Predictive Poincaré control: A control theory for chaotic systems

Jörg Schweizer*
Chaire des Circuits et Systèmes–CIRC, Ecole Polytechnique Fédérale de Lausanne, 1015 Lausanne, Switzerland

Michael Peter Kennedy
Department of Electronic and Electrical Engineering, University College Dublin, Dublin 4, Ireland
(Received 15 December 1994)

One of the most interesting features of chaotic systems is the large number of unstable orbits embedded in a chaotic attractor. In this work, we propose a *global* chaos-control technique called predictive Poincaré control (PPC) that permits stabilization of a predefined solution, using only *small* control pulses. We prove this result for a large class of n-dimensional chaotic systems. The predefined solution can be a periodic or nonperiodic oscillation, expressed by a periodic or nonperiodic *symbolic sequence* [S. Hayes, C. Grebogi, and E. Ott, Phys. Rev. Lett. **70**, 3031 (1993)]. We apply the general PPC scheme to the well known Lorenz model and study its robustness with respect to parasitic effects.

PACS number(s): 05.45.+b

I. INTRODUCTION

While chaotic phenomena have been observed and analyzed in various scientific disciplines for almost 30 years, physicists and engineers have only recently begun to synthesize systems that exploit the nature of chaos. For example, the possibility of synchronizing two identical chaotic systems has prompted developments in spread spectrum and secure communications [2–6]. *Chaos control* represents another application domain exploiting the large number of steady-state solutions of chaotic systems (a chaotic attractor is close to an infinite number of unstable orbits) and high sensitivity to initial conditions. Chaos control is concerned with stabilizing certain predefined orbits. It turns out that only *small* control signals or parameter perturbations are required in order to force a chaotic system to exhibit a prescribed solution.

Ott, Grebogi, and Yorke state [7] that the advantage of having chaotic motion in a system is that one can change the overall system behavior with small parameter variations. Thus a "multiuse" system is available that might be accommodated without changing the overall system configuration.

In the past three years, unstable orbits have been stabilized in chemical chaos [8], electronic circuits [9], laser physics [10], and higher-dimensional mechanical systems [11]. For an overview on chaos control in engineering, see [12].

In recent years, three main control techniques have been proposed for chaotic systems.

(i) The method introduced by Ott, Grebogi, and Yorke [7] is designed to stabilize the unstable periodic orbit Γ of the autonomous chaotic system $\dot{\mathbf{x}}=\mathbf{f}(\mathbf{x},p)$, where p is a control parameter. Based on the linearization $\partial\mathbf{x}^*(p)/\partial p|_{p=0}$ in the vicinity of $\mathbf{x}^*$, which is a fixed point of Γ, a closed orbit can be stabilized by varying p. The Ott-Grebogi-Yorke method is therefore a *local* control scheme that operates correctly only if the trajectory $\mathbf{x}(t,\mathbf{x}_0)$ passes close to $\mathbf{x}^*$ [13].

(ii) The more traditional global control techniques developed by Singer, Wang, and Bau [14] and Chen and Dong [15] use *error feedback* to stabilize a point $\bar{\mathbf{x}}$ [14] or a predefined, explicitly known trajectory $\bar{\mathbf{x}}(t)$ [15]. In this case, the feedback signal is proportional to $\mathbf{x}(t)-\bar{\mathbf{x}}(t)$, where $\mathbf{x}$ is the state of the system to be controlled. Another interesting approach was suggested by Pyrgas [16], where the feedback is proportional to $\mathbf{x}(t)-\mathbf{x}(t-T)$. The feedback control approach has been proven to work for some systems where a Lyapunov function can be found (see, for example, [15]).

(iii) More recently, a qualitatively different approach to controlling chaos has emerged. In the method described by Hayes, Grebogi, and Ott, a system may be forced to follow a prescribed *symbolic sequence* by applying small control pulses each time the trajectory passes through a Poincaré section [1]. Using this scheme, periodic and nonperiodic symbolic control sequences result in periodic and nonperiodic system behavior, respectively. This technique has been applied to simulation to Chua's circuit in order to modulate information onto a chaotic carrier signal [1]. In this work, we develop the idea of using symbolic sequences for chaos control; we describe in detail a general *global* control theory for n-dimensional chaotic systems.

The different features of our scheme are (i) *global* control, in contrast with the local scheme described by Ott, Grebogi, and Yorke [7], (ii) reduced computation complexity of the controller compared with [1], and (iii) guaranteed *small* control pulses.

In the following section, we give an overview of predictive Poincaré control (PPC) by explaining how it can be used to control the output of a simple discrete-time sys-

*Electronic address: joerg@circhp.epfl.ch

tem: the Bernoulli shift. In Secs. III and IV we develop the mathematical foundations of PPC for continuous-time dynamical systems. We describe in detail the implementation of PPC in the Lorenz system in Sec. V.

II. MOTIVATIONAL EXAMPLE

Consider the first-order discrete-time dynamical system described by

$$z(k+1)=2z(k) \bmod 1 ,$$
$$s(k)=\begin{cases}1 \text{ if } z(k)\geq 0.5\\ 0 \text{ if } z(k)<0.5 ,\end{cases} \tag{1}$$

where $z\in[0,1)$ is called the *state* and s is the *output* symbol. The next state is obtained from the present state $z(k)$ by multiplying $z(k)$ by 2 and keeping just the fractional part, e.g., 0.654 321 is mapped on to 0.308 642.

Note that the state space has effectively been partitioned into two distinct regions $H_0=[0,0.5)$ and $H_1=[0.5,1)$, in each of which the function $2z(k)$ mod1 increases monotonically. The symbol $s(k)$ indicates whether $z(k)$ is in H_0 or H_1.

The operation of the dynamics is perhaps clearer if we represent the state $z(k)$ as a binary expansion

$$z(k)=0.b_k b_{k+1} b_{k+2} b_{k+3}\ldots=\sum_{j=k}^{\infty} 2^{-(j-k+1)} b_j ,$$

where each digit b_j is either 0 or 1. The next state $z(k+1)$ is derived from the present state by *shifting* the digits to the left and setting the leftmost digit to zero: hence the name Bernoulli shift. Thus

$$z(k+1)=0.b_{k+1} b_{k+2} b_{k+3} b_{k+4}\ldots ,$$

with output $s(k+1)=b_{k+1}$. Notice that (1) produces a sequence of output symbols $s(k)=b_0,b_1,b_2,b_3,\ldots$ when started from initial state $z(0)=0.b_0 b_1 b_2 b_3\ldots$, where this sequence of binary digits b_k is simply the binary expansion of $z(0)$.

To each initial condition corresponds a unique binary sequence. In order to generate a prescribed sequence (for example, if one wished to transmit binary information), one could in principle set the initial state $z(0)$ of (1) to the binary expansion of the message and transmit the output sequence $\{s(k)|k=0,1,\ldots\}$. The method is *predictive* in the sense that by setting the initial state *now*, we are predicting future outputs.

There are two major drawbacks to this idea. First, to generate a sequence of length N, the initial condition must be specified with precision of order 2^{-N}. Second, in a practical implementation of the system, random noise that perturbs the state by an amount greater than 2^{-N} will show up as an error in the output sequence because the "insignificant" rightmost bits of the state eventually shift into the most significant leftmost position.

Both of these limitations can be overcome by using a prediction length shorter than that of the message. For example, if the noise level is less than $2^{-(m+1)}$, a sequence of length N may be generated by setting only the next m bits at a time. At each iterate, a new bit is shifted into the $(m+1)$st (rightmost) position, which may or may not correspond to the output that we wish to produce m steps forward in time. If it does not, the bit is corrected by adding a *control* signal $u(k)$ to the current state $z(k)$, such that $z(k+1)=2[z(k)+u(k)]$ mod1. Note that $u(k)$ is no more than $2^{-(m+1)}$ in magnitude. Thus an arbitrary output sequence may be generated by using a prediction length m and a control signal of magnitude $2^{-(m+1)}$.

Figure 1(a) shows the first 12 digits of the binary expansion of $\pi/4$. The dark squares denote the desired output symbols $s(k)$ as system (1) evolves. To produce this sequence of 12 bits, one must set the initial condition to any value in the interval [0.785 278 320 312, 0.785 400 390 624) and let the system run for 12 steps. Equivalently, the initial condition must be specified to 12 bits.

Alternatively, by using a prediction length of 4, the system can be started from any value of $z(0)$ in the interval [0.781 25,0.8125) if we are willing to apply a correction pulse of magnitude less than 0.031 25 at each step to modify the rightmost bit; this is illustrated in Fig. 1(b).

While we have described predictive control for a simple discrete-time system, it may also be applied to a continuous-time dynamical system of the form $\dot{\mathbf{x}}=\mathbf{f}(\mathbf{x})$, $\mathbf{x}\in\mathbb{R}^n$ by using a Poincaré map to define the equivalent discrete-time dynamics.

The essential components of a continuous-time predic-

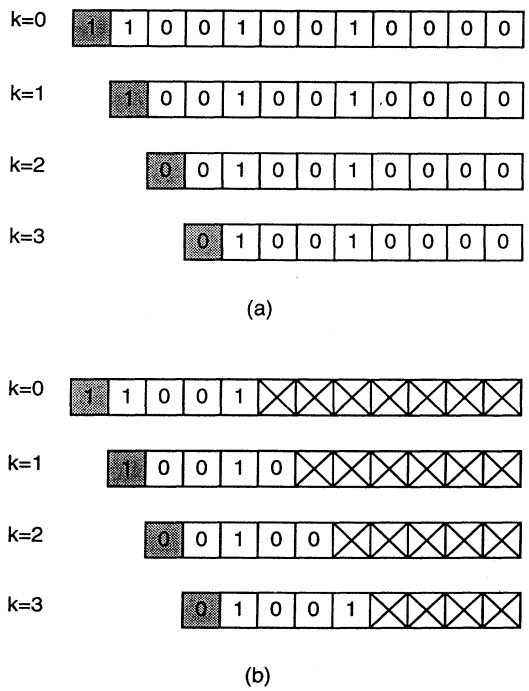


FIG. 1. Evolution of (1). The dark squares denote the output symbols $s(k)$. (a) To produce the sequence of N bits, the initial condition must be set with N-bit resolution. (b) With a prediction length $m=4$, the initial state can be set with five-bit resolution if a control signal of magnitude 2^{-5} is applied to correct the rightmost bit at each step.

tion Poincaré controller are a sensor to measure the state $\mathbf{x}(t_k)$ of the system as it crosses the Poincaré section at time t_k, a prediction register to keep track of the next m desired outputs, a map from the prediction register to the state $\mathbf{x}_w(t_k)$, which is known to generate the desired subsequence $\{s(k+i),\ i=0,1,2,\ldots,m\}$, and a transducer to apply the required correction $\mathbf{u}(t_k)=\mathbf{x}_w(t_k)-\mathbf{x}(t_k)$ to the state in order to produce the prescribed sequence. In Secs. III and IV we lay the mathematical foundations of predictive Poincaŕe control for continuous-time dynamical systems, showing how to define the Poincaré map, the symbolic dynamics, and the map from the prediction register to the state.

III. THEORY OF PREDICTIVE POINCARÉ CONTROL

The predictive Poincaré control technique can be used to force a general n-dimensional chaotic continuous-time dynamical system described by the state equations $\dot{\mathbf{x}}=\mathbf{f}(\mathbf{x})$, $\mathbf{x}\in\mathbb{R}^n$ to follow a predefined trajectory, expressed by the symbolic sequence $\mathbf{w}=\{w(i),\ i=0,1,\ldots,i\in\mathbb{N}\}$, where symbol $w(i)\in\mathbb{N}^+$. Small control pulses are applied to the system by means of a control vector $\mathbf{u}\in\mathbb{R}^n$

$$\dot{\mathbf{x}}=\mathbf{f}(\mathbf{x})+\mathbf{u}(t)\ . \tag{2}$$

Before discussing a detailed implementation of PPC in the Lorenz system, we define the Poincaré map $\mathbf{g}$ from a Poincaré section to itself and the maps $\mathbf{v}$ and $\mathbf{v}^{-1}$, which translate between the continuous-time dynamics and the discrete-time dynamics of the Poincaré map. We define a partition of the state space in the Poincaré map by identifying the monotone branches $\mathbf{g}_i$ of $\mathbf{g}$ and define the symbolic dynamics in terms of iterates of $\mathbf{g}$ and this partition. For a more detailed introduction to nonlinear and symbolic dynamics, see [17–20].

A. Poincaré map

We use a *Poincaré map* to transform the dynamics of the n-dimensional autonomous continuous-time dynamical system $\dot{\mathbf{x}}=\mathbf{f}(\mathbf{x})$, $\mathbf{x}\in\mathbb{R}^n$, to the $(n-1)$-dimensional discrete-time system $\mathbf{z}(k+1)=\mathbf{g}(\mathbf{z}(k))$, $k\in\mathbb{N}$, $\mathbf{z}\in\mathbb{R}^{(n-1)}$.

The Poincaré map $\mathbf{g}$: $\mathbb{R}^{n-1}\mapsto\mathbb{R}^{n-1}$ is generated in the following way. First, the Poincaré section Σ must be defined as an $(n-1)$-dimensional hyperplane of the n-dimensional state space of system $\dot{\mathbf{x}}=\mathbf{f}(\mathbf{x})$. We note $\mathbf{x}(t_k)$ at each time $t=t_k$ the trajectory $\mathbf{x}(t,\mathbf{x}_0)$ passes through Poincaré section Σ. The coordinate system $\{z_i,\ i=1,\ldots,(n-1)\}$, $z_i\in\mathbb{R}$ parametrizes Σ and $\mathbf{v}$: $\mathbb{R}^{n-1}\mapsto\mathbb{R}^n$ maps each $\mathbf{z}\in\mathbb{R}^{(n-1)}$ to a $\mathbf{x}^*\in\Sigma\subset\mathbb{R}^n$. There also exists a unique inverse of map $\mathbf{v}$ that transforms each point $\mathbf{x}^*$ on Σ to the $(n-1)$-dimensional coordinate system $\mathbf{z}$ such that $\mathbf{v}^{-1}$: $\mathbb{R}^n\mapsto\mathbb{R}^{n-1}$, $\mathbf{x}^*\mapsto\mathbf{z}^*$. Since $\mathbf{x}(t_k)\in\Sigma\forall k$, we can define $\mathbf{z}(k)=\mathbf{v}^{-1}(\mathbf{x}(t_k))$.

The maps $\mathbf{v}^{-1}$ and $\mathbf{v}$ allow us to move from the original continuous-time dynamics to a discrete-time Poincaré map $\mathbf{z}(k+1)=\mathbf{g}(\mathbf{z}(k))$ and back. Assuming that the Poincaré map $\mathbf{g}$ has been determined, we can predict $\mathbf{x}(t_{k+i})$ by transforming $\mathbf{x}(t_k)$ into z space, evaluating the ith iterate of map $\mathbf{g}$, and transforming back to the original coordinate system. Thus

$$\mathbf{z}(i)=\mathbf{v}^{-1}(\mathbf{x}(t_k))\ , \tag{3}$$

$$\mathbf{z}(k+i)=\mathbf{g}^i(\mathbf{z}(k))\ , \tag{4}$$

$$\mathbf{x}(t_{k+i})=\mathbf{v}(\mathbf{z}(k+i))\ . \tag{5}$$

B. Symbolic dynamics

Symbolic dynamics can be described as a "coarse-grained description" of an evolutionary process [18]. In the Bernoulli shift example, $s(i)$ indicates whether $z(k)$ is less than or greater than 0.5, i.e., $z(k)\in H_0$ or $z(k)\in H_1$. More generally, the invariant set Λ ($\Lambda=\{\mathbf{z}\in\mathbb{R}^{(n-1)}|\mathbf{z}\in\Lambda,\mathbf{g}(\mathbf{z})\in\Lambda\}$) of a Poincaré map $\mathbf{g}$ is partitioned into $(\rho+1)$ so-called *"hypersectors"* $H_0,H_1,H_2,\ldots,H_\rho$ and symbol $s(k)=j$ if $\mathbf{z}(i)\in H_j$. Knowing $\mathbf{x}(t_k)$ and the maps $\mathbf{v}^{-1}$, $\mathbf{v}$, and $\mathbf{g}$, we can use (3)–(5) to predict the future crossings of the Poincaré section $\mathbf{x}(t_{k+i})$ and thus the sequence of hypersectors visited by a trajectory of the system. We partition Λ into hypersectors so that the map $\mathbf{g}$ restricted to H_j (denoted $\mathbf{g}_j$) is a one-to-one mapping from each H_j to its image H_j^*.

Definition 1. Let Λ be the invariant set of the discrete system $\mathbf{z}(k+1)=\mathbf{g}(\mathbf{z}(k))$; then H_j, $j=0,1,\ldots,\rho$, are defined such that each H_j is compact and connected; H_j provides

$$\bigcup_{j=0}^{j=\rho}H_j=\Lambda\ ,\quad H_j\cap H_{j'}=\emptyset\ \ \forall\ j\neq j'\ ,$$

where $j,j'\in\{0,1,\ldots,\rho\}$; and for all $j=0,1,\ldots,\rho$

$$\mathbf{g}_j\colon H_j\mapsto H_j^*\subseteq\Lambda\ ,$$

where $\mathbf{g}_j$ (the restriction of $\mathbf{g}$ to H_j) is a one-to-one mapping. Following [18], we call $\mathbf{g}_j$ the *monotone branches* of map $\mathbf{g}$. The hypersectors H_j can be further divided into subsectors $H_{j,i}$ as follows.

Definition 2. The subsectors $H_{j,i}$, $i=0,1,\ldots,\rho$, are defined such that each $H_{j,i}$ is compact and connected, $H_{j,i}$ provides

$$\bigcup_{i=0}^{i=\rho}H_{j,i}=H_j\ ,$$

$$H_{j,i}\cap H_{j,i'}=\emptyset\ \forall\ i\neq i',\ i,i'\in\{0,1,\ldots,\rho\}\ ,$$

and for each j and i the maps

$$\mathbf{g}_{j,i}\colon H_{j,i}\mapsto H_{j,i}^*\subseteq H_i\ ,$$

$$\mathbf{g}_{j,i}^{-1}\colon H_{j,i}^*\mapsto H_{j,i}$$

are continuous and in a one-to-one correspondence.

Let the sequence $\{\mathbf{z}(k+i),\ i=0,1,\ldots,m\}$ be the evolution of the discrete system $\mathbf{z}(k+1)=\mathbf{g}(\mathbf{z}(k))$, $\mathbf{z}\in\Lambda\subset\mathbb{R}^{(n-1)}$; then $\mathbf{s}_m(k)=\{s(k+i),\ i=0,\ldots,m\}=\{s_i,\ i=0,\ldots,m\}$ is the symbolic sequence of length $(m+1)$ starting at $\mathbf{z}(i)$ with $s_i=j$ if $\mathbf{z}(k+i)\in H_j$. In other words, the system $\dot{\mathbf{x}}=\mathbf{f}(\mathbf{x})$ produces the symbol $s_i=j$ at time t_{k+i} if $\mathbf{v}^{-1}(\mathbf{x}(t_{k+i}))\in H_j\subset\Lambda$.

In terms of symbolic sequences, we first partition Λ into hypersectors so that the map $\mathbf{g}$ restricted to H_{s_0} (denoted $\mathbf{g}_{s_0}$) is in a one-to-one correspondence with each H_{s_0} to its image $H^*_{s_0}$. We further refine the partition H_{s_0} into the subsectors $H_{\mathbf{s}_m}=H_{s_0,\ldots,s_m}$ using the more general notation $\mathbf{g}_{\mathbf{s}_m}$ to indicate the part of $\mathbf{g}$ that maps $H_{\mathbf{s}_m}$ to $H^*_{\mathbf{s}_m}$.

In this way, we generate a set of subsectors $H_{\mathbf{s}_m}\subset\Lambda$ such that the system $\mathbf{z}(k+1)=\mathbf{g}(\mathbf{z}(k))$ produces the symbolic sequence $\mathbf{s}_m(k)$ if $\mathbf{z}(k)\in H_{\mathbf{s}_m}$. Recalling our example of the Bernoulli shift map, the output sequence will be $\mathbf{s}_4=\{1,1,0,0,1\}$ if the state begins from subsector $H_{\mathbf{s}_4}=[0.781\,25, 0.8125)$.

The continuous-time system $\dot{\mathbf{x}}=\mathbf{f}(\mathbf{x})$ can be forced to follow any *desired symbolic sequence* $\mathbf{w}_m(k)=\{w(k+i), i=0,1,\ldots,m\}$, $w(k+i)=j$ if $\mathbf{z}(k+i)\in H_j$, by applying a control pulse $\mathbf{u}$ at time t_k that drives the state $\mathbf{z}(k)$ into subsector $H_{\mathbf{w}_m}$. The control pulse is defined explicitly by

$$\mathbf{u}(t)=[\mathbf{x}_w(t_k)-\mathbf{x}(t_k)]\delta(t-t_k)$$

where

$$\mathbf{v}^{-1}(\mathbf{x}_w(t_k))\in H_{\mathbf{w}_m}\ . \tag{6}$$

In Sec. IV we propose a numerical method for determining the point $\mathbf{z}_w=\mathbf{v}^{-1}(\mathbf{x}_w(t_k))$ in $H_{\mathbf{w}_m}$. First, we will prove that the control required to predict m steps ahead is smaller than that required to predict $(m-1)$ steps ahead.

C. Relationship between magnitude of control signal and m

Assuming that $H_{\mathbf{s}_m}=H_{s_0,\ldots,s_m}$ is *compact, connected,* and a subset $H_{\mathbf{s}_{m-1}}$, then, after a transient, the control energy of $\mathbf{u}$ is small compared to that of the states $\mathbf{x}$; one can imagine that $\mathbf{z}(k)$ is in $H_{\mathbf{s}_{m-1}}(k)$ and that the symbol to be generated m steps ahead is s_m. We then have to steer the system from a point somewhere in $H_{\mathbf{s}_{m-1}}(k)$ to *inside* $H_{\mathbf{s}_m}(k)$. The correction to be made will be small if $H_{\mathbf{s}_{m-1}}(k)$ is small and $H_{\mathbf{s}_m}(k)$ lies inside $H_{\mathbf{s}_{m-1}}(k)$. This means that the "size" of subsector $H_{\mathbf{s}_m}$ decreases as the *prediction length* m increases. Subsector $H_{\mathbf{s}_m}$ should also be connected in order to avoid multiple control targets. If the control pulse at time t_{k-1} was applied successfully, then the state $\mathbf{z}(k-1)$ is inside $H_{\mathbf{s}_m}(k-1)$ [and hence inside $H_{\mathbf{s}_{m-1}}(k)$] at time t_k. The ideal transient behavior (for a control pulse of finite height) lasts from t_{k-1} to t_k. Note that $\mathbf{s}_m(k-1)=\{s(k-1+i), i=0,\ldots,m\}$ and $\mathbf{s}_{m-1}(k)=\{(k+i), i=0,\ldots,m-1\}$.

Theorem 1. If the hypersectors H_{s_0} and subsectors H_{s_0,s_1}, $s_0,s_1=0,\ldots,\rho$, are defined according to Definitions 1 and 2, then $H_{\mathbf{s}_m}$ is compact, connected, and a subset of $H_{\mathbf{s}_{m-1}}$.

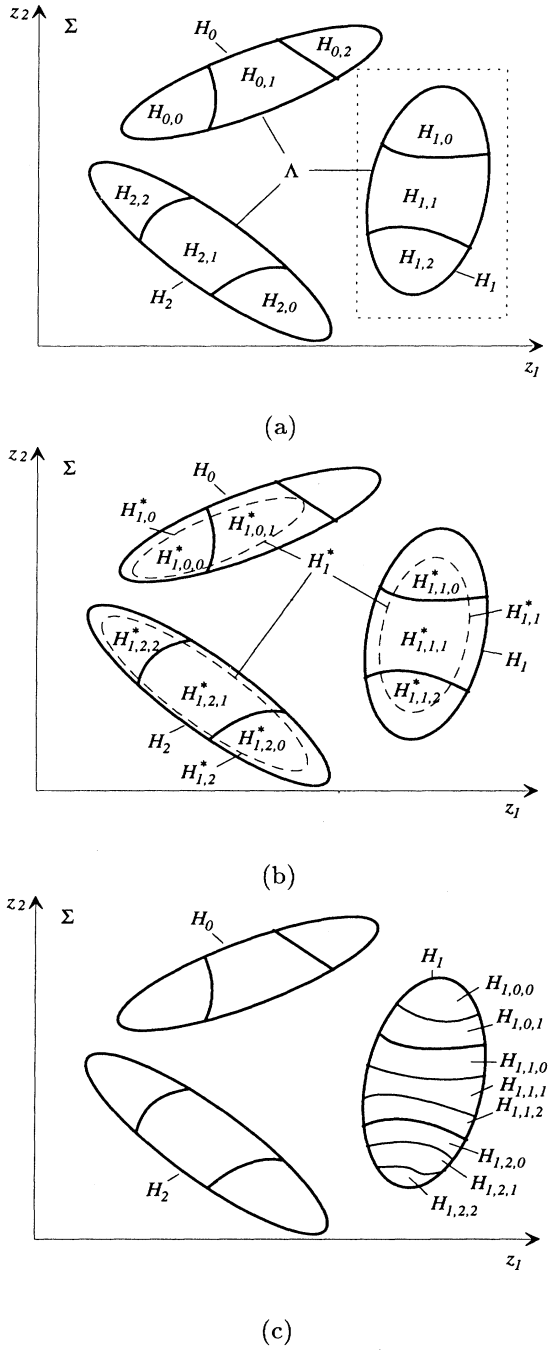


FIG. 2. Hypersectors H_{s_0}, $s_0=0,1,2$, are shown schematically for a two-dimensional map $\mathbf{g}$. The subsectors H_{s_0,s_1,s_2} are generated successively by forward mapping, splitting, and inverse mapping. (a) In the case shown, all subsectors H_{s_0,s_1}, $s_0,s_1=0,1,2$, exist; (b) shows the forward mapping for a fixed $s_0=1$. $\mathbf{g}_{1,s_1}$ maps the subsectors H_{1,s_1} to H^*_{1,s_1} for each s_1. Then each H^*_{1,s_1} is divided in H^*_{1,s_1,s_2}. Note that $H^*_{1,0,2}$ does not exist because $H^*_{1,0}\cap H_{0,1}=\emptyset$. Therefore all sequences $\{1,0,2,s_3,s_4,\ldots\}$ are inadmissible. (c) The new subsectors H^*_{1,s_1,s_2} are mapped backward into H_{1,s_1} by the inverse map $\mathbf{g}^{-1}_{1,s_1}$. The newly generated subsector H_{1,s_1,s_2} are all *inside* H_{1,s_1}.

For the proof of this theorem see the Appendix and Fig. 2.

Note that H_{s_m} can result in an empty set for some $\mathbf{s}_m$. If we assume that the empty set can also be divided into $(\rho+1)$ empty sets, we have no loss of generality. However, if a particular H_{s_m} is an empty set, the sequence $\mathbf{s}_m$ is *nonadmissible.* In practice, nonadmissible sequences will never be generated by a free-running chaotic system. The reason for this is that no initial value $\mathbf{z}(0)$ can be found such that the system $\mathbf{z}(k+1)=\mathbf{g}(\mathbf{z}(k))$ will produce a nonadmissible sequence.

In the special case that $H^*_{s_0}=\Lambda\ \forall\ s_0$, we are dealing with *fully developed chaos* and all sequences $\mathbf{s}_m$ are admissible [18]. Usually in real chaotic systems $H^*_{s_0}\subset\Lambda$ and we introduce the term *constrained fully developed chaos:* if all H_{s_m} are nonempty sets for $m\leq\mu$ for a finite μ, we call this constrained fully developed chaos of degree μ. The average surface of H_{s_m} will then shrink as $1/(\rho+1)^{(m+1)}$ since $m\leq\mu$. Recall the Bernoulli shift map example where $\rho=1$ and $m=4$; in this case, $\max[|u(k)|]<2^{-5}=0.031\,25$.

IV. IMPLEMENTATION OF PPC

In this section we describe how PPC theory can be used to realize a chaos control scheme for a general n-dimensional chaotic system. As outlined in Sec. II, the essential components of a predictive Poincaré controller are a sensor to measure the state of the system as it crosses the Poincaré section Σ, a prediction register to keep track of the current output and next m desired outputs $\mathbf{w}_m$, a control function that identifies a target state $\mathbf{x}_w$ with the desired sequence, and a transducer to apply the required correction to the state. In this section, we describe each of these functional units.

The application of PPC to the autonomous system $\dot{\mathbf{x}}=\mathbf{f}(\mathbf{x})$ as shown in Fig. 3 requires knowledge of all n states. [One might use a derivative or delay coordinates to generate the state(s) that cannot be measured.] In general we also need access to some states of the system in order to apply the control pulses via control vector $\mathbf{u}$.

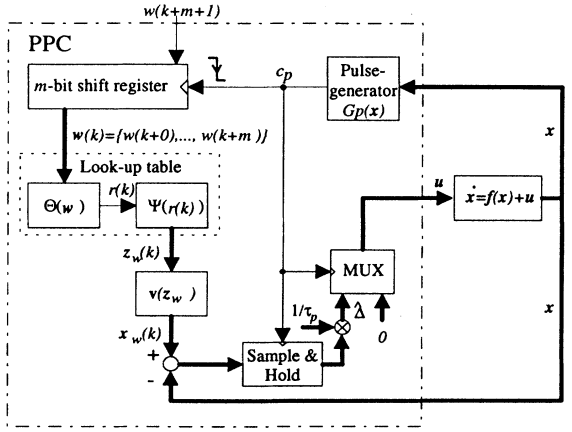


FIG. 3. Block diagram of PPC scheme.

The principle of the PPC setup in Fig. 3 is to find the $\mathbf{x}_w(t_k)$ in (6) to which the system must be controlled in order to produce the desired symbolic output sequence $\mathbf{w}_m(k)$. We must construct a control function Ψ that identifies an analog target state $\mathbf{z}_w$ in the Poincaré section with a discrete symbolic sequence $\mathbf{w}_m$.

In the following, we propose a constructive design algorithm that allows us to determine all unknown functions ($\mathbf{v}^{-1}$, $\mathbf{v}$, and $\mathbf{g}$) of a general PPC setup just by analyzing the sampled data $\{\mathbf{x}(t_k),\ k=0,1,\ldots,\zeta\}$.

A. Sensor

In the sensor, the sampled data $\{\mathbf{x}(t_k),\ k=0,1,\ldots,\zeta\}$ must be transformed into the coordinate system of the Poincaré section Σ by the function $\mathbf{v}^{-1}$: $\mathbf{z}(k)=\mathbf{v}^{-1}(\mathbf{x}(t_k))$. For experimental reasons, it is often useful to identify $\mathbf{z}$ directly with $(n-1)$ states of vector $\mathbf{x}$. Thus

$$\begin{bmatrix} z_1(k) \\ \vdots \\ z_{j-1}(k) \\ z_j(k) \\ \vdots \\ z_{n-1}(k) \end{bmatrix} = \begin{bmatrix} x_1(t_k) \\ \vdots \\ x_{j-1}(t_k) \\ x_{j+1}(t_k) \\ \vdots \\ x_n(t_k) \end{bmatrix} = \mathbf{v}^{-1}(\mathbf{x})\ . \tag{7}$$

Sampling takes place at each time t_k when $x_j(t)$ crosses a specified threshold c. For simplicity, we choose c to be the jth coordinate of an equilibrium point $\bar{\mathbf{x}}$ of system $\dot{\mathbf{x}}=\mathbf{f}(\mathbf{x})$. If enough data are sampled, the Poincaré map $\mathbf{g}(\mathbf{z}(k))=\mathbf{z}(k+1)$ can be determined approximately, and if necessary interpolated, to make $\mathbf{g}$ sufficiently smooth [21].

B. Partition

In order to find the hypersectors H_{s_0} we first determine the sectors $\tilde{H}_i$ in which $\mathbf{g}$ is connected, continuous, *and* a one-to-one-mapping. [One can compute $\tilde{H}_i$ by determining the regions where the sign of $\partial g_j(\mathbf{z})/\partial z_j$ is constant for $i=1\cdots(n-1)$ and for all $\mathbf{z}\in\tilde{H}_i$.] Each sector $\tilde{H}_i$ could serve directly as one of the hypersectors H_{s_0}. Alternatively, some monotone branches, say, $\tilde{H}_i$ and $\tilde{H}_{i'}$, could be combined to form one hypersector $H=H_i\cup\tilde{H}_{i'}$ if $\mathbf{g}:H\mapsto H^*$ is a one-to-one mapping. The motivation for connecting these monotone branches is that there will be fewer empty sets in the subsectors and we will therefore expect a higher number of admissible symbolic sequences. The conditions of Definition 1 are not violated by connecting $\tilde{H}_i$ and $\tilde{H}_{i'}$. With the procedure described above we obtain $(\rho+1)$ distinct hypersectors H_{s_0}, $s_0=0,\ldots,\rho$.

C. Control function

In the Bernoulli shift example, we saw that an initial state lying in the interval $z(0)\in[0.781\,25,0.8125)$ produced the output sequence $\mathbf{s}=\{1,1,0,0,1\}$. The control

function associates an initial condition with each desired sequence. Here we can associate an integer r with each sequence by considering it as a binary number (e.g., $1\times2^4+1\times2^3+1\times2^0=25$) and define a function Ψ: $\{r=0,1,\ldots,31\}\rightarrow[0,1)$ such that $\Psi(25)\in[0.781\,25, 0.8125)$.

In general, we write a symbolic sequence $\mathbf{s}_m$ as an integer as

$$r=\Theta(\mathbf{s}_m)=\sum_{i=0}^{m} s_i(\rho+1)^{m-i} . \tag{8}$$

For convenience, subsectors $H_{\mathbf{s}_m}$ can be denoted equivalently by H_r. To find a $\mathbf{z}_w$ for each r, we first determine H_r. To do so we evaluate the symbolic sequence $\mathbf{s}_m$ and the corresponding value r for each measured or interpolated point $\mathbf{z}$ using the relation $s_i=j$ if $\mathbf{g}^i(\mathbf{z})\in H_j$, where $\mathbf{g}^i$ is the ith iterate of map $\mathbf{g}$ and H_j are the hypersectors found in Sec. IV B. Subsectors H_r are those regions of Σ where r is constant. It is again a numerical problem to find a $\mathbf{z}_w$ inside each H_r. In practice one finds a $\mathbf{z}_w$ that is in the "middle" of H_r and stores it together with r in a lookup table.

Hence we define the lookup table Ψ such that $\Psi(r)=\mathbf{z}_w$ and call it the *control function.* Depending on the prediction length m, the lookup table in Fig. 3 has $(\rho+1)^{m+1}$ entries. In reality, one can imagine that r represents an address and $\Psi(\ )$ a memory containing analog values $\mathbf{z}_w$.

Let $\mathbf{w}_m(k)$ be the desired symbolic sequence at time t_k for the next m steps. Then the system must be controlled to $\mathbf{x}_w(t_k)$ with $r(k)=\Theta(\mathbf{w}_m(k))$ and

$$\mathbf{x}_w(t_k)=\mathbf{v}(\Psi(r(k))) . \tag{9}$$

D. Actuator

In a real system, a δ-function-impulse as given in (6) cannot be implemented, but may be approximated by a signal with a nonzero pulse length τ_p:

$$\mathbf{u}(t)=\begin{cases}\hat{\Delta} \text{ if } c_p(t)=1\\ \mathbf{0} \text{ if } c_p(t)=0 .\end{cases} \tag{10}$$

The signal $c_p(t)$ is a sequence of pulses of length τ_p, triggered by the pulse generator $G_p(\mathbf{x})$ each time t_k the trajectory $\mathbf{x}(t)$ crosses through the Poincaré section Σ:

$$c_p(t)=G_p(\mathbf{x})=\begin{cases}1 \text{ if } t_k<t<t_k+\tau_p\\ 0 \text{ otherwise} .\end{cases} \tag{11}$$

The choice of τ_p is not critical, but τ_p must be significantly less than the natural frequency of the system. Finally, the constant $\hat{\Delta}$ must satisfy the equation $\int_{t_k}^{t_k+\tau_p}\hat{\Delta}dt=(\mathbf{x}_w(t_k)-\mathbf{x}(t_k))$. Thus

$$\hat{\Delta}=(\mathbf{x}_w(t_k)-\mathbf{x}(t_k))/\tau_p . \tag{12}$$

The constant $\hat{\Delta}$ may be realized by means of a sample-and-hold circuit, which holds the value of $(\mathbf{x}_w(t_k)-\mathbf{x}(t_k))$ while $c_p(t)=1$, followed by a multiplication by the constant $1/\tau_p$. The switch between $\hat{\Delta}$ and $\mathbf{0}$ in Eq. (10) may be performed by an analog multiplexer.

After the control pulse has been applied, the falling edge of the signal $c_p(t)$ causes a leftshift of symbols through the shift register. At the same time, the new symbol $w(k+m+1)$, which represents the symbol to be performed $(m+1)$ steps ahead, is inserted on the rightmost side.

V. IMPLEMENTATION OF PPC WITH THE LORENZ MODEL

To give an example of the design procedure, we apply the PPC scheme to the well known *Lorenz model:*

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\end{bmatrix}=\begin{bmatrix}\sigma(x_2-x_1)\\ \alpha x_1-x_2-x_1x_3\\ x_1x_2-\beta x_3\end{bmatrix}=\mathbf{f}(\mathbf{x}) . \tag{13}$$

Solving $\mathbf{f}(\mathbf{x})=\mathbf{0}$ for $\mathbf{x}$ we obtain the two unstable equilibrium points $\bar{\mathbf{x}}_1=(-\sqrt{\beta(\alpha-1)},-\sqrt{\beta(\alpha-1)},\alpha-1)^T$ and $\bar{\mathbf{x}}_2=(\sqrt{\beta(\alpha-1)},\sqrt{\beta(\alpha-1)},\alpha-1)^T$. For all simulations, we used the parameter set $(\sigma,\alpha,\beta)=(10,34,\frac{8}{3})$. In the following design process, we determine step by step all the parameters that are required to implement PPC in this system.

A. PPC design process for the Lorenz model

As proposed in the literature [18] we choose the Poincaré section

$$\Sigma=\{\mathbf{x}|x_3=\alpha-1,\bar{\mathbf{x}}_{1_2}<x_2<\bar{\mathbf{x}}_{2_2},\bar{\mathbf{x}}_{1_1}<x_1<\bar{\mathbf{x}}_{2_1}\}$$

where $\mathbf{x}_{i_j}$ is the jth element of the ith equilibrium point. To compute the Poincaré map, we take $\alpha-1$ as the threshold for state x_3. Thus the transformations $\mathbf{v}^{-1}$ and $\mathbf{v}$ are defined by

$$\mathbf{v}^{-1}(\mathbf{x}(t_k))=\begin{bmatrix}x_1(t_k)\\ x_2(t_k)\end{bmatrix}$$

and

$$\mathbf{v}(\mathbf{z}(k))=\begin{bmatrix}z_1(k)\\ z_2(k)\\ \alpha-1\end{bmatrix},$$

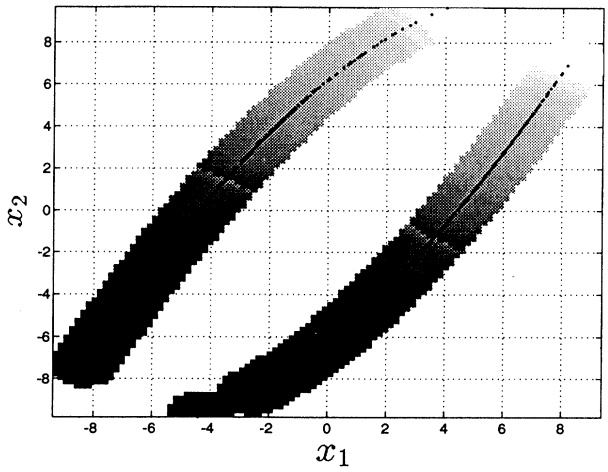
where

$$\mathbf{z}(k)=\begin{bmatrix}z_1(k)\\ z_2(k)\end{bmatrix}, \quad \mathbf{x}_{(t_k)}=\begin{bmatrix}x_1(t_k)\\ x_2(t_k)\\ x_3(t_k)\end{bmatrix} .$$

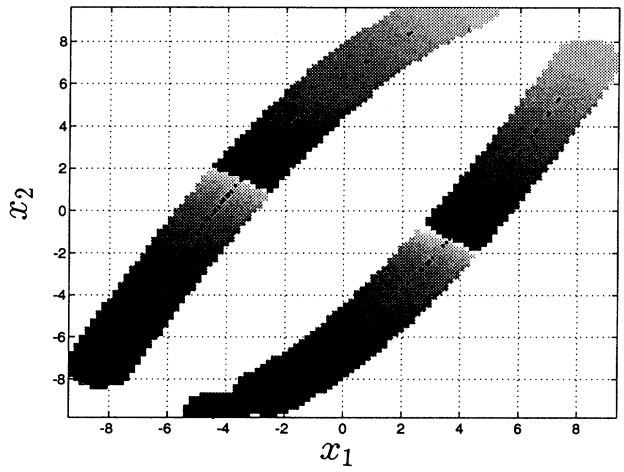
The two-dimensional Poincaré map

$$\mathbf{g}(\mathbf{z})=\begin{bmatrix}g_{1(z_1,z_2)}\\ g_2(z_1,z_2)\end{bmatrix}$$

(see Fig. 4) is separated into two regions. Each region is divided by a discontinuity at about the middle. The maps $\mathbf{g}_{s_0,s_1}$ are continuous and in a one-to-one correspondence in each of the four regions $H_{0,0}$, $H_{0,1}$, $H_{1,0}$, and $H_{1,1}$, as



(a)



(b)

FIG. 4. Two-dimensional Poincaré map $\mathbf{g}(\mathbf{z})$ of the Lorenz model. The map was generated from $\zeta=800$ measured data points (indicated by the black dots). For reasons of better visualization, the map is highlighted in the area around the measured points. The values of g_1 and g_2 are proportional to the gray level; black indicates the minimum value (≈ -9) and white the maximum value ($\approx +9$). (a) The component of $\mathbf{g}$ in the z_1 direction $g_1(z_1,z_2)$. (b) The component of $\mathbf{g}$ in the z_2 direction $g_2(z_1,z_2)$. Note that in this case $z_1=x_1$ and $z_2=x_2$.

required for $\tilde{H}_i$ in Sec. IV. These four branches are then joined such that the two patches in Fig. 4 represent the two sectors H_0 and H_1. In this way, we obtain a two-letter symbolic dynamics for the Lorenz model, giving $\rho=1$.

Since the sectors H_0 and H_1 are known, the symbolic sequence $\mathbf{s}_m$ of length m for each measured point $\mathbf{z}(k)$, $k\in\{0,1,\ldots\}$ can be evaluated. With $r=\Theta(\mathbf{s}_m)$ as defined by (8), each $\mathbf{s}_m$ corresponds to an integer r. Finally, Fig. 5 shows r evaluated for each $\mathbf{z}(k)$ with a prediction length of $m=3$.

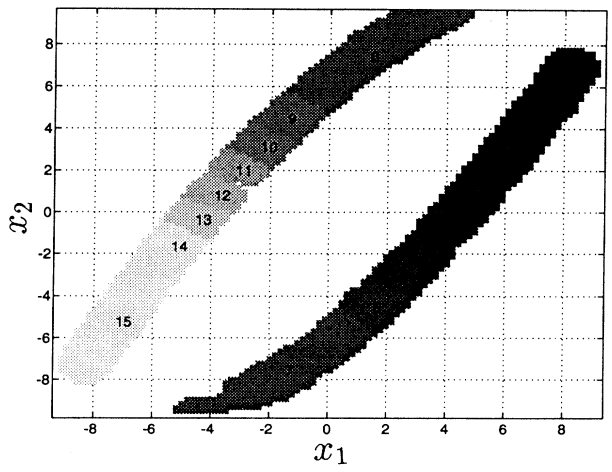


FIG. 5. Integer r evaluated for each $\mathbf{z}$ with a prediction length of $m=3$. The maximum value of r is therefore $(\rho+1)^{m+1}-1=15$. The gray levels in this figure indicate the value of r. One can observe the regions where r is constant. Clearly, these regions are the subsectors H_r.

Figure 5 also indicates the subsectors H_r, where r is constant. Picking a suitable $\mathbf{z}_w$ in the middle of each subsector H_r, we obtain the function $\mathbf{z}_w=\Psi(r)$. Furthermore, up to a prediction length of $m=6$, *all* subsectors H_r, $r=0,1,\ldots,127$, can be identified. We conclude that for this special parameter set the Lorenz model exhibits constrained fully developed chaos of order 6.

B. Simulation results

Before controlling the Lorenz model to certain periodic and nonperiodic orbits, we observe the symbolic se-

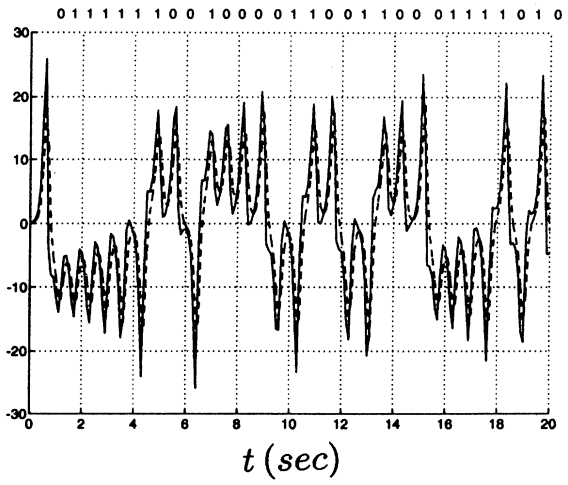


FIG. 6. Symbolic sequence of the uncontrolled Lorenz model, using the parameter set $(\sigma,\alpha,\beta)=(10,34,8/3)$ and starting with the initial conditions $\mathbf{x}(t=0)=(0.2,0.1,25)^T$. The two-letter symbolic sequence is displayed on the top of the figure. Below are the states $x_1(t)$ (solid) and $x_2(t)$ (dashed).

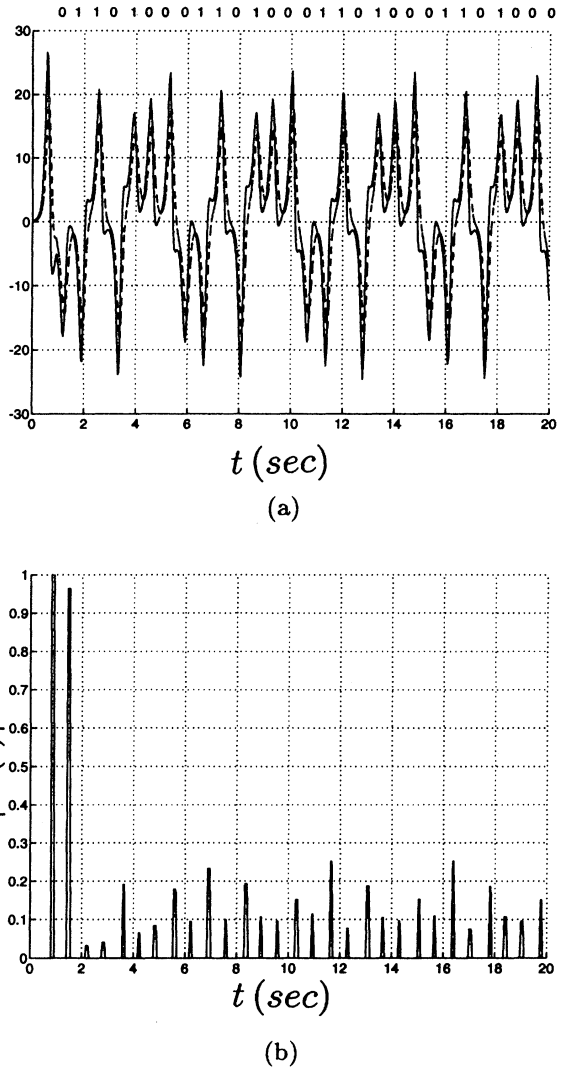


FIG. 7. Lorenz model, controlled to the period-7 orbit $w=\{0,1,1,0,1,0,0,0,1,1,0,1,0,0,\ldots\}$ using a prediction length $m=6$. (a) Symbolic sequence and the time evolution of states $x_1(t)$ (solid) and $x_2(t)$ (dashed) below. (b) The size of the control pulses is $|\Delta(t_k)|=|x_w(t_k)-x(t_k)|$.

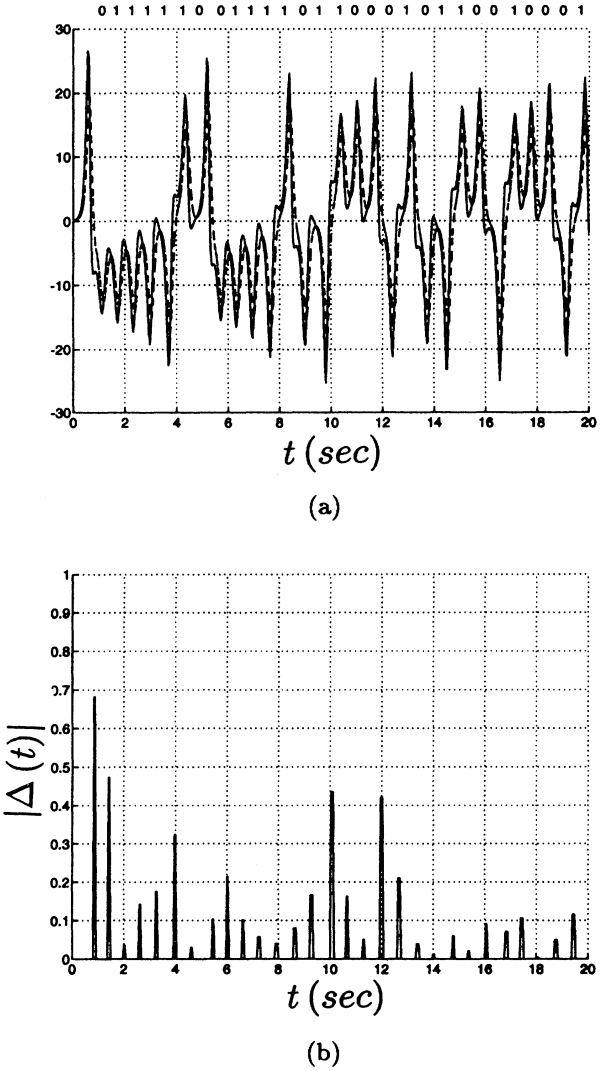


FIG. 8. Lorenz model, controlled to the pseudorandom sequence $w=\{0,1,1,1,1,1,0,0,1,1,1,1,0,1,1,0,0,0,1,\ldots\}$ using a prediction length of $m=6$. (a) Symbolic sequence and the time evolution of states $x_1(t)$ (solid) and $x_2(t)$ (dashed) below. (b) The size of the control pulses is $|\Delta(t_k)|=|\mathbf{x}_w(t_k)-\mathbf{x}(t_k)|$.

quence of the uncontrolled system (Fig. 6). All simulations were performed with a third-order Runge Kutta integrator and a step size of 0.001.

The Lorenz model is first controlled to a period-7 orbit. As mentioned in Sec. V A, all symbolic sequences are admissible up to a length of $m=6$. The control function $\mathbf{z}_w=\Psi(r)$ is therefore a table of 128 discrete inputs and for each of the 128 different values of r there are two analog outputs z_{w_1} and z_{w_2}. We choose $\tau_p=0.1$ sec as the length of control pulses. The time evolution of states $x_1(t)$ and $x_2(t)$ as well as the size of the applied control pulses are shown in Fig. 7. Note that after an initial transient, the average of the control size $E\{|\Delta(t_k)|\}=E\{|\mathbf{x}(t_k)-\mathbf{x}_w(t_k)|\}\approx 0.1$ while the peak-to-peak value $x_{pp}=x_{\max}-x_{\min}$ of x_1 and x_2 is about 50. This means that the average size of the control pulses is only 0.2% of x_{pp}. The size of control pulses can also be compared to the average size of the vector field $E\{|\mathbf{f}(\mathbf{x}(t))|\}\approx 110$. The ratio obtained is $E\{|\Delta(t_k)|\}/E\{|\mathbf{f}(\mathbf{x}(t))|\}\approx 0.004$.

Next, we wish to control the Lorenz model to a pseudorandom symbolic sequence. The time evolution of states $x_1(t)$ and $x_2(t)$ as well as the sizes of the applied control pulses are shown in Fig. 8.

If the prediction length is reduced to $m=5$, the number of subsectors H_{s_5} is only 64 and therefore sectors H_{s_5} are larger on average than sectors H_{s_6}. Note that the average pulse amplitude increases as the prediction length m is reduced. Indeed, Fig. 9 confirms that the control pulses are larger than predicting with $m=5$ com-

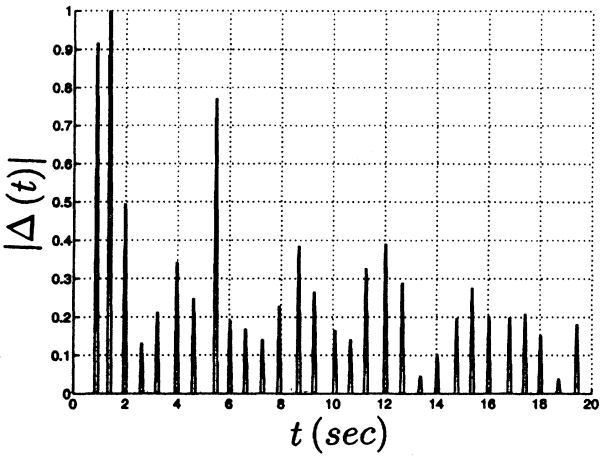


FIG. 9. The size of the control pulses is $|\Delta(t_k)|=|\mathbf{x}_w(t_k)-\mathbf{x}(t_k)|$ while the Lorenz model is controlled to the pseudorandom sequence $w=\{0,1,1,1,1,1,0,0,1,1,1,1,0,1,1,0,0,0,1,\ldots\}$ using a *reduced* prediction length $m=5$.

pared with the case $m=6$. A comparison of Figs. 8(b) and 7(b) illustrates the independence of the average size of control pulses and the desired symbolic sequence **w**.

C. Lorenz model controlled by one state

Since the control vector $\hat{\Delta}$ given in (12) has components in the x_1 and x_2 directions [the third component of $\hat{\Delta}$ equals 0 for all times because $x_{w_3}=v_3(\mathbf{z}_w)=\alpha-1\equiv x_3(t_k)$], the first and second states are required to control the Lorenz model. In this section, we propose a method where just the state x_1 must be accessible in order to control the system. In this case, the control vector **u** in Eq. (6) has the form $\mathbf{u}=(u_1(t),0,0)^T$.

As shown in Fig. 4, all section crossings are arranged on two curves that are approximately the intersection of the unstable two-dimensional manifolds W^u of each equilibrium point with the Poincaré section Σ. Notice that trajectories in the Lorenz model are damped onto W^u along the stable component of the eigenspace. We know that $\mathbf{x}_w$ must lie on W^u and therefore we can exploit the system's stable dynamics to replace one component of the control vector. Making the rough assumption that the trajectory $\mathbf{x}(t)$ near W^u is damped almost *perpendicularly* onto W^u, we obtain a simple solution for u_1. A geometrical approach then gives

$$\hat{\Delta}_1=\frac{1}{\tau_p}\left[\Delta_1+\frac{\Delta_2^2}{\Delta_1}\right]. \tag{14}$$

The first component in Eq. (10) is thus

$$u_1(t)=\begin{cases}\hat{\Delta}_1 \text{ if } c_p(t)=1\\ 0 \text{ if } c_p(t)=0\end{cases}, \tag{15}$$

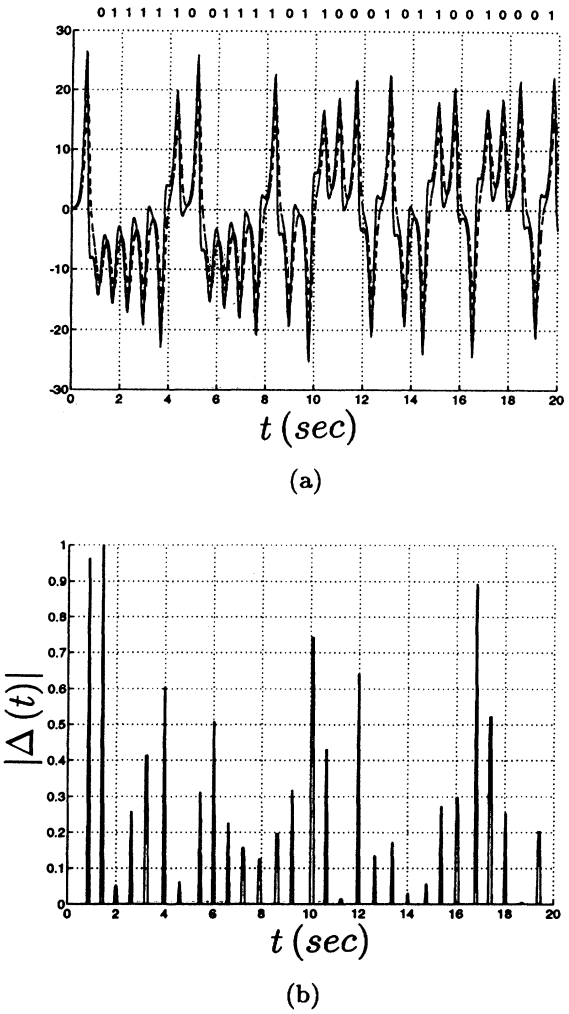


FIG. 10. Lorenz model, controlled to the pseudorandom sequence $w=\{0,1,1,1,1,1,0,0,1,1,1,1,1,0,1,1,0,0,0,1,\ldots\}$ using a prediction length $m=6$. The control pulses were added to the first state only. (a) Symbolic sequence and the time evolution of states $x_1(t)$ (solid) and $x_2(t)$ (dashed) below. (b) The size of the control pulses is $|\tilde{\Delta}_1|$.

where $\Delta_1=x_{w_1}(t_k)-x_1(t_k)$ and $\Delta_2=x_{w_2}(t_k)-x_2(t_k)$. Solutions for certain systems with nonperpendicular damping can also be found, but a generalization for n-dimensional systems is rather difficult. However, if the pulses $u_1(t)$ are limited to an acceptable size (this must be done since Δ_1 can vanish and $\hat{\Delta}_1$ becomes infinite) we can again achieve reasonably small control pulses, as demonstrated in Fig. 10.

When correcting just one state, the control pulses are larger on average than if both states are simultaneously corrected; compare, for example, the control pulses in Fig. 10(b) with those of Figs. 8(b) and 7(b). Nevertheless, even these pulses are small compared to the peak-to-peak value of states x_1 and x_2 and compared to the magnitude of the vector field.

D. Robustness of PPC

In real applications, there will be stochastic and quantization noise in the measurement instruments or the system parameters may change slightly *after* the approximate Poincaré map has been determined. Even in simulations, the effect of a nonideal integrator disturbs the control process. Thus we should discuss how these parasitic effects influence the size of the control pulse $|\Delta|=|\mathbf{x}_w(t_k)-\mathbf{x}(t_k)|$.

In the ideal noise-free case, the trajectory through $\mathbf{x}(t_k)$ next crosses the Poincaré section at the predicted point $\mathbf{x}(t_{k+1})$. Assume that, due to parasitic effects, the first return crossing is not at $\mathbf{x}(t_{k+1})$ but at $\mathbf{x}(t_{k+1})+\mathbf{x}_p(t_{k+1})$. For the correction at time t_{k+1} we obtain

$$|\Delta|=|\mathbf{x}_w(t_{k+1})-\mathbf{x}(t_{k+1})-\mathbf{x}_p(t_{k+1})| ,$$

where the component $\mathbf{x}_p(t_{k+1})$ is caused by parasitic effects during the interval t_k to t_{k+1}.

It appears from simulations that if the prediction length m is sufficiently large, then $\mathbf{x}_w(t_{k+1})-\mathbf{x}(t_{k+1})$ becomes small and the *main* task of the control pulse is to compensate for the component $\mathbf{x}_p(t_{k+1})$. Parasitic effects with offset, such as parameter variations, generally produce a larger $\mathbf{x}_p(t_{k+1})$ and therefore stronger control pulses are required.

In order to simulate realistic conditions, a -50-dB noise source [by a -50-dB noise source we mean that $10\log_{10}(\bar{P}_{\mathbf{n}}/\bar{P}_{\mathbf{f}(\mathbf{x})})=-50$ dB, where $\bar{P}_{\mathbf{f}(\mathbf{x})}$ and $\bar{P}_{\mathbf{n}}$ are the mean powers of $\mathbf{f}(\mathbf{x})$ and $\mathbf{n}(t)$, respectively] $\mathbf{n}(t)$ was added to the state equations in (13) and all system parameters were changed by 1% after the control function Ψ had been determined. Furthermore, we assume that only the first state is accessible and so use the method in Sec. V C. Figure 11(b) shows the size of the control pulses for this situation. Even though the control pulse are larger than in the noise-free case, the correct symbolic sequence is still produced, as predicted.

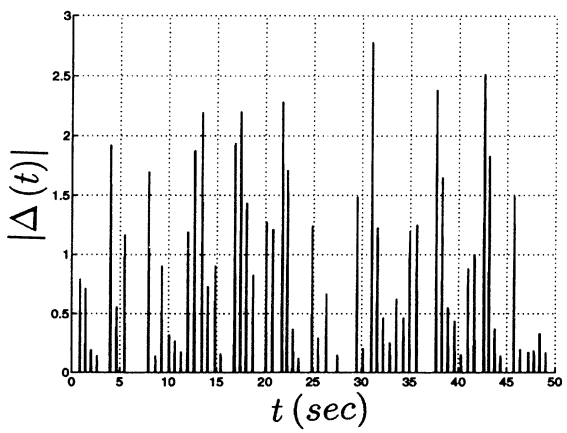


FIG. 11. Simulated control pulses when the process is disturbed by parasitic effects.

VI. CONCLUSIONS

Predictive Poincaré control [22] is a *global* control theory that permits one to stabilize a predefined solution of a dynamical system using only *small* control pulses. This has been proven for n-dimensional chaotic systems, provided that certain states of the system are accessible. The predefined solution can be a periodic or nonperiodic trajectory, expressed by a periodic or nonperiodic symbolic sequence. We have applied the general PPC scheme to the well-known Lorenz model and have shown that the method is robust in the presence of parasitic effects.

Because of strong damping along the unstable manifold, it is possible to control the Lorenz model with just one state; we conjecture that, in general, the number of states required for predictive Poincaré control is proportional to the dimensions of the unstable manifolds W^u of the system. PPC provides a method for controlling chaotic systems with large time constants even if only small amounts of control energy are available. Alternatively, PPC could be employed in environments where only small control energy is desired, for example, in medical applications.

The technique suggests that greater flexibility may be achieved in a dynamical system by first introducing constrained fully developed chaos and then controlling it by means of PPC. Our technique also offers a direct means of modulating digital information onto a chaotic carrier signal for spread spectrum applications; for further details, see [22–24].

ACKNOWLEDGMENTS

This work is supported in part by the President's Resarch Fund of the University College Dublin and by the Swiss National Science Foundation under Contract No. 20-36562.92.

APPENDIX: PROOF OF THEOREM 1

Definition 1 suggests an inductive proof of Theorem 1. The proof itself, together with Fig. 2, which illustrates the first two induction steps, gives a deeper insight into the functioning of PPC.

Proof: $m=1$. To prove the first induction step we use the definition of H_j. The induction process can be divided into three steps: a forward mapping, a splitting, and a backward or inverse mapping.

(i) Forward mapping:

$$\mathbf{g}_{s_0,s_1}: H_{s_0,s_1}\mapsto H^*_{s_0,s_1} . \tag{A1}$$

Maps $\mathbf{g}_{s_0,s_1}$ are in a one-to-one correspondence and continuous, $H^*_{s_0,s_1}$ are compact, connected, and $H^*_{s_1,s_1}\subseteq H_{s_1}$, as given in Definition 1.

(ii) Splitting. New subsectors $H^*_{s_0,s_1,s_2}$ are created by splitting up $H^*_{s_0,s_1}$ so that

$$H^*_{s_0,s_1,s_2}=H^*_{s_0,s_1}\cap H_{s_1,s_2} . \tag{A2}$$

According to Definition 1, H_{s_1,s_2} is compact and connected and therefore $H^*_{s_0,s_1,s_2}$ is compact and connected too

because $H^*_{s_0,s_1,s_2} \subseteq H_{s_1,s_2}$. It is also clear that $\bigcup_{s_2=0}^{s_2=\rho} H^*_{s_0,s_1,s_2} = H^*_{s_0,s_1}$ and all $H^*_{s_0,s_1,s_2}$ are disjoint.

(iii) Inverse mapping. To complete the procedure for $m=1$, the sectors $H^*_{s_0,s_1,s_2}$ are mapped back:

$$\mathbf{g}^{-1}_{s_0,s_1}: H^*_{s_0,s_1,s_2} \mapsto H_{s_0,s_1,s_2} . \tag{A3}$$

Since $\mathbf{g}^{-1}_{s_0,s_1}$ is a one-to-one map and continuous (Definition 1), we obtain the sectors H_{s_0,s_1,s_2} which are also compact, connected, disjoint, and $\bigcup_{s_2=0}^{s_2=\rho} H_{s_0,s_1,s_2} = H_{s_0,s_1}$; therefore $H_{s_0,s_1,s_2} \subseteq H_{s_0,s_1}$.

For $m>1$, we now use the following abbreviation for the forward maps:

$$\mathbf{g}_{[\mathbf{s}_i]} =: \mathbf{g}_{s_i,s_{i+1}} \circ \mathbf{g}_{s_1,s_2} \circ \cdots \circ \mathbf{g}_{s_0,s_1} \tag{A4}$$

and for the inverse maps

$$\mathbf{g}^{-1}_{[\mathbf{s}_i]} =: \mathbf{g}^{-1}_{s_0,s_1} \circ \mathbf{g}^{-1}_{s_1,s_2} \circ \cdots \circ \mathbf{g}^{-1}_{s_i,s_{i+1}} . \tag{A5}$$

Recalling some results from topology [25,26], we know that $\mathbf{g}_{[\mathbf{s}_i]}$ and $\mathbf{g}^{-1}_{[\mathbf{s}_i]}$ are continuous and in a one-to-one correspondence if each $\mathbf{g}_{j,j'}$ is a continuous and one-to-one map for all $j,j'=0,\ldots,\rho$. We now proceed with the three induction steps for the general case.

(i) Forward mapping. The map $\mathbf{g}_{[\mathbf{s}_{m-2}]}$ from the previous induction step is used to generate $H^{(m-1)*}_{\mathbf{s}_m}$:

$$\mathbf{g}_{[\mathbf{s}_{m-2}]}: H_{\mathbf{s}_m} \mapsto H^{(m-1)*}_{\mathbf{s}_m} .$$

Then the continuous map $\mathbf{g}_{s_{m-1},s_m}$ is applied to $H^{(m-1)*}_{\mathbf{s}_m}$:

$$\mathbf{g}_{s_{m-1},s_m}: H^{(m-1)*}_{\mathbf{s}_m} \mapsto H^{m*}_{\mathbf{s}_m} . \tag{A6}$$

$H^{(m-1)*}_{\mathbf{s}_m} \subseteq H_{s_{m-1},s_m}$ is compact, connected, and with $\mathbf{g}_{s_{m-1},s_m}$ it follows from Definition 1 that $H^{(m-1)*}_{\mathbf{s}_m} \subseteq H_{s_m}$.

(ii) Splitting:

$$H^{m*}_{\mathbf{s}_{m+1}} = H^{m*}_{\mathbf{s}_m} \cap H_{s_m,s_{m+1}} . \tag{A7}$$

$H^{m*}_{\mathbf{s}_{m+1}}$ is compact and connected, $\bigcup_{s_{m+1}=0}^{s_{m+1}=\rho} H^{m*}_{\mathbf{s}_{m+1}} = H^{m*}_{\mathbf{s}_m}$ and all $H^{m*}_{\mathbf{s}_{m+1}}$ are disjoint.

(iii) Inverse mapping. The inverse mapping follows in two steps. First $\mathbf{g}^{-1}_{s_{m-1},s_m}$ is applied:

$$\mathbf{g}^{-1}_{s_{m-1},s_m}: H^{m*}_{\mathbf{s}_{m+1}} \mapsto H^{(m-1)*}_{\mathbf{s}_{m+1}} , \tag{A8}$$

where subsectors $H^{(m-1)*}_{\mathbf{s}_{m+1}}$ are compact, connected, disjoint, and $\bigcup_{s_{m+1}=0}^{s_{m+1}=\rho} H^{(m-1)*}_{\mathbf{s}_{m+1}} = H^{(m-1)*}_{\mathbf{s}_m}$. Therefore $H^{(m-1)*}_{\mathbf{s}_{m+1}}$ is a subset of $H^{(m-1)*}_{\mathbf{s}_m}$.

In the second step, $H^{(m-1)*}_{\mathbf{s}_{m+1}}$ is mapped back by $\mathbf{g}^{-1}_{[\mathbf{s}_{m-2}]}$:

$$\mathbf{g}^{-1}_{[\mathbf{s}_{m-2}]}: H^{(m-1)*}_{\mathbf{s}_{m+1}} \mapsto H_{\mathbf{s}_{m+1}} ,$$

where $H_{\mathbf{s}_{m+1}}$ is compact, connected, disjoint, and $\bigcup_{s_{m+1}=0}^{s_{m+1}=\rho} H_{\mathbf{s}_{m+1}} = H_{\mathbf{s}_m}$. Hence, $H_{\mathbf{s}_{m+1}}$ is a subset of $H_{\mathbf{s}_m}$ and Theorem 1 is proven.

The inverse maps $\mathbf{g}^{-1}_{s_{m-1},s_m}$ and $\mathbf{g}^{-1}_{[\mathbf{s}_{m-2}]}$ can be combined to the new one-to-one and continuous map $\mathbf{g}^{-1}_{[\mathbf{s}_{m-1}]}$, using the convention of (A5):

$$\mathbf{g}^{-1}_{[\mathbf{s}_{m-1}]}: H^{m*}_{\mathbf{s}_{m+1}} \mapsto H_{\mathbf{s}_{m+1}} .$$

[1] S. Hayes, C. Grebogi, and E. Ott, Phys. Rev. Lett. **70**, 3031 (1993).

[2] L. M. Pecora and T. L. Carroll, Phys. Rev. Lett. **64**, 821 (1990).

[3] A. V. Oppenheim, G. W. Wornell, S. H. Isabelle, and K. M. Cuomo (unpublished).

[4] H. Dedieu, M. P. Kennedy, and M. Hasler, IEEE Trans. Circuits Syst. **CAS-40**, 634 (1993).

[5] M. Hasler, H. Dedieu, J. Schweizer, and M. P. Kennedy, in *Nonlinear Dynamics of Electronic Systems,* edited by A. C. Davis and W. Schwarz (World Scientific, Singapore, 1994), p. 224.

[6] M. Itoh, H. Murakami, K. S. Halle, and L. O. Chua (unpublished).

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

[8] Bo Peng, V. Petrov, and K. Showalter, J. Phys. Chem. **95**, 4957 (1991).

[9] E. R. Hunt, Phys. Rev. Lett. **67**, 1953 (1991).

[10] S. Bielawski, M. Bouazaoui, D. Derozier, and P. Glorieux, Phys. Rev. Lett. **47**, 3276 (1992).

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

[12] M. J. Ogorzalek, IEEE Trans. Circuits Syst. **CAS-40**, 693 and 700 (1993).

[13] Z. Galias, J. Nossek, and M. J. Ogorzalek, in *Nonlinear Dynamics of Electronic Systems,* edited by M. J. Ogorzalek (University of Mining and Metallurgy, Krakow, 1994), p. 173.

[14] J. Singer, Y.-Z. Wang, and Haim H. Bau, Phys. Rev. Lett. **66**, 1123 (1991).

[15] G. Chen and X. Dong, J. Circuits Syst. Comput. **3**, 139 (1993).

[16] K. Pyragas, Phys. Lett. A **170**, 421 (1992).

[17] J. Guckenheimer and P. Holmes, *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields* (Springer-Verlag, Berlin, 1983).

[18] Bai-Lin Hao, *Elementary Symbolic Dynamics* (World Scientific, Singapore, 1989).

[19] M. P. Kennedy, IEEE Trans Circuits Syst. **CAS-40**, 640 and 657 (1993).

[20] M. P. Kennedy, in *Circuits and Systems Tutorials,* edited

by C. Toumazou (Institute of Electrical and Electronics Engineers, London, 1994), Chap. 6.1.

[21] R. A. Lorentz, in *Multivariant Birkhoff Interpolation,* edited by A. Dold, Lecture Notes in Mathematics Vol. 1516 (Springer-Verlag, Berlin, 1991).

[22] J. Schweizer and M. P. Kennedy, in *Irish DSP and Control Colloquim,* edited by J. Ringwood and A. Murray (Dublin University Press, Dublin, 1994), p. 125.

[23] J. Schweizer and M. P. Kennedy, Ref. [13], p. 51.

[24] J. Schweizer, Master's thesis, University College Dublin, 1994 (unpublished).

[25] S. Lipschutz, *General Topology* (McGraw-Hill, New York, 1965).

[26] M. Henle, *A Combinatorial Introduction to Topology* (Freeman, San Francisco, 1979).