Experimental Control of Chaos for Communication

Scott Hayes,[1,*] Celso Grebogi,[2,†,‡] Edward Ott,[2,*,†,§] and Andrea Mark[1]
[1]*U.S. Army Research Laboratory, Adelphi, Maryland 20783*
[2]*University of Maryland, College Park, Maryland 20742*
(Received 6 May 1994)

The use of chaos to transmit information is demonstrated experimentally. The symbolic dynamics of a chaotic electrical oscillator is controlled to carry a prescribed message by use of extremely small perturbing current pulses.

PACS numbers: 05.45.+b

It has been argued on theoretical grounds that it is possible to use the close connection between chaos and information theory for communication [1]. The realization that chaos can be controlled by small perturbations [2] leads to the idea that chaotic systems can be caused to produce a signal bearing desired information. In this Letter, we report the first *experimental* verification that communicating with chaos is indeed feasible, and that one can utilize the naturally occurring chaotic orbits on the attractor to carry information. We demonstrate experimentally that we can make the symbolic dynamics of a chaotic electrical oscillator follow a prescribed symbol sequence; thus we can encode and transmit any desired message [3].

To understand the ideas behind our experiment, consider a simple electrical oscillator working in a nonlinear regime and producing a large-amplitude chaotic signal consisting of a seemingly random sequence of peaks. Now imagine that one can use small perturbing current pulses to cause the peaks to fall above or below a preassigned threshold value in a desired order. The peaks that fall above the threshold are assigned the binary symbol 1, and those that fall below are assigned a 0. By controlling this *symbol sequence,* we can encode a desired message in the signal. The message can thus be transmitted through a signal transmission path, and a receiver can extract the message by observing the sequence of peaks relative to the threshold amplitude. From the practical point of view, we envision that the power oscillator that generates the information-bearing signal can be simple and efficient, while the small controlling current pulses can be produced by a low-power microelectronic circuit.

Figure 1(a) is a schematic diagram of the electrical circuit used in our experiment [4]. The nonlinearity comes from a nonlinear negative resistance represented by the voltage v_R. The negative resistance i-v characteristic g is shown in Fig. 1(b). The frequency of oscillation is approximately 5 kHz. The capacitor C_1 is variable so that we can tune the circuit for various types of behavior. For this experiment, we tuned the circuit to produce a Rössler band. An uncontrolled experimental trajectory is shown in Fig. 2.

The first step toward controlling symbol sequences is to experimentally obtain a description of the symbolic dynamics of the uncontrolled oscillator. Because in practice some symbol sequences are not produced by the free-running oscillator, the rules specifying the allowed symbol sequences (the *grammar*) need to be determined. Methods for determining the grammar of chaotic systems are the topic of much current research in symbolic dynamics [5]. We first discuss our method for determining the symbolic dynamics and allowed symbol sequences. To determine the symbolic dynamics and to control the system, we use a computer-assisted measurement system that rapidly samples the circuit voltages and can perform all the computations necessary for control. The system can also provide current pulses to control the system. We will henceforth refer to this system as the *controller* [6].

To characterize the dynamics, we take samples of v_{C_1} on the Poincaré surface $v_{C_2} = 0$ as v_{C_2} passes through zero volts in a negative direction. The intersection of the

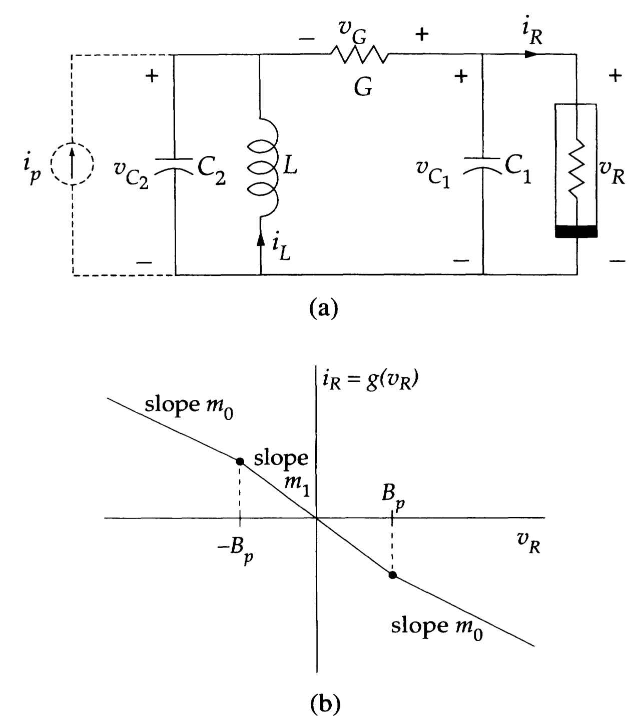

FIG. 1. Double scroll oscillator: (a) Electrical circuit with $L = 8.2$ mH, $C_1 = 0.0055$ μF, $C_2 = 0.05$ μF, and $1/G = 1.33$ kΩ. Controlled current pulses are injected by the source i_p. (b) Negative resistance i-v characteristic.

0031-9007/94/73(13)/1781(4)$06.00

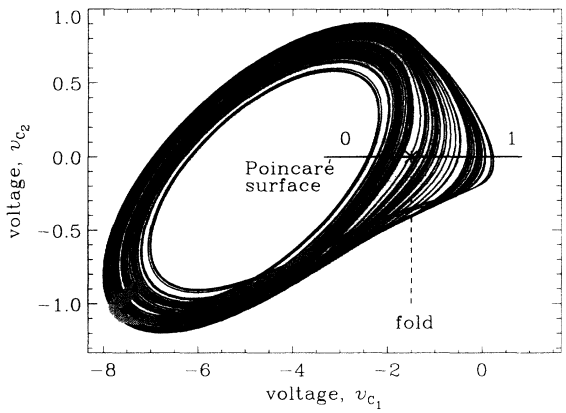

FIG. 2. Measured trajectory on Rössler-type attractor showing Poincaré surface, symbolic partition, and locus of control pulses. Units are volts.

attractor with the surface $v_{C_2} = 0$ is well approximated by a single thin arc, and thus we can describe the dynamics of v_{C_1} on the surface of section by one coordinate. Each time the voltage v_{C_2} (see Fig. 2) passes through zero in the negative direction, the controller is triggered (using the trigger circuit in an oscilloscope) to sample the value of v_{C_1}. We call the value of v_{C_1} *on the Poincaré surface the state point* $x = v_{C_1}|_{v_{C_2}=0}$. The natural way to partition the Poincaré surface into symbolic regions is to choose the partition boundary at the point on the surface of section which leads into the fold in the Rössler band under the action of the dynamics. (If the boundary is not taken at the fold, then two different state points can generate the same symbol sequence.) The symbol 0 is thus generated when the state point is to the left of the boundary shown in Fig. 2, and a 1 is generated when the state point is to the right of the boundary. The symbolic partition is shown on the Poincaré surface with respect to the state point that passes into the fold in Fig. 2. Hence, the voltages $x = v_{C_1}|_{v_{C_2}=0}$ that fall above the boundary correspond to a 1 and the voltages that fall below the boundary to a 0, and the dividing threshold is the partition boundary at the point that leads into the fold in Fig. 2. [In practice, if the transmitted signal is the scalar load voltage $v_G(t)$, one can detect the symbols by observing the *peaks* in this signal alone relative to a threshold voltage.]

Whenever the system state coordinate crosses the Poincaré surface, the state point $x = v_{C_1}|_{v_{C_2}=0}$ is stored in the controller. The controller then records the sequence of symbols generated after the crossing by determining whether the next ten crossings (including the present crossing) of the Poincaré surface are to the left or right of the partition boundary in Fig. 2. The ten-bit symbol sequence stored in the symbol register can be viewed as a binary number from 0 (all 0's) to 1023 (all 1's). We denote this number by r. We define an inverse coding function [1] $s(r)$ for r ranging from 0 to 1023, where $s(r)$ is the average value of the state point producing the symbol sequence specified by r. In practice, $s(r)$ is computed by following a long orbit and taking a running average of all x that produced the symbol sequence r. If the symbol sequence 1111111111 is produced by a trajectory passing through the state point $x = -1.5$ V, for example, we set the variable $s(1023) = -1.5$, where 1023 is the binary value of 1111111111. When another trajectory produces the same symbol sequence, perhaps at a voltage $x = -1.502$ V, we update the value to $s(1023) = -1.501$, the average of the state points for the two trajectories that produced this symbol sequence.

An experimental inverse coding function is shown in Fig. 3. This finite approximation of the inverse coding function represents a table with $2^{10} = 1024$ entries stored in controller memory of the state points s that correspond to each possible ten-bit symbol sequence r. Some of the $2^{10} = 1024$ ten-bit sequences are never produced by the free-running oscillator. This is typical of the symbolic dynamics of chaotic systems in general and is known as a constraint on the *grammar*, or, in the context of communication, it is known as a *channel constraint.* The type of constraint that occurs for our oscillator is a generalized type of *run-length constraint* [7]. (A simple run-length constraint places a maximum and minimum on the allowed length of strings of 1's and the allowed length of strings of 0's appearing in the code.)

To be able to control the system and hence produce a signal carrying a desired message by use of small perturbations, we must incorporate the effect of perturbations on our description of the dynamics. We coupled a digital-to-analog converter in our controller to the positive terminal of capacitor C_2 shown in Fig. 1(a) in series with a large buffer resistor, thus providing a simple method for injecting small current pulses. The current pulse generator is represented schematically as dashed lines in Fig. 1. These pulses had an approximate duration of 4 μs as compared to the typical cycle time of the chaotic oscillator of about 200 μs.

In our experiment we applied controlling current pulses 80 μs after each crossing of the surface of section. The

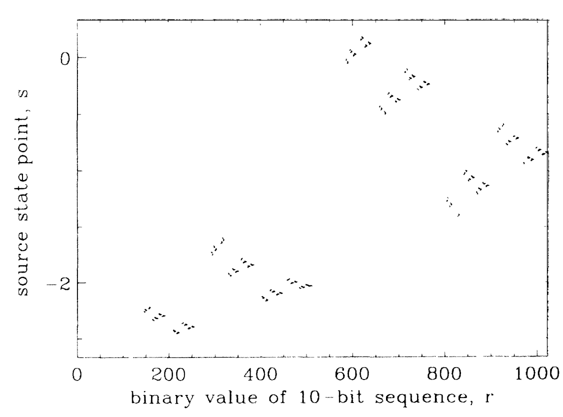

FIG. 3. Binary inverse coding function. The binary number corresponding to each binary sequence is given by its base-10 value.

distance around the attractor at which a pulse occurs varies depending on the average velocity of the phase space flow along a given orbit. The location on the attractor at which pulses occur is indicated in Fig. 2 by the gray swath at the bottom of the attractor. The key to our control procedure is the realization that the application of a pulse affects a very small change in the *state point* that produces a *given* symbol sequence. Because the pulse amplitudes are very small, this change Δx is approximately linear in the pulse amplitude p. That is, $p = d\Delta x$, where in general the proportionality constant d depends on x, or, equivalently, on r. We calculate $d(r)$ as follows. After a state point x is sampled, the pulse is applied using a small reference value $p = p_r$, and ten symbols (including the one corresponding to x) are shifted into the symbol register in the same way as before. Another reference pulse is not applied until this ten-bit sequence has been accumulated. The procedure is then repeated. In this way, we obtain the functional relationship between x and r in the presence of the reference pulse of amplitude $p = p_r$. We denote this relationship by $w(r)$, where w is the value of x that produces symbol sequence r in the presence of the reference perturbation. The proportionality constant $d(r)$ is then given by $d(r) = p_r/[w(r) - s(r)]$. We can thus compute the *required* pulse amplitude to affect a *desired* small change in the value of the state point by $p = d(r)[x - s(r)]$, where x is the actual state point upon crossing the Poincaré surface and r is the *desired* future symbol sequence, which will be specified by the message bits. We thus compute the required pulse amplitude using only one subtract and one multiply, a highly efficient algorithm that could be used in a *microelectronic controller* at much higher speeds than in the experiment described here.

We now discuss the procedure used to cause the oscillator to track a prescribed binary symbol stream. We now have, in controller memory, the function $s(r)$ containing the state points that will produce each allowed symbol sequence if no perturbation is applied and a function $d(r)$ that we use to compute the required control pulse amplitude. We now switch the controller from the *learn* phase to the *control* phase. When the system coordinate passes through the Poincaré surface, the controller samples v_{C_1} and obtains the first state point x_0. The controller then loads the symbol register with the symbol sequence that will naturally evolve if no control perturbation is applied; that is, the value in the symbol register is set to the value of r corresponding to the array element $s(r) = x_0$. The symbol register now contains ten bits that are determined not by our message, but by the state point that happens to occur the first time. While the trajectory travels toward its next encounter with the Poincaré surface, the controller shifts the symbol register left and inserts the first message bit into the rightmost (least significant bit) slot. The leftmost (most significant) bit is discarded; this bit corresponds to the symbol that was just produced and is no longer needed. The symbol sequence now appearing in the symbol register is the one that we *want* the oscillator to produce after the upcoming encounter with the Poincaré surface. Corresponding to this *desired* symbol sequence, r_1 is the *desired* state point $x_1 = s(r_1)$ that will produce this sequence. It is unlikely, however, that the state point x_1 will exactly correspond to $s(r_1)$, so the controller must apply a perturbation to correct the trajectory. When the trajectory passes through the Poincaré surface again, the error in the state point $\Delta x_1 = x_1 - s(r_1)$ is computed. The controller then applies the first control perturbation $p_1 = \Delta x_1 d(r_1)$. The controller then shifts the symbol register left and places the next message bit into the rightmost slot, discarding the leftmost bit corresponding to the symbol just produced. As each successive state point x_n becomes available, the controller looks up the target state point $s(r_n)$, where r_n is the *desired* symbol sequence in the symbol register, computes the error $\Delta x_n = x_n - s(r_n)$, applies a control pulse $p_n = \Delta x_n d(r_n)$, and shifts one new message bit into the symbol register. This procedure is repeated continuously as long as the system is under control. Because we first load the symbol register with the symbol sequence that will naturally evolve without control, the first ten iterates of the control procedure target the system from chaos into the information transmission.

As previously mentioned, the constraint on the grammar for our oscillator is a *generalized* run-length constraint, but we can satisfy the grammar with a *simple* run-length constraint such that no runs of more than one 0 occur. The grammar can be satisfied with this simple constraint because the dynamics of the Rössler system is well approximated by a one-dimensional single-hump map [7]. We have encoded the message [8], "Yea, verily, I say unto you: A man must have chaos yet within him to birth a dancing star. I say unto you: You have yet chaos in you." We used the standard seven-bit ASCII computer standard for the characters in the message and then mapped the message bits to code bits according to

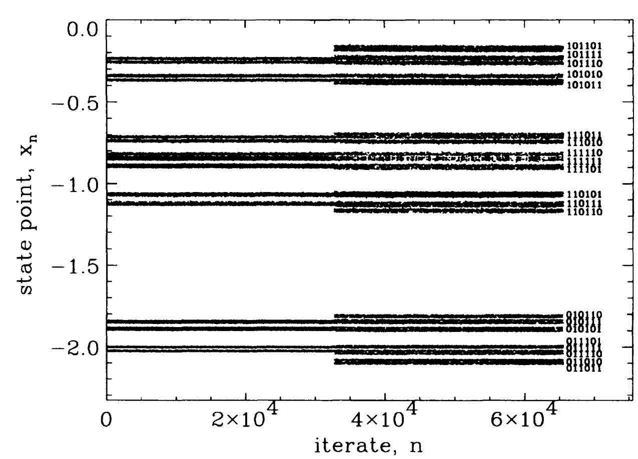

FIG. 4. State-point sequence with encoded message showing the Cantor-set structure produced by the constrained code during control with two different encoded messages.

the rule $0 \to 01$, $1 \to 11$, and thus no more than one 0 will ever occur in a row. (One could also use the variable-length code $0 \to 01$, $1 \to 1$ to improve the rate of information transmission.) To demonstrate the effect of the control symbol sequence on the signal, we have also produced another "message"—a random sequence of 0's and 1's constrained to no more than one 0 in a row. A state point sequence corresponding to the repeated [9] transmission of the Nietzsche quotation followed (starting at cycle $n \cong 3.3 \times 10^4$) by transmission of the random bit stream is shown in Fig. 4. Because of the overly restrictive simple run-length constraint, the state points x_n fall within bands on the Poincaré surface during transmission of both messages. (The rms control current during the whole sequence was 0.2 μA; compare this to circuit currents of a few milliamps.) Both binary streams cause the sequence of state points to approximately lie on a Cantor set [10]. This occurs because an overly constrained *symbol* sequence also constrains the *state-point* sequence. Because the binary sequence representing the quotation is *more* restrictive than the random bit sequence (because the code $0 \to 01$, $1 \to 11$ used to satisfy the run-length constraint is itself more restrictive than necessary), the signal is confined to *narrower* bands during the quotation than during the random bit sequence.

An important aspect of this type of signal is that more than one information bit can be extracted from a single state-point sample—the Cantor-set structure of the signal consists of bands that represent *sequences* of symbols not just single encoded binary symbols. One can visually resolve bands corresponding to up to six-bit-long binary sequences in our experimental signal; we have labeled these sequences in Fig. 4. Only one *new* code bit becomes available for each cycle of the system, but more than one code bit can be extracted from one state point sample. A signal such as this could thus be detected and decoded by only sampling once every three cycles of the signal, for example, which would give the observer three code bits per sampled state point.

In conclusion, the experimental results reported support the feasibility of communicating with chaos by controlling the symbolic dynamics of a chaotic oscillator and demonstrate some important features of this method of communication.

This research was supported by the U.S. Army Research Laboratory and the U.S. Department of Energy (Office of Scientific Computing, Office of Energy Research).

*Department of Physics and Astronomy, University of Maryland, College Park MD.
†Laboratory for Plasma Research.
‡Institute for Physical Science and Technology and Department of Mathematics.
§Department of Electrical Engineering and Institute for Systems Research.

[1] S. Hayes, C. Grebogi, and E. Ott, Phys. Rev. Lett. **70**, 3031 (1993); S. Hayes and C. Grebogi, Mater. Res. Soc. Symp. Proc. **2038**, 153 (1993).

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

[3] A different use of chaos in communication is the use of chaos synchronism for the purpose of secure (i.e., difficult to decipher by an interceptor) transmission of information, see L. M. Pecora and T. L. Carroll, Phys. Rev. Lett. **64**, 821 (1990); K. M. Cuomo and A. V. Oppenheim, Phys. Rev. Lett. **71**, 65 (1993).

[4] T. Matsumoto, Proc. IEEE **75**, 1033 (1987). For further information about chaos in electronic circuits, see, for example, P. S. Lindsay, Phys. Rev. Lett. **47**, 1349 (1981); C. Jeffries and J. Perez, Phys. Rev. A **27**, 601 (1983); S. K. Brorson, D. Dewey, P. S. Linsay, *ibid.* **28**, 1201 (1983); H. Izeki, J. S. deGrasse, T. H. Jensen, *ibid.* **28**, 1207 (1983); R. W. Rollins and E. R. Hunt, *ibid.* **29**, 3327 (1984).

[5] P. Cvitanović, G. Gunaratne, and I. Procaccia, Phys. Rev. A **38**, 1503 (1988); P. Grassberger, H. Kantz, and U. Moenig, J. Phys. A **22**, 5217 (1989); D. P. Lathrop and E. J. Kostelich, Phys. Rev. A **40**, 4028 (1989); L. Flepp, R. Holzner, E. Brun, M. Finardi, and R. Badii, Phys. Rev. Lett. **67**, 2244 (1991); G. B. Mindlin, H. G. Solari, M. A. Natiello, R. Gilmore, and X.-J. Hou, J. Nonlinear Sci. **1**, 147 (1991).

[6] The controller is an HP3567A signal analyzer containing a high-speed signal processing computer and is used to digitize the incoming signal, compute the control algorithm, and provide a control pulse within one cycle of the oscillator.

[7] An introductory discussion of codes for constrained communication channels can be found in R. E. Blahut, *Principles and Practice of Information Theory* (Addison-Wesley, Reading, MA, 1988). The general constraint on symbol sequences in systems described by single-hump maps will be discussed in a longer paper (to be published).

[8] Friedrich Nietzsche, *Thus Spake Zarathustra*. We have this liberal yet appealing translation of the German "Ich sage euch: man muß noch Chaos in sich haben, um einen tanzenden Stern gebären zu können. Ich sage euch: ihr habt noch Chaos in euch." from the frontispiece of the book by Manfred Schroeder, *Fractals, Chaos, Power Laws—Minutes from an Infinite Paradise* (W. H. Freeman and Company, New York, 1991). The more literal translation by Thomas Common (Random House Modern Library, New York) is "I tell you: one must still have chaos in one, to give birth to a dancing star. I tell you: ye have still chaos in you."

[9] For the first 3.3×10^4 cycles, the oscillator repeats the transmission of the Nietzsche quotation. There are 133 characters in the quotation, thus 7×133 ASCII bits, and $7 \times 133 \times 2 = 1862$ code bits. The encoded quotation thus appears in the signal about 17 times during the first half of the controlled state-point sequence.

[10] This signal represents the first experimental stabilization of a Cantor-set invariant density.

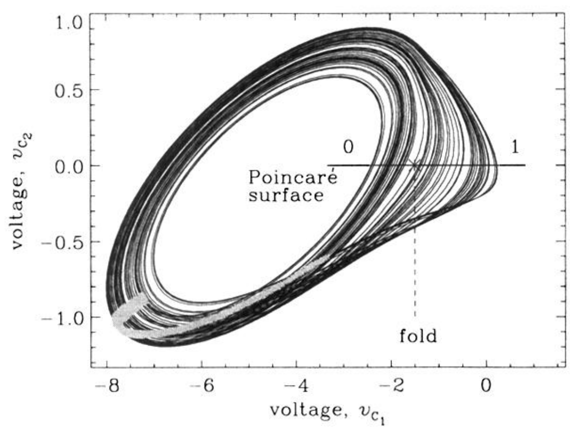

FIG. 2. Measured trajectory on Rössler-type attractor showing Poincaré surface, symbolic partition, and locus of control pulses. Units are volts.

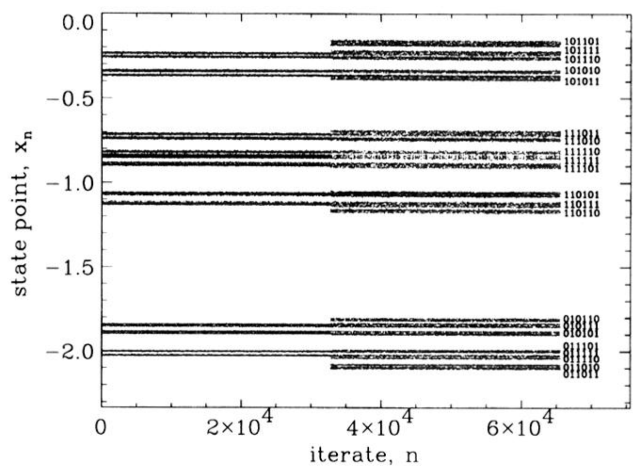

FIG. 4. State-point sequence with encoded message showing the Cantor-set structure produced by the constrained code during control with two different encoded messages.