included because we must determine the position of the desired fixed point. The position of the fixed point $\boldsymbol{x}_{fp}$, can be evaluated from Eq. (19) to give

$$\boldsymbol{x}_{fp} = (\hat{1} - \hat{\mathbf{M}})^{-1}\boldsymbol{b}. \tag{20}$$

Thus we must determine $d(d+3)$ scalar values (the d^2 components of the matrix $\hat{\mathbf{M}}$ and the three d-dimensional vectors $\boldsymbol{b}$, $\boldsymbol{v}$, and $\boldsymbol{w}$) in order to know the linearized system dynamics in the neighborhood of the desired fixed point of the Poincaré section and the position of the fixed point. Constructing a $(d+3)$-dimensional column vector of $\boldsymbol{x}_n$, 1, δp_{n-1}, δp_n, and a $d\times(d+3)$-dimensional matrix of $\hat{\mathbf{M}}$, $\boldsymbol{b}$, $\boldsymbol{v}$, $\boldsymbol{w}$, Eq. (19) can be written as

$$[\hat{\mathbf{M}}{:}\boldsymbol{b}{:}\boldsymbol{v}{:}\boldsymbol{w}]\begin{bmatrix} \boldsymbol{x}_n \\ 1 \\ \delta p_{n-1} \\ \delta p_n \end{bmatrix} = [\boldsymbol{x}_{n+1}]. \tag{21}$$

To implement the control, the system dynamics described by $\mathbf{M}$, $\boldsymbol{b}$, $\boldsymbol{v}$, and $\boldsymbol{w}$ must be determined. The adaptive algorithm waits for the system to have a close return, that is when the values of $\boldsymbol{x}_{n+1}$ and $\boldsymbol{x}_n$ are close to each other. For each close return, the current values of $\boldsymbol{x}_n$, δp_{n-1}, δp_n, $\boldsymbol{x}_{n+1}$ are recorded. Using a collection of such close returns, Eq. (21) can be expanded to

$$[\hat{\mathbf{M}}{:}\boldsymbol{b}{:}\boldsymbol{v}{:}\boldsymbol{w}]\begin{bmatrix} \boldsymbol{x}_n & \cdots & \boldsymbol{x}_k \\ 1 & & 1 \\ \delta p_{n-1} & & \delta p_{k-1} \\ \delta p_n & \cdots & \delta p_k \end{bmatrix} = \begin{bmatrix} \boldsymbol{x}_{n+1}^T \\ \vdots \\ \boldsymbol{x}_{k+1}^T \end{bmatrix}^T. \tag{22}$$

The second matrix henceforth will be referred to as the learning matrix. This system of linear algebraic equations can be solved uniquely if $(d+3)$ close returns are observed. In general, we collect $N > (d+3)$ incidents and use singular value decomposition (SVD) to obtain a linear least squares fit to find the best values for the elements of $\hat{\mathbf{M}}$ and vectors $\boldsymbol{x}_f$, $\boldsymbol{v}$, and $\boldsymbol{w}$. This will improve the determination of these values in the presence of noise. To allow for system drift, we erase the oldest close-return data ($\delta\boldsymbol{x}_i$, 1, δp_{i-1}, δp_i, $\delta\boldsymbol{x}_{i+1}$) from Eq. (22) and replace it with the most current close-return data. This ensures adaptability in the presence of system drift.

In order to simultaneously apply the above adaptive learning and control procedure it is essential to add a small amount of random noise to the control signal. Without this noise, the controlled system would be constrained to a hypersurface given by control equation (10), which is a subspace of the linearized system dynamics, Eq. (4). However, while applying control we need to simultaneously update the coefficients of Eq. (4). When the system is tightly controlled, it is constrained on a projection $\delta\boldsymbol{x}_{i+1}=0$, and does not exploit the full dimensionality of Eq. (4). Therefore, it becomes impossible to extract complete information about the linearized system dynamics, given by the coefficients ($\hat{\mathbf{M}}$,$\boldsymbol{w}$,$\boldsymbol{v}$) of Eq. (4), from successive controlled iterations. Stated in another way, the learning matrix in Eq. (22) matrix becomes singular, since the rows satisfy a linear dependency implied by the control equation (11). This can be resolved by adding small random noise to the control signal δp_n, such that the system is still controlled well enough to stay in the vicinity of the fixed point, yet successive iterations do not perfectly satisfy Eq. (11). The randomness of the additional control component guarantees that the system probes the full dimensionality of Eq. (4) and removes the singularity in the learning matrix allowing the determination of the system dynamics while applying control.

IV. SIMULATION: CONTROL OF THE HYPERCHAOTIC RÖSSLER SYSTEM

The first simple example of a system possessing an attractor with more than one unstable direction (hyperchaos) was provided by Rössler [18] by adding a fourth variable to a simple three-variable model of a chemical reaction scheme. The hyperchaotic Rössler system described by

$$\begin{aligned} \dot{x} &= -y - z, \\ \dot{y} &= x + .25y + w, \\ \dot{z} &= b + xz, \\ \dot{w} &= (-0.5 + p)z + .05w, \end{aligned} \tag{23}$$

has an attractor with two positive Lyapunov exponents for a range of the bifurcation parameter b. Rössler originally took $b=3$ and $p=0$ where we find the Lyapunov exponents of the attractor to be about (0.160, 0.0275, 0.000, -30.3) bits/unit-time using the Wolf *et al.* [19,20] algorithm. The hyperchaotic attractor contains a period-two UPO at these parameter values (see Fig. 1). This orbit has two unstable Lyapunov multipliers with the largest one greater than 5. Our attempts to apply control on this UPO directly were not successful because the system seldom came close enough to the orbit and the time required to determine the linearized dynamics in the neighborhood of the orbit was too long. However, for $b=13$, the period-two orbit is stable. Thus, the dynamics are easily obtained for $b=13$ and then we use the

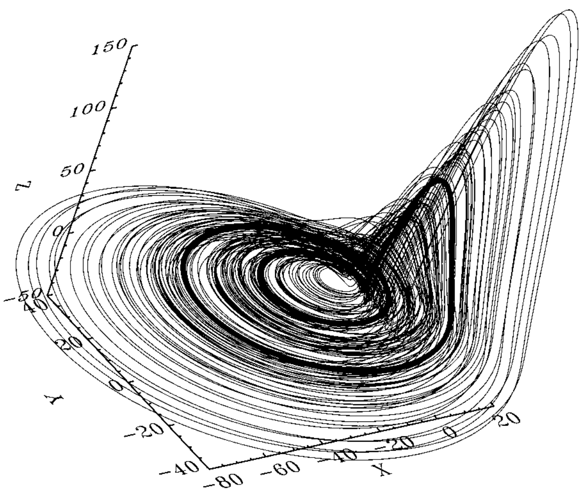

FIG. 1. A projection of the hyperchaotic attractor with embedded period-two orbit into (x,y,z) space for the Rössler system with $b=3$ and $p=0$.

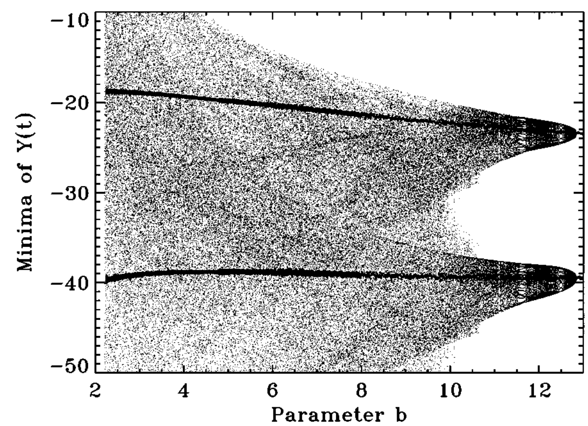

FIG. 2. Controlling and tracking the period-two orbit from $b=13$ to $b=2.2$. The minima of $y(t)$ are plotted versus b. Results of two runs, one with control and learning on, and one with control off, are superimposed.

adaptive algorithm to maintain the stability of the orbit while slowly taking the system to the region around $b=3$ where the system is hyperchaotic.

To simulate an experimental situation, we consider only a single scalar signal $y(t)$ available for measurement and use parameter p for control while system drift is simulated by varying parameter b. The time-delay coordinate vector $\boldsymbol{x}$ is constructed from $y(t)$ according to Eq. (2) where t_n indicates the time when $y(t)$ goes through a minimum. We found that an embedding dimension of two was optimal for the orbit shown as the solid line in Fig. 1. We were unable to control using an only one-dimensional embedding. Control was possible for $d>2$ but more difficult than for $d=2$. This can be explained by noting that the Lyapunov spectra contains two positive exponents, which shows that the unstable manifold is two-dimensional. We chose an orbit that completes two oscillations per cycle and we chose to take each minimum of $y(t)$ as data; placing our Poincaré section at the smaller of the two minima. The embedding vector in this particular case reduces to

$$\boldsymbol{x}_n=\begin{pmatrix} y(t_n{}^1) \\ y(t_n{}^2) \end{pmatrix}, \tag{24}$$

where $(t_n{}^1)$ and $(t_n{}^2)$ indicate the nth time $y(t)$ goes through the first and the second minima, respectively. Note that the delay time τ used in this case is determined by the time between successive minima in the control orbit and hence, τ will adapt automatically along with the orbit.

To be able to control a particular orbit of the system, the orbit must first be identified and then the control weights must be determined. In the case of a low-dimensional system, like the model of a thermal pulsed combustor [17], this was done by simply waiting for close returns in the Poincaré map to identify a period-one orbit and determine the control parameters. But the Rössler system very seldom comes close to the orbit which we choose to control. Therefore the strategy of waiting for close returns in the return map is impractical.

In a situation like this tracking becomes an important issue. We start the system for the parameter value where the appropriate low-period orbit is naturally stable. Since the system will be always in the neighborhood of this orbit, finding the control weights by estimating the linearized dynamics can be done in a few iterations. Our aim is to control the chosen orbit when the parameter setting is in the chaotic regime. This is done by slowly changing a parameter to reach chaotic region while adapting the control parameters to this change without losing control. Our adaptive learning strategy simultaneously applies control signal and uses the system response to reestimate the system dynamics and determine the control weights.

We found that the period-two orbit is stable when Rössler's parameter $b=13$ (see Fig. 2). Thus, we start with $b=13$ and apply small random control parameter changes δp_n to cause the system to sample the region of state space near the desired orbit (which is now stable) and use the adaptive control algorithm to learn about the system dynamics. Then we implement control and slowly decrease b well beyond the value where the period two orbit becomes unstable. The two-dimensional recursive control algorithm is able to stabilize the orbit over a wide range of b as shown in Fig. 2. Continuous learning through updating of the learning matrix ensures the adaptability of the control to the drift of the system parameter b.

In Fig. 3 at $b=8$ we discontinue learning and hold the control parameter of Eq. (11), $\boldsymbol{k}_1$, $\boldsymbol{x}_{fp}$, and R_1 fixed. At $i=100$ we release control and the system goes chaotic. Then, at iteration $i=200$, we switch control on again. At this value of b, the system comes sufficiently close to the fixed point that control can be reestablished after about 150 iterations. However, as b becomes smaller the orbit becomes more and more unstable and the region of the attractor containing the orbit is visited less often.

To confirm the estimation of the system dynamics by our control algorithm, we independently perform a stability analysis of the particular period-two orbit. Using real phase

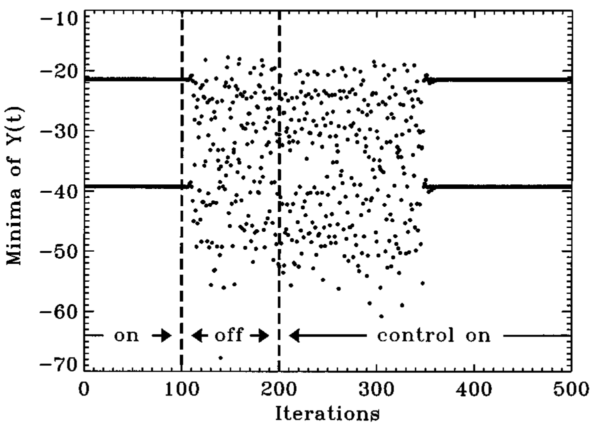

FIG. 3. The controlled Rössler system for $b=8$. After 100 cycles, control is turned off. The system falls off the period-two orbit into a chaotic state. At 200 iterations control is turned back on. The control recaptures the period-two orbit the first time the system happens to fall inside the control window, given by the maximum allowed control shift δp_m. Throughout the entire process learning was turned off and the control weights were held constant.

space coordinates (x,z,w), we take the three-dimensional Poincaré section at the minima of $y(t)$. From this three-dimensional Poincaré map, we numerically calculate the Jacobian around the period-two fixed point by taking differences for neighboring points. We keep track of the fixed point using Newton's method, while drifting the system through the parameter space well below the point where the orbit becomes unstable.

The eigenvalues of the 3×3 Jacobian matrix indicate the stability of the orbit. One of the eigenvalues was found to be almost zero. The other two are complex conjugates of each other for b between 13 and 4.05, and become real for $b<4.05$.

By comparing the eigenvalues of the matrix $\hat{\mathbf{M}}$ with the eigenvalues found for the Poincaré map in the true phase space, we can judge the quality of the system dynamics determined by the learning-control algorithm. Such a comparison is shown in Fig. 4. For integer values of the parameter b, we stop drifting, but continue learning $\hat{\mathbf{M}}$ with control and noise on. The triangles in Fig. 4 depict the modulus of the eigenvalues of $\hat{\mathbf{M}}$, averaged over around 100 learning cycles. Agreement is very good with the continuous curve that shows the modulus of the eigenvalues (the two which have modulus greater than one) of the Jacobian of the Poincaré map in the true phase space calculated while following the fixed point using Newton's method as explained above. Note that the largest Lyapunov multiplier, which is the largest eigenvalue magnitude, approaches 10 as the parameter b approaches 1.

The quality of the determination of system dynamics by the learning algorithm depends on the drift rate. The steady drift of a system parameter induces a small steady translation of the state space variables of the system indicating the slow change in the position of the fixed point. This causes the matrix $\hat{\mathbf{M}}$ that is obtained while drifting to have the tendency for one of its eigenvalues to move toward one and an associated distortion of the other eigenvalue. For higher drift rates, this effect becomes more pronounced and if the drift rate is too high the closed loop system is unable to maintain control and learn at the same time.

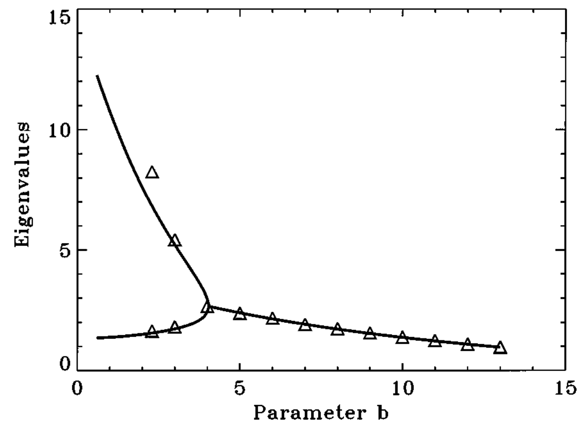

FIG. 4. Stability analysis for the period-two orbit. Taking a Poincaré section when $y(t)$ goes through a minimum, we numerically calculate the eigenvalues of the linearized map around the associated period-two fixed point using two methods. The solid line is the magnitude of the eigenvalues greater than one of the Jacobian of the Poincaré map in the true phase space coordinates x,z,w. Newton's method is used to follow the fixed point. One eigenvalue is near zero throughout. The other two are complex conjugates for b between 13 and 4.05, and become real for $b<4.05$. The triangles are the eigenvalues of the matrix $\hat{\mathbf{M}}$ obtained from the learning-control algorithm using a two-dimensional delay coordinate embedding as described in the text.

Figure 5 shows the effect of drift on the eigenvalues of $\hat{\mathbf{M}}$ found by the learning algorithm. We first start the controlled system at $b=9$ and then we initiate the drift. For drift step sizes $\delta b=10^{-3}$ and 10^{-4} per cycle, we plot the eigenvalues of $\hat{\mathbf{M}}$ obtained at each step by the learning-control

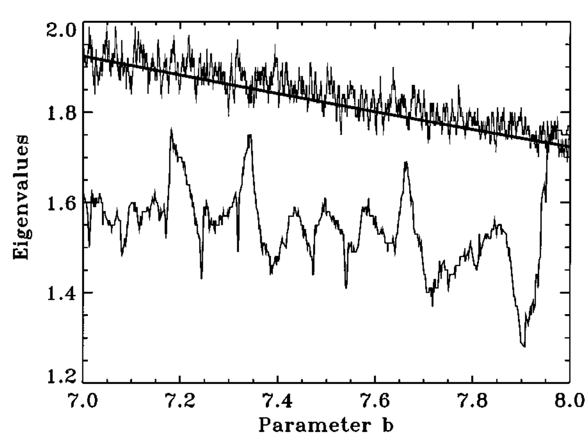

FIG. 5. The effect of drift on the eigenvalues of $\hat{\mathbf{M}}$ found by the learning-control algorithm. For a drift rate of $\delta b=10^{-4}$ per period the eigenvalues (upper plot) stay close to the curve given by the true state space stability analysis (thick line). A drift rate ten times larger causes a sudden decrease in the eigenvalues of $\hat{\mathbf{M}}$ (lower plot).

algorithm. While for the small drift rate the eigenvalues stay close to the curve given by the stability analysis in the true state space, increasing the drift rate by a factor of ten gives rise to a sudden decrease of the modulus of the eigenvalues.

V. CONCLUSIONS

An automated adaptive control algorithm based on a reconstruction of the relevant linearized dynamics from time-delay coordinates was developed for high-dimensional chaotic systems using a single control parameter. The algorithm was applied in a simulation using the Rössler hyperchaotic system with relatively large unstable Lyapunov multipliers. Although the Poincaré map for this system is three-dimensional, we found a two-dimensional embedding to reconstruct the state space was sufficient to control the system. The magnitude of the eigenvalues of the linearized map in the two-dimensional embedding agrees very well with the magnitudes of the unstable eigenvalues of the Jacobian matrix of the Poincaré map in the true state space. This suggests that in a case where the stable manifold is rather stable, i.e., the stable Lyapunov multipliers are all near 0, it is only necessary to use an embedding dimension equal to the dimension of the unstable manifold and use a control method that directs the system to the fixed point rather than the stable manifold. We found that it was necessary to apply small random perturbations to the system in order to both learn the dynamics and control the system at the same time.

Our simulation used the adaptive learning algorithm to stabilize an orbit with two unstable directions. The method is designed to be applicable when more than two unstable directions are present. The difficulties encountered in practical applications will no doubt increase with the number of unstable directions and it remains for future research to determine the limits and robustness of the algorithm under such conditions. In any case, the simultaneous learning and control capability is likely to be important in any future application of this type of active feedback control to real systems with multiple unstable directions.

ACKNOWLEDGMENTS

This work was funded by the Electric Power Research Institute (EPRI) of Palo Alto, California, under Contract No. WO9000-35. The support and encouragement of EPRI's Dr. John Stringer are gratefully acknowledged. There were many useful discussions with many colleagues but most notably Markus Löcher. We also thank Kenneth Showalter, Valery Petrov, and Eugene Mihaliuk for invigorating discussions and a preprint describing their work on stabilizing unstable states in high-dimensional systems.

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

[2] E. Ott and M. Spano, Phys. Today **48**, 34 (1995).

[3] T. Shinbrot, Adv. Phys. **44**, 73 (1995); T. Shinbrot, C. Grebogi, E. Ott, and J. A. Yorke, Nature **363**, 411 (1993); G. Chen and X. Dong, Int. J. Bifurc. Chaos **3**, 1363 (1993).

[4] A. J. Markworth, J. Stringer, and R. W. Rollins, Mater. Res. Soc. Bull. **20**, 20 (1995); E. Hunt and G. Johnson, IEEE Spectrum **30**, 32 (1993); W. L. Ditto and L. Pecora, Sci. Am. **269**, 78 (1993).

[5] P. So and E. Ott, Phys. Rev. E **51**, 2955 (1995).

[6] V. Petrov, E. Mihaliuk, S. K. Scott, and K. Showalter, Phys. Rev. E **51**, 3988 (1995).

[7] M. Ding, W. Yang, V. In, W. L. Ditto, M. L. Spano, and B. Gluckman, Phys. Rev. E **53**, 4334 (1996).

[8] T. Shinbrot, E. Ott, C. Grebogi, and J. A. Yorke, Phys. Rev. Lett. **65**, 3215 (1990); Phys. Rev. A **45**, 4165 (1992); Phys. Rev. Lett. **68**, 2863 (1992); Phys. Lett. A **169**, 349 (1992).

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

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

[11] E. Ott, C. Grebogi, and J. A. Yorke, in *Chaos/XOAC Soviet-American Perspectives on Nonlinear Science*, edited by D. K. Campbell (AIP, New York, 1990), pp. 153–172.

[12] U. Dressler and G. Nitsche, Phys. Rev. Lett. **68**, 1 (1992).

[13] G. Nitsche and U. Dressler, Physica D **68**, 153 (1992).

[14] D. Auerbach, C. Grebogi, E. Ott, and J. Yorke, Phys. Rev. Lett. **69**, 3479 (1992).

[15] R. W. Rollins, P. Parmananda, and P. Sherard, Phys. Rev. E **47**, R780 (1993); P. Parmananda, R. W. Rollins, P. Sherard, and H. Dewald, in *Proceedings of the 2nd EXPERIMENTAL CHAOS Conference*, edited by W. Ditto, S. Vohra, M. Shlesinger, and M. Spano (World Scientific Press, River Edge, NJ, 1994).

[16] M. A. Rhode, R. W. Rollins, and C. A. Vassiliadis, in *Proceedings of the 26th Southeastern Symposium on System Theory* (The Institute of Electrical Engineers, Los Alamitos, CA, 1994), p. 638.

[17] M. A. Rhode, R. W. Rollins, A. J. Markworth, K. D. Edwards, K. Nguyen, C. S. Daw, and J. F. Thomas, J. Appl. Phys. **78**, 2224 (1995).

[18] O. E. Rössler, Phys. Lett. **71A**, 155 (1979).

[19] A. Wolf, J. B. Swift, H. L. Swinney, and J. A. Vastano, Physica D **16**, 285 (1985).

[20] A modified version of *Chaotic Dynamics Workbench* by R. W. Rollins (Physics Academic Software-AIP, New York, 1990) was used.