Communication with chemical chaos in the presence of noise

Milos Dolnik
Department of Chemistry and Center for Complex Systems, Brandeis University, Waltham, Massachusetts 02254-9110

Erik M. Bollt
Mathematics Department, 572 Holloway Road, U.S. Naval Academy, Annapolis, Maryland 21402-5002

(Received 4 August 1997; accepted for publication 4 March 1998)

We use control of chaos to encode information into the oscillations of the Belousov-Zhabotinsky reaction. An arbitrary binary message is encoded by forcing the chaotic oscillations to follow a specified trajectory. The information manipulating control requires only small perturbations to vary the binary message. In this paper we extend our recent theoretical work [Bollt and Dolnik, Phys. Rev. E **64**, 1196 (1990)] by introducing a new and simplified encoding technique which can be utilized in the presence of experimental noise. We numerically and theoretically study several practical aspects of controlling symbol dynamics including: modeling noisy time-series, learning underlying symbol dynamics, and evaluation of derivatives for control by observing system responses to an intelligent and deliberate sequence of input parameter variations. All of the modeling techniques incorporated here are ultimately designed to learn and control symbol dynamics of experimental data known only as an observed time-series; the simulation assumes no global model. We find that noise affects reliability of encoding information and may cause coding errors. But, if the level of noise is confined to relatively small values, which are achievable in experiments, the control mechanism is robust to the noise. Thus we can still produce a desired symbolic code. However, scarce errors in encoding may occur due to rare but large fluctuations. These errors may be corrected during the decoding process by a variation of the filtering technique suggested by Rosa *et al.* [Phys. Rev. Lett. **78**, 1247 (1997)]. © *1998 American Institute of Physics.* [S1054-1500(98)01402-5]

Recent applications of controlling chaotic dynamical systems focus on manipulation of information flow, for communications, encryption, and targeting. A particularly interesting possibility concerns encoding information into a chemical system. We propose an alternative to a popular and intuitive, but not well founded belief which is an analogy to electronic computers, that information is stored in a biological version of physical switching devices. In contrast, an analogy between chemical systems and biological systems[1] implies an interesting possibility that living systems may store information in underlying dynamics. In our previous work,[2,3] we demonstrated the possibility of encoding information into chemical chaos utilizing a model of the Belousov-Zhabotinsky (BZ) reaction. In this work, we extend and improve these methods with the aim of successfully implementing a message into a controlled oscillatory reaction in a ''real world'' noisy laboratory environment. We have therefore included numerical stochastic studies to argue that the improved technique promises success given measurement tolerances within laboratory feasible levels.

I. INTRODUCTION

In recent years several methods of encoding information into chaotic oscillators have been proposed. One class of techniques is based on the synchronization of chaotic systems between both receiver and transmitter. The receiver can then strip off the commonly known chaotic signal to recover the encoded information signal.[4–9] The main purpose of this technique is for encryption and secure communications. The subject of this paper concerns the other main class of techniques, based on controlling symbol dynamics, to encode a binary message by forcing the chaotic oscillations to follow a desired trajectory. Sensitive dependence on initial conditions allows manipulation of the trajectory through a symbol partition, and hence the corresponding binary message, using only small parameter perturbations.[2,10,11]

An unavoidable fact of real world experiments is that observations of a chaotic system are affected by noise. In fact, even highly accurate, but finite precision numerical experiments with chaos are profoundly influenced by noise. Given finite numerical precision, and therefore rounding errors, the presence of sensitive dependence to initial conditions renders impossible any long term prediction of chaotic oscillations. For years, this was considered to be a disadvantage of chaos, but control of chaos techniques actually take advantage of sensitive dependence to initial conditions. Generically, the distance between two close points grows exponentially with time. The implication is that long term prediction is impossible because even a small error in specifying the system states yields drastically varied results. In terms of control, this means that a small and judiciously specified control perturbation can result in a large and varied system response.

In this work we demonstrate how the presence of experimental noise affects the reliability of encoding. It is rather evident that when increasing the noise level, the probability of error occurrence also increases. In this paper we describe how our control technique copes in the case of experiments with realistic noise levels.

Our paper is organized in the following way. In Sec. II we present the Györgyi-Field model[12] of the BZ reaction, which is used in our numerical experiments. We introduce a stochastic component to the deterministic model of the BZ reaction and thus we create a numerical analog of the real experimental system. In Sec. III we review the symbol dynamics description of the one-dimensional map. We also show how the one-dimensional map can be recovered from the scattered noisy data. Then we describe a simple method of learning the observed grammar, based on revealing all observed codes. In Sec. IV we present the technique to control the symbolic code sequences. Control of symbol dynamics requires determination of the maps responses due to parameter perturbations. We demonstrate how the map changes can be determined from the experiments, even in the presence of noise. Then in Sec. V we present numerical experiments in which we have encoded a short message in the presence of noise. By increasing the standard deviation of the imposed noise, we search for a critical noise level. We demonstrate that for the experimental noise confined below this critical value, the encoding technique works in a reliable manner.

II. MODEL OF THE BZ REACTION

A simple model of the BZ reaction in a continuous-flow stirred-tank reactor (CSTR) suggested by Györgyi and Field[12] consists of three ODE's:

$$\frac{dX}{dt}=-k_1HXY+k_2AH^2Y-2k_3X^2+0.5[k_4(HA)^{1.5}\times(C-Z)X^{0.5}-k_5XZ]+k_0(X_o-X),$$

$$\frac{dZ}{dt}=k_4(HA)^{1.5}(C-Z)X^{0.5}-k_5XZ-\alpha k_6VZ-\beta k_7BZ+k_0(Z_o-Z), \tag{1}$$

$$\frac{dV}{dt}=2k_1HXY+k_2AH^2Y+k_3X^2-\alpha k_6VZ+k_0(V_o-V),$$

where

$$Y=\frac{\alpha k_6ZV}{k_1HX+k_2AH^2+k_0}. \tag{2}$$

X, Z, V and Y denote the concentrations of $HBrO_2$, Ce^{4+}, bromomalonic acid and Br^-, respectively. The subscript ''o'' indicates the concentrations on the input of the reactor. The parameter A represents the concentration of $HBrO_3$, B is the concentration of malonic acid, and C is the total concentration of the catalyst $C=[Ce^{4+}]+[Ce^{3+}]$. The values of these parameters, as well as the kinetic parameters denoted α and β and the rate constants k_1–k_6, are given in Table I. The parameter k_0 denotes the flow rate and in this study we choose it to be the adjustable control parameter, written as λ. A nominal value of the control parameter $\lambda_0=3.5\times10^{-4}\ s^{-1}$ is within the widest chaotic window of the low flow rate parameter bifurcation diagram.[12] To control symbol sequences, we apply small perturbations to the control parameter λ.

TABLE I. Parameters used in the simulations.

Parameter	Value	
k_1	4×10^6	$dm^6\ mol^{-2}\ s^{-1}$
k_2	2	$dm^6\ mol^{-2}\ s^{-1}$
k_3	3×10^3	$dm^3\ mol^{-1}\ s^{-1}$
k_4	55.2	$dm^{7.5}\ mol^{-2.5}\ s^{-1}$
k_5	7×10^3	$dm^3\ mol^{-1}\ s^{-1}$
k_6	0.09	$dm^3\ mol^{-1}\ s^{-1}$
k_7	0.23	$dm^3\ mol^{-1}\ s^{-1}$
α	600/9	
β	8/23	
A_0	0.1	$mol\ dm^{-3}$
B	0.25	$mol\ dm^{-3}$
H	0.26	$mol\ dm^{-3}$
C	8.33×10^{-4}	$mol\ dm^{-3}$
X_o	0	$mol\ dm^{-3}$
Z_o	0	$mol\ dm^{-3}$
V_o	0	$mol\ dm^{-3}$

Previously, we have employed the model (1), with parameters from Table I, to demonstrate the encoding technique.[2] In the present study, we introduce small fluctuations to one of the fixed parameters to demonstrate reliability of the encoding technique in the presence of noise. In a real chemical experiment in a flow reactor, there are several possible and unavoidable sources of experimental noise, including imperfect mixing, flow rate fluctuations and temperature fluctuations.[13] It is cumbersome to introduce every source of noise into the model. Rather, for simplicity, we assume that only one parameter is affected by noise and all other parameters remain fixed (but affected due to coupling).

We allow parameter A to experience noise according to the equation,

$$A=A_0(1+\xi_n), \tag{3}$$

where A_0 is the average value of parameter A, and ξ_n is Gaussian white noise with zero mean value and a standard deviation of σ_ξ. Our map based simulation assumes that the noise randomly affects the parameter A at the moment of intersection with the Poincaré surface, and that A is held constant during the time of flight between piercings. To accurately integrate the stiff differential equations, Eq. (1), we use the 4th order modified semi-implicit Runge-Kutta method, with automatic step length control.

III. ONE-DIMENSIONAL MAPS AND THEIR SYMBOLIC CODES

Rather than directly studying trajectories in the full three-dimensional phase space, we analyze the sequence of maxima of $Z(t)$, which can be considered to be a special case of the Poincaré surface of section mapping technique. The sequence of successive oscillatory maxima of the noise-free model of the BZ reaction defines a map,

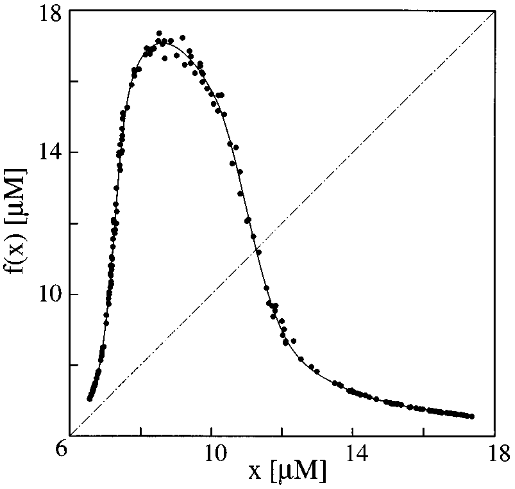

FIG. 1. The one-dimensional map $f(x)$ for the nominal value of control parameter $\lambda_0=3.5\times10^{-4}\ \mathrm{s}^{-1}$. The data set contains 200 successive intersections with the Poincaré surface. Data are obtained from simulations with added Gaussian white noise with the standard deviation $\sigma_\xi=0.001$. The solid line depicts the cubic smoothing spline fit calculated by the general cross validation method (Ref. 17).

$$x_{n+1}=f_{\lambda_0}(x_n). \tag{4}$$

which is truly one dimensional after the transients have settled. However, in the presence of noise, as modeled by Eq. (3), the model Eq. (1) generates scattered data points of the ''next-maxima map,'' shown in Fig. 1. This picture agrees qualitatively with experimental data.[14–16] Given purely experimental data, we will not know the underlying noise-free map to which noise has been added, but we can recover an approximation of this underlying noise-free map, by a cubic spline-based least-squares fitting procedure. We employ the procedure named CUBGCV[17] to fit the noisy data points by cubic smoothing spline. The degree of smoothing is chosen to minimize the generalized cross validation. To account for unequally spaced data, which is distributed according to the invariant measure, we use nonuniform weights w_i defined,

$$w_i=\sqrt{\frac{x_{i+1}-x_{i-1}}{2\langle\Delta x\rangle}}, \tag{5}$$

where $\langle\Delta x\rangle$ is the average spacing:

$$\langle\Delta x\rangle=\frac{1}{N-1}\sum_{i=1}^{N-1}(x_{i+1}-x_i)=\frac{x_N-x_1}{N-1}. \tag{6}$$

We have tested several other weighting functions, including uniformly weighted data, but the weights definition, Eq. (5), produced the best smoothed fit of the noisy data, in comparison to the noise-free one-dimensional (1-D) map.[2]

To represent the symbolic dynamics of the map, we assign a binary code, for every initial condition x_0. The maximum of the cubic smoothing spline defines the decision point d. For $x_0\leqslant d$ we designate the first bit of the symbolic code as ''0,'' and for $x_0>d$, we designate the first bit as ''1.'' A decimal point character (.) separates the first bit of the initial state from successive bits which represent successive iterations x_i. We call the first bit of an n-bit symbol sequence, ''the most significant bit'' and the last bit, ''the least significant bit.''

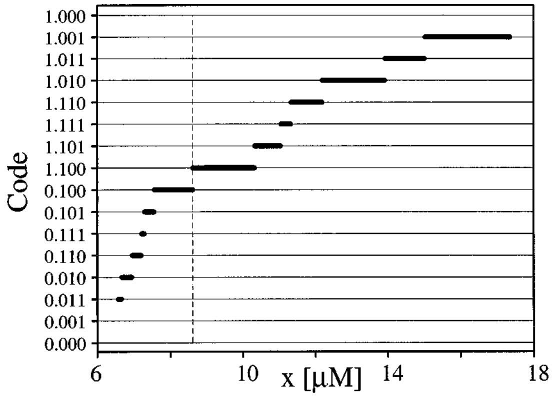

FIG. 2. Symbolic binary codes produced by three iterations of the cubic smoothed spline map. Symbolic codes are Gray ordered along the vertical axis. Dashed line illustrates the decision point d. Empty code levels represent the forbidden codes 0.000, 0.001 and 1.000.

Figure 2 shows the 4-bit symbol sequences produced by initial conditions plus their three successive iterations of the cubic smoothing spline map. The codes are displayed along the vertical axis according to the Gray-code order.[18] It is evident that the symbolic codes are Gray ordered for gradually increasing initial conditions x_0 from $x_{\min}$ to $x_{\max}$. The Gray ordering of symbol sequences has the important property that the binary representations, of any two adjacent codes, differ by exactly one bit, which is key to the equivalence between the symbol dynamics, and the dynamics in the phase space. This property is important for encoding messages into the symbol dynamics through sensitively controlling trajectories of the corresponding phase space variable x, as we present below.

An important technical issue arises from the nonexistence of some codes, as not all 4-bit binary codes are produced by the 1-D map. We observe that for the parameter values chosen, and in the 4-bit representation, the codes ''0.000,'' ''0.001'' and ''1.000'' are not present; these levels in Fig. 2 are unoccupied. We can take this directly as the grammar of the symbol dynamics, but in this special case, we can reduce the grammar. These three codes have a 3-bit pattern in common: each includes three ''0'' bits in a row. Therefore this is the *observed* grammatical restriction on the *observed* symbol dynamics. In terms of the phase variable, this means that the chaotic time series never contains three successive oscillatory maxima with $Z<d$. To achieve our goal of communicating a digital message *using only small controls*, we must respect this grammar.

IV. CONTROLLING THE SYMBOLIC CODE SEQUENCE

In this section, we describe how to encode information into chaotic oscillations by controlling corresponding symbolic codes, and we extend our previous methods[2] to allow for the experimentally feasible situation in which (noisy) data and the system responses may only be sampled on-the-fly. We also describe the encoding technique using the cubic

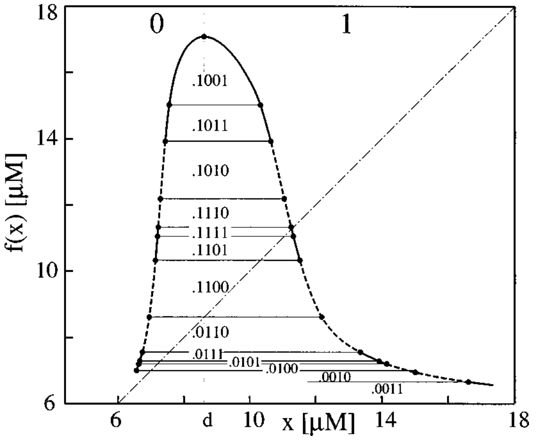

FIG. 3. Cubic smoothing spline of the map $f(x)$ divided into domains of the identical 5-bit symbolic codes. The solid lines indicate the map sections with symbolic codes containing a bit "1" in the least significant position, while dashed line indicate a bit "0" in the least significant position. Point d shows the decision point. The diagonal (dash-doted line) crosses the map at the domain with symbolic code "1.1111."

smoothed spline map $f(x)$ divided into sections with the identical symbolic codes. Figure 3 shows the assignment of 5-bit symbolic codes to corresponding sections of the 1-D map. Note that, except in the vicinity of the symbol partition, there is always a region with a "1" bit in the least significant position adjacent to a region with a "0" bit in the least significant position, and *vice versa*. This follows from the Gray ordering of codes found on the interval, which has the additional property that two adjacent regions with differing bits in least significant positions must agree in all the other bits. The equivalence between the dynamics of the phase space variable in the topology of the interval, and the dynamics of the shift on symbol sequences in the symbol space topology, is in terms of a semiconjugacy (a continuous and onto change of coordinates). This is due to the Gray-code ordering of the symbolic codes on the unit interval, as indicated by the monotone nondecreasing coding function in Fig. 2. In fact, the lack of any observed attracting periodic set on the interval is evidence of a faithful representation of the chaotic oscillations in the symbol dynamics, i.e., a full conjugacy. Thus a small perturbation of the map in the phase variable representation results in encoding a desired bit into the least significant position in the symbol sequence representation.

Figure 4 shows a schematic representation of the control technique on a blow-up of the 1-D map. Let x_i be the ith iterate on the surface of section, and, for example, we want to encode a "0" bit into the least significant position of the 4-bit symbolic representation of the next iterate. In this case, the uncontrolled iteration causes the "1" bit to shift into the least significant position because $f_{\lambda_0}(x_i)$ has the symbolic code "0.111." However, the encoding of a "0" bit into the least significant position of the symbolic representation requires a small controlling parameter perturbation. The perturbation $\delta\lambda$ must cause a variation in the iteration of the 1-D map $f_{\lambda_0+\delta\lambda}(x_i)$ from the symbolic region "0.111" to the adjacent "0.110" region. This "upward" shift in the one-dimensional map is illustrated by the dot-dashed line. To minimize the effect of noise, we target the code centers, which are depicted by small triangles in Fig. 4. Similarly in the jth iteration, when "1.101" is assigned to the uncontrolled iteration $f_{\lambda_0}(x_j)$, a "0" bit can be encoded into the least significant position by application of a small perturbation. In this case, the adjacent coded region with a "0" bit in the least significant position, i.e., the region coded "1.100" is located below the like-coded region of points corresponding to "1.101." This means that the 1-D map must be perturbed "downward," as illustrated by the double dot-dashed line.

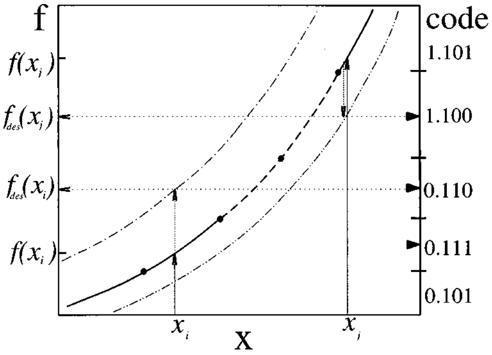

FIG. 4. A schematic representation of the encoding technique—enlargement of the smoothing spline map. Two examples of shifting "0" in the least important bit when the natural iteration would produce shifts in a "1" bit. The centers of the symbolic codes (solid triangles) are the desired values. Dot-dashed lines represent the desired "shifted" 1-D maps. $f(x)$ are the uncontrolled iterations and $f_{\text{des}}(x)$ are the desired iterations.

Encoding an arbitrary message requires a sequence of small variations to the symbol representations, which corresponds to a sequence of displacements $\Delta f(x)$. Let $\Delta f(x)$ denote the difference between the desired iteration $f_{\text{des}}(x)$ (corresponding to a desired bit) and the uncontrolled iteration $f(x)$:

$$\Delta f(x) = f_{\text{des}}(x) - f(x). \tag{7}$$

Then as presented in our previous work,[2] the formula for the required parameter perturbation $\delta\lambda$ is

$$\delta\lambda = \frac{\Delta f(x)}{(\partial f^1/\partial\lambda)\,|_{(x,\lambda_0)}}. \tag{8}$$

The partial derivative $\partial f^1/\partial\lambda$ represents the "dynamic shift" of the one-dimensional map. The original system is described by the three ODE's, and there are transient effects connected with parameter perturbations. Our simulations with the noise free system[2] revealed that the intersection with the Poincaré surface does not correspond to a perturbed 1-D map, immediately following a parameter perturbation. Instead, the response is significantly larger than predicted by the corresponding 1-D model. However, if the control parameter is fixed at the new parameter value, then after the first intersection, all subsequent intersections with the Poincaré surface settle onto the attractor of the perturbed 1-D map. To distinguish between the transient (first) response and the transient-free (steady) response, we use notation f^1 and f, respectively. We denote the dynamic map by $f^1(x)$, and the static map by $f(x)$.

The fact, that an arbitrary message requires controlling parameter perturbations during every iteration, is an important technical issue which prevents us from returning to the nominal parameter value λ_0. Instead, the $(n+1)$th ''uncontrolled'' iteration of x assumes that the parameter from the nth iteration λ_n is unchanged. Therefore instead of the nominal 1-D map $f_{\lambda_0}(x)$, the estimate of $f(x)$ in Eq. (7) uses the map $f_{\lambda_n}(x)$ which corresponds to the parameter λ_n,

$$f(x)=f_{\lambda_n}(x_n). \tag{9}$$

We estimate $f_{\lambda_n}(x_n)$ by linearization around the nominal map $f_{\lambda_0}(x)$:

$$f_{\lambda_n}(x_n)=f_{\lambda_0}(x_n)+(\lambda_n-\lambda_0)\left.\frac{\partial f}{\partial\lambda}\right|_{(x_n,\lambda_0)}. \tag{10}$$

Combining Eqs. (7)–(10) we obtain the required parameter perturbation $\delta\lambda_{n+1}$.

$$\delta\lambda_{n+1}=\lambda_{n+1}-\lambda_n$$
$$=\frac{f_{\mathrm{des}}(x_n)-f_{\lambda_0}(x_n)-(\lambda_n-\lambda_0)\partial f/\partial\lambda|_{(x_n,\lambda_0)}}{\partial f^1/\partial\lambda|_{(x_n,\lambda_0)}}. \tag{11}$$

The derivative $\partial f/\partial\lambda$ characterizes the rate of the static map variations. Figure 5(a) shows the one-dimensional maps obtained for 50 successive Poincaré sections, for three values of the fixed flow rate; solid circles and the dashed line belong to the parameter $\lambda=3.45\times10^{-4}\ \mathrm{s}^{-1}$, while crosses and the dotted line belong to $\lambda=3.55\times10^{-4}\ \mathrm{s}^{-1}$. Each time a simulation with a new value of λ is started, the first intersection with Poincaré surface is neglected, because it is not part of static 1-D map. The solid line depicts the cubic smoothing spline fit for the nominal value of the fixed flow rate $\lambda_0=3.50\times10^{-4}\ \mathrm{s}^{-1}$. We estimate the derivatives $\partial f/\partial\lambda$ from the equation,

$$\left.\frac{\partial f}{\partial\lambda}\right|_{(x,\lambda_0)}=\frac{f_{\lambda_0+\epsilon}(x)-f_{\lambda_0}(x)}{\epsilon}, \tag{12}$$

where $\epsilon=\pm5\times10^{-6}\ \mathrm{s}^{-1}$. Figure 5(b) shows the estimates of the derivatives $\partial f/\partial\lambda$ obtained from Eq. (12). The scattering of data in Fig. 5 is caused by the added white noise. Again, we have used CUBGCV to fit the data by a smoothed cubic spline fit. The solid line in Fig. 5(b) displays the smoothed spline fit of $\partial f/\partial\lambda$.

The direct application of the above formulas requires random access to initial conditions, which is very difficult and time consuming in an experiment. However, we now present a new method of learning these quantities on-the-fly by appropriate manipulations of a running experiment. The dynamic rate of map variations, $\partial f^1/\partial\lambda$, can be learned experimentally by the method shown schematically in Fig. 6. The top part of Fig. 6 shows the flow rate, and the bottom part shows the concentration of the catalyst Z. We wait for two iterates at the nominal value $\lambda_0=3.50\times10^{-4}\ \mathrm{s}^{-1}$, between experimental parameter variations, to make sure that the transients settle onto the 1-D attractor, to a high degree of accuracy. The observed next piercing of the Poincaré surface, after a parameter variation, is the dynamics response $f^1(x)$. Each two-iterate time interval of fixed $\lambda(t)=\lambda_0$ is followed by a one ''iterate'' time interval of $\lambda(t)=\lambda_0\pm\epsilon$, where we choose the small variation $\epsilon=2\times10^{-6}\ \mathrm{s}^{-1}$. The response function $Z(t)$ must be read accordingly. For example, the second maximum of $Z(t)$, denoted by an empty

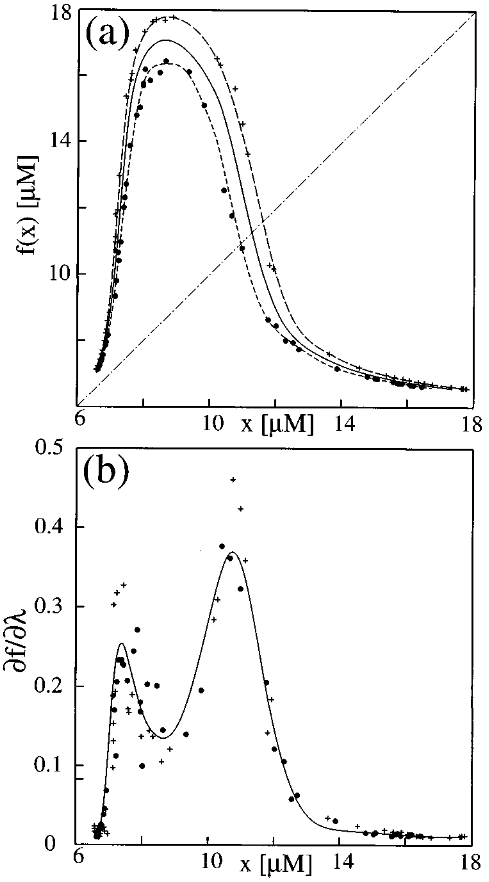

FIG. 5. The one-dimensional maps constructed for the constant value of control parameter λ (a) and estimates of the derivatives $\partial f\ /\partial\lambda$ (b). Each data sets contain 50 successive Poincaré sections. Crosses - $\lambda=3.55\times10^{-4}\ \mathrm{s}^{-1}$; solid circles - $\lambda=3.45\times10^{-4}\ \mathrm{s}^{-1}$. The standard deviation of the added noise is $\sigma_\xi=0.001$. Dashed lines represent smoothed cubic splines for the data sets; the solid line in (a) illustrates smoothed map from Fig. 1; the solid line in (b) is a smoothed cubic spline fit of all data points.

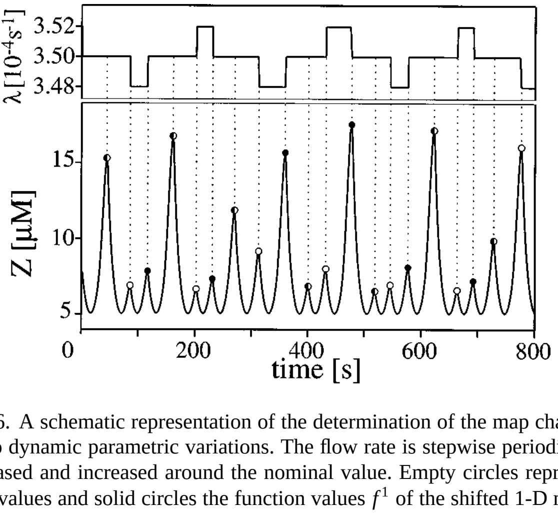

FIG. 6. A schematic representation of the determination of the map changes due to dynamic parametric variations. The flow rate is stepwise periodically decreased and increased around the nominal value. Empty circles represent the x values and solid circles the function values f^1 of the shifted 1-D maps.

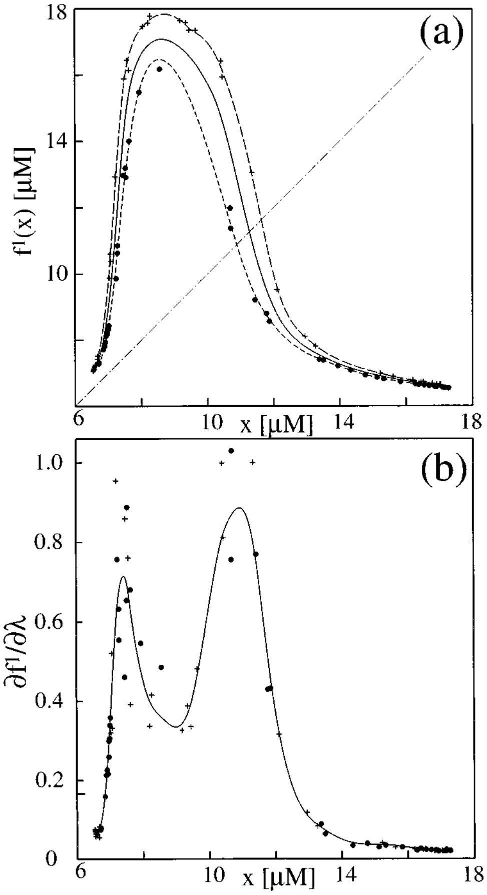

FIG. 7. The one-dimensional maps constructed from the small dynamic variations of the control parameter (a) and estimates of the derivatives $\partial f^1/\partial k$ (b). Crosses - λ is increased from the nominal value λ_0 to 3.52 $\times 10^{-4}$ s^{-1}; solid circles -λ is decreased from the nominal value λ_0 to 3.48×10^{-4} s^{-1}. There are 40 data points for each data set. Noise level and lines notation are the same as in Fig. 5.

circle in Fig. 6, represents the x coordinate. Immediately following, we set $\lambda(t)=\lambda_0-\epsilon$ and therefore the following maximum is the $f^1(x)$-coordinate (solid circle) of the perturbed 1-D map. After sampling this dynamic response value $f^1(x)$, we reset the flow rate back to the nominal value for two reference events. This procedure can be repeated, or a perturbation with the same amplitude but of opposite sign can be applied, as shown in Fig. 6. By periodically repeating this process with positive and negative perturbations, we learn dynamic responses $f^1_{\lambda_0\pm\epsilon}(x)$ for values of x ergodically scattered throughout the interval. Only the points, depicted as empty circles (x values) and solid circles (f^1 values), are used to construct the perturbed 1-D maps, $f^1_{\lambda_0\pm\epsilon}(x)$.

Figure 7(a) shows the maps constructed from data generated by the method depicted in Fig. 6. One set of data points (squares) represents the map when the flow rate was increased and the other set of data points represents the map for decreased flow rate. The solid line illustrates the cubic smoothed spline fit of the fixed flow rate at the nominal value. The derivative $\partial f^1/\partial\lambda$ is estimated similarly to the derivative $\partial f/\partial\lambda$, from the difference quotient:

TABLE II. The encoded message.

Character	ASCII Decimal	Binary
C	67	1000011
h	104	1101000
a	97	1100001
o	111	1101111
s	115	1110011

$$\frac{\partial f^1}{\partial\lambda}\bigg|_{(x,\lambda_0)}=\frac{f^1_{\lambda_0+\epsilon}(x)-f_{\lambda_0}(x)}{\epsilon}, \tag{13}$$

where $\epsilon=\pm 2\times 10^{-6}$ s^{-1}. The solid circles in Fig. 7(b) depict the estimated derivative for decreasing flow rate and the crosses for increasing flow rate. Both data sets represent coinciding dependences and we use all of the data points to calculate the cubic smoothed spline fit, which is shown as the solid line in Fig. 7(b).

These technical considerations are generally necessary to any experimental control problem in which the dynamics and the control responses are learned only through an observed time-series. The experimentally controlled encoding technique requires: the cubic smoothed spline fit of map from Fig. 1, the fitted derivatives $\partial f/\partial\lambda$ from Fig. 5, and the fit of $\partial f^1/\partial\lambda$ from Fig. 7. The grammar derived from the map together with the smoothed fits allows us to efficiently control the symbol sequences of the chaotic oscillations using small parameter variations.

V. ENCODING IN THE PRESENCE OF NOISE

In next two sections we describe how to encode the message ‘‘Chaos,’’ in 7-bit ASCII binary form shown in Table II, bit by bit, into the chaotic dynamics of the BZ reaction, and we evaluate the success of our ‘‘arbitrary’’ message encoded in the presence of increasing noise amplitudes. We also discuss techniques to improve noise resistance. We have used several different lengths of the binary code representation of phase points (ranging from 4 bits through 8 bits) with the conclusion that in the presence of noise, more than 6-bits do not improve reliability. Therefore we employ the 6-bit code representations in these simulations. Thus there is a five iterate delay between controlling a message bit into the chaotic oscillations, and the observability of the bit in the most significant position in terms of the position of $Z(t)$ relative to the decision point d. In this specific case, just as for the 4-bit grammar, the codes with three zeros in a row are also forbidden for the 6-bit grammar, and all other codes are allowed.[19] Once the grammar has been determined, we can predetermine the desired values $f_{\text{des}}(x)$, for each possible code as the centers of the like-coded regions (see the solid triangles in Fig. 4). Throughout the experiments these centers of like-coded regions are used as the targets for the control mechanism.

Encoding starts at an arbitrary initial condition x_0, to which we assign its corresponding 6-bit code $a_0.a_1a_2a_3a_4a_5$, where a_i is either a ‘‘0’’ or a ‘‘1.’’ For example, assume that at least one of the bits a_4 and a_5 is a ‘‘1.’’ Therefore the first bit of the message from Table II, $m_1 = 1$, is controlled into the least significant position by targeting the appropriate bin center $f_{\mathrm{des}}(x)$. After the decimal character is shifted one bit to the right (the Bernoulli-shift map) and the most significant bit is forgotten, a new target code m_1 is inserted, forcing the code $a_1.a_2a_3a_4a_5m_1$. The center of this like-coded region, or bin, defines the desired value $f_{\mathrm{des}}(x_0)$ used in Eq. (11) to estimate the required perturbation. The perturbation applied to the system is expected to bring the next intersection on the Poincaré surface x_1 near the desired value $f_{\mathrm{des}}(x_0)$. Therefore the code assigned to x_1 should be identical to the targeted code $a_1.a_2a_3a_4a_5m_1$.

Due to modeling errors, internal control errors and noise, and external channel noise, there can be mistakes in the transmitted bits. When and where a bit error occurs, and how we deal with such errors, affects whether a message error occurs. As shown in Figs. 2 and 3 some codes are assigned to a very narrow range of x and a few ‘‘misfits’’ may land in an unintended region, corresponding to another code. When the targeted and the actual code differ in other then in the most significant bit, the difference can be corrected during the next control step by using the proper target code $a_2.a_3a_4a_5m_1m_2$. That is, the solution is to target the intended second iterate as if no error occurred, and if successful, the previous error will be eliminated. The penalty is that a slightly larger than usual control signal (larger than the radius of the largest bin) may be necessary to overcome the noise. Correction works in any case other than with errors which cause a most significant digit difference between the actual and the target code. Most significant bit errors represent message errors.

As mentioned above, the grammar of the attractor forbids more than two zeros in a row; to use only small control signals we must respect this grammar. So if both bits of the initial code a_4 and a_5 are ‘‘0’’ bits, then a nonbearing bit $b_n = 1$ must be transmitted, between message bits. The message bit m_1 is then encoded in the second controlled iteration, targeting the code $a_2.a_3a_4a_5b_nm_1$. Any time there is a sequence of two ‘‘0’’ bits in the message, a nonbearing $b_n = 1$ bit must transmitted, and then removed by the receiver to decode the message. The grammar can be additionally restricted to introduce a degree of noise-resistance.

Avoiding the region near the symbol partition d decreases the probability that white noise can push the signal across the partition causing a most significant digit error. The region around $x = d$, and its preiterates, can be avoided by designing an appropriate further restriction on the grammar, thus introducing a ‘‘noise-resistant gap’’.[20] The idea here is that all regions of phase space correspond to a digital code in the symbol space, and so we avoid the region near the symbol partition by never transmitting the code corresponding to that region. For example, we note that regions near the symbol partition have the 6-bit codes ‘‘0.10011’’ on the left, and ‘‘1.10011’’ on the right. So if we simply never transmit the codes ‘‘0.10011’’ and ‘‘1.10011,’’[11] then these neighborhoods of $x = d$ will never be visited, nor will any preiterates of that region (and so we will actually be controlling orbits on a subset of the chaotic attractor which is a Cantor set nonattracting chaotic saddle). We use buffer bits to respect the attractor's grammar of the forbidden sequence ‘‘000,’’ and we also use buffer bits for our designed grammar to avoid transmitting ‘‘10011’’ to introduce a degree of noise resistance. That is, in this example, both transmitter and receiver must know that ‘‘100’’ will be followed by a buffer bit ‘‘1’’ to avoid three zeros and yielding ‘‘1001,’’ which will be followed by a buffer ‘‘0,’’ yielding ‘‘10010’’ to avoid the noise resistant gap. The noise resistance in communicating with chaos has been studied by Bollt *et al.*[20] who addressed two points: (1) the noise resistance comes at the cost of slightly slowing transmission rate of the message, due to the extra time required to transmit extra buffer bits, (2) more (or less) noise resistance can be designed into an appropriately designed grammar, but the trade-off is that increasing noise resistance decreases the channel capacity or transmission rate of the chaotic oscillator.

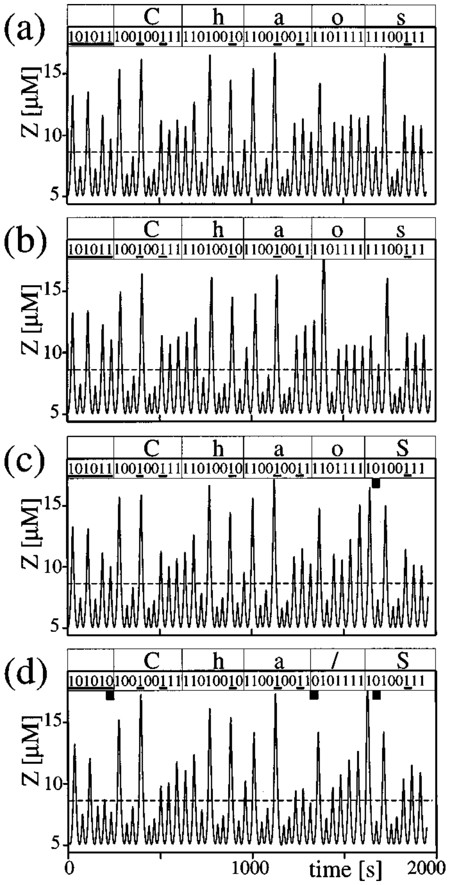

FIG. 8. Oscillations of the Belousov-Zhabotinsky reaction controlled to contain the message ‘‘Chaos’’ in 7-bit ASCII characters. The dashed line indicates the decision point d. A maximum below the line represents a ‘‘0’’ bit and above corresponds to a ‘‘1’’ bit. The bits are displayed at the top of each figure with the decoded message. The encoding technique obeys the grammar - after every two ‘‘0’’s in a row a mandatory nonbearing bit ‘‘1’’ (underlined) is encoded. (a) Errorless encoding with the added white noise with standard deviation $\sigma_\xi = 0.001$; (b) $\sigma_\xi = 0.002$ - encoding without error; (c) $\sigma_\xi = 0.002$ - encoding with one error. (d) $\sigma_i = 0.004$ - encoding with multiple errors. The error bits are emphasized by solid squares.

Figure 8 shows several examples of encoding of the word ‘‘Chaos’’ in 7-bit binary ASCII characters, in the presence of noise according to Eq. (3). The dashed line indicates the symbol decision value—all oscillatory maxima of $Z(t)$

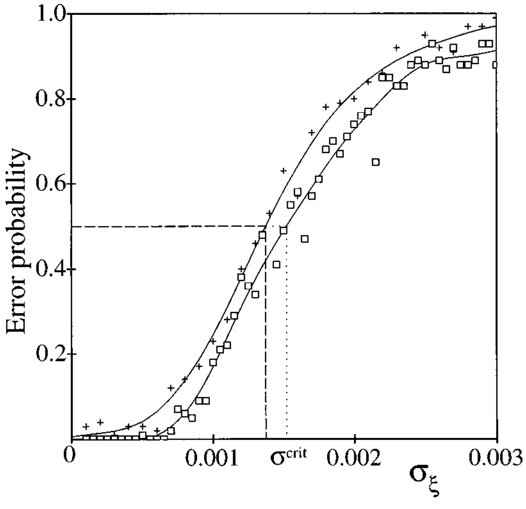

FIG. 9. Probability of an error occurrence when encoding the message ''Chaos'' in 7-bit binary ASCII form. The error probability is estimated for every fixed value of the noise standard deviation σ_ξ from 100 trials using pseudorandom initial conditions. There are two sets of trials. The top curve (''+'' signs) tests the grammar of the attractor, with forbidden three zeros in a row. The bottom improved curve (''□'' signs) includes also the grammar restriction with forbidden code sequence ''10011'' to introduce a noise-resistant gap. σ^{crit} designates the critical value of the standard deviation corresponding to an error probability of 0.5.

which are below the dashed line are read as a ''0'' bit and maxima above the dashed line are read as a ''1'' bit. The message is decoded by removing the first six random bits a_i and all the nonbearing bits b_n to respect the grammar. The nonbearing bits are underlined in Fig. 8.

Figure 8(a) shows an example of errorless encoding for a standard deviation of $\sigma_\xi = 0.001$. When $\sigma_\xi = 0.002$, the encoding is often errorless as in Fig. 8(b), but errors are possible, as demonstrated by the case of the one error shown in Fig. 8(c). For $\sigma_\xi = 0.004$, in Fig. 8(d), errors are more frequent; encoding the word ''Chaos'' includes more than one incorrectly encoded bit. We used identical initial conditions in all the examples shown in Fig. 8, but the sequence of pseudorandom numbers is different for every example.

VI. DISCUSSION

The reliability of the encoding method can be tested in the presence of noise by repeatedly encoding the 7-bit ASCII binary form of the message ''Chaos,'' using pseudorandom initial conditions. Figure 9 shows probability of an error occurrence as a function of the noise standard deviation σ_ξ. There are two curves depicted: the upper curve, fitting the crosses, shows transmission trials without designing a noise resistant gap, and the lower curve, fitting the squares, shows trials with noise resistant gap by never transmitting ''10011'' (as well as the forbidden sequence of the attractor ''000''). The probability of an error occurrence is estimated from the relative number of trials in which at least one error in the encoded message occurs. The trials depicted in Fig. 9 give very good estimates of the experimental noise level requirements for successful encoding. For $\sigma_\xi > 0.003$ at least one error occurs essentially in every trial, for $\sigma_\xi < 0.0005$ there is very low probability of an error in encoding. We choose as successful encoding the encoding where the error probability is lower than 0.5. The dashed line in Fig. 9 designates the critical standard deviation σ^{crit} with an error probability equal to 0.5. The conclusion from the lower curve in Fig. 9 is that by avoiding the sequence ''10011'' of the noise gap around the symbol partition, one can improve the encoding reliability. The dotted line shows an increased σ^{crit} by using the noise resistant gap, and we can see that for $\sigma_\xi < 0.0007$, there are essentially no observed errors.

To further increase reliability, there are modifications to the encoding algorithms which can be introduced. As we have already mentioned, one can increase the size of the noise resistant gap around the symbol partition by using an appropriately designed code, but at the cost of transmission rate. Also, the recent work of Rosa *et al.*[21] has outlined a method to filter a noisy signal generated by an *a priori known* one-dimensional model by using hyperbolicity to diminish error through inverse iteration along a ''probabilistically most likely path.'' In summary, their noise cleaning method makes use of two main ideas: (1) (Noise) errors which grow in unstable directions on iteration, shrink on inverse-iteration. (2) There is no well defined inverse of a one-dimensional-hump map, which is often two-to-one; there are typically two branches to choose between. They resolve this issue by choosing the branch contained within confidence intervals; only one of the two alternative preiterates of the Gaussian distribution, around a particular iterate, is statistically reasonable with a high degree of confidence. Their iterative method has shown outstanding error correcting abilities, but it does not apply directly to our problem, in which we will learn the 1-D map from cubic spline interpolation of experimental data. Thus there will be an unavoidable modeling error, so we will not have a clean model against which to clean the message bearing signal. A possible avenue of improvement might be to adapt the filtering technique suggested by Rosa *et al.* to our scenario by designing a ''predictor-corrector'' type algorithm, to generate a cleaner model.

Another method which may increase the reliability of our encoding technique is *tuning*. According to this method, the centers of the codes are first preset as targeted desired values $f_{\text{des}}(x)$. Then, thorough testing of the encoding method may reveal codes sensitive to experimental noise. For such *sensitive* codes we can estimate new, more noise resistant targets of the desired values $f_{\text{des}}(x)$. The new targets can be further tuned by another statistical testing. We have not pursued these possibilities at this time as our current approach promises to be experimentally successful while still maintaining simplicity.

In this work, we lay groundwork for experimental control of chemical chaos, with the intention to encode an arbitrary message into its oscillations. To successfully encode information into chaotic oscillations of the BZ reaction we need to suppress the experimental noise below the critical noise level. The primary question is, can it be done? If ''yes,'' how can the noise be confined below critical levels? We claim that the experiments to control chaos in the BZ reaction can be successfully accomplished and that the experimental noise level can be kept below the critical value of

0.0015. This requires an experimental setup which maintains the flow rate, stirring rate and temperature fluctuations at a minimum. We can achieve this by replacing peristaltic pumps by piston pumps, and by usage of a precisely controlled synchronous motor for mixing. The method also requires precise temperature control of both stock solutions and the reaction mixture inside the reactor. There is an experimental evidence that these precautions can reduce the fluctuations in the BZ reaction in a well mixed system to around 0.1% ($\sigma_\xi = 0.001$).[16] This is below our estimated critical standard deviation value and therefore our claim, that an experimental encoding can be successful, has historical support.

A technical note of importance concerns a remark by Nosticzius *et al.*[22] who pointed out that the chaotic oscillations of the BZ reaction strongly depend on the purity of the input reagents, mainly on the iron contents of the malonic acid. Therefore malonic acid should be purified to obtain reproducible results with different samples of malonic acids.

VII. CONCLUSION

We have demonstrated, by numerical simulations, that an arbitrary message can be encoded into chaotic chemical oscillations in the presence of white noise. The standard deviation of the experimental noise directly affects the number of errors which occur during encoding. This trade-off can be improved by introducing a noise-resistant gap. When encoding contains only a very small number of errors, then it may still be possible to guess the correct message. But a large number of errors will strongly modify the message. This suggests that even though the encoding technique will work well in the presence of experimental noise, it is necessary to confine it to an acceptable level.

In the present study we have demonstrated that information can be encoded in the dynamical behavior of a chemical oscillating system by forcing the system to follow a desired trajectory, in a numerical chemical reaction model. This result is not intended to change the classical means of human communication. But an analogy of oscillating chemical systems to biological systems suggests that living systems could use the information bearing capabilities inherent in chaotic dynamics. Finally, it is our hope that the techniques developed here, in the noisy setting of a chemical reaction, will prove useful to experimentalists working on controlling symbol dynamics in other chaotic media such as optical lasers, and electronic circuits, which could serve as chaotic communication devices.

ACKNOWLEDGMENTS

This work was supported by the National Science Foundation Grant No. CHE-9615834 to M. D., and by the Army Research Labs and by the National Science Foundation Grant No. DMS-9704639 to E.M.B.

[1] J. D. Murray, *Mathematical Biology* (Springer-Verlag, New York, 1990).

[2] E. Bollt and M. Dolnik, Phys. Rev. E **64**, 1196 (1997).

[3] E. Bollt and M. Dolnik, ''Learning the grammar of a chaotic dynamical system in a format suitable for controlling symbol dynamics and communications,'' Proc. 1st. International Conference on Controlling Chaotic Oscillations, St. Petersburg, Russia (IEEE, Piscataway, NJ, USA, 1997), pp. 573–577.

[4] K. M. Cuomo and A. V. Oppenheim, Phys. Rev. Lett. **71**, 65 (1993).

[5] L. Kocarev and U. Parlitz, Phys. Rev. Lett. **74**, 5028 (1995).

[6] U. Parlitz, L. O. Chua, L. Kocarev, K. S. Halle, and A. Shang, Int. J. Bifurcation Chaos **2**, 973 (1992).

[7] U. Parlitz, L. Kocarev, T. Stojanovski, and H. Preckel, Phys. Rev. E **53**, 4351 (1996).

[8] K. Murali and M. Lakshmanan, Phys. Rev. E **48**, R1624 (1993).

[9] C. Zhou and T. Chen, Phys. Lett. A **225**, 60 (1997).

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

[11] S. Hayes, C. Grebogi, E. Ott, and A. Mark, Phys. Rev. Lett. **73**, 1781 (1994).

[12] L. Györgyi and R. J. Field, Nature (London) **355**, 808 (1992).

[13] F. W. Schneider and A. F. Münster, J. Phys. Chem. **95**, 2130 (1991).

[14] V. Petrov, V. Gaspar, J. Masere, and K. Showalter, Nature (London) **361**, 240 (1993).

[15] J. C. Roux, R. H. Simoyi, and H. L. Swinney, Physica D **8**, 257 (1983).

[16] K. G. Coffman, W. D. McCormick, Z. Noszticzius, R. H. Simoyi, and H. Swinney, J. Chem. Phys. **86**, 119 (1987).

[17] M. F. Hutchinson, ACM Transactions on Mathematical Software **12**, 150 (1986).

[18] W. H. Press, B. P. Flannery, S. A. Teukolovsky, and W. T. Vetterling, *Numerical Recipes* (Cambridge University Press, Cambridge, England, 1986).

[19] In general, this is not always true; we may expect new n-bit rules to emerge as we improve the approximation of the grammar by increasing n.

[20] E. Bollt, Y.-C. Lai, and C. Grebogi, Phys. Rev. Lett. **79**, 3787 (1997).

[21] E. Rosa, S. Hayes, and C. Grebogi, Phys. Rev. Lett. **78**, 1247 (1997).

[22] Z. Noszticzius, W. D. McCormick, and H. L. Swinney, J. Phys. Chem. **91**, 5129 (1987).