PARTIAL CONTROL OF TRANSIENT CHAOS IN ELECTRONIC CIRCUITS

ALEXANDRE WAGEMAKERS*, SAMUEL ZAMBRANO and
MIGUEL A. F. SANJUÁN
Nonlinear Dynamics, Chaos and Complex Systems Group,
Departamento de Física, Universidad Rey Juan Carlos,
Tulipán s/n, 28933 Móstoles, Madrid, Spain
*alexandre.wagemakers@urjc.es

We present an analog circuit implementation of the novel partial control method, that is able to sustain chaotic transient dynamics. The electronic circuit simulates the dynamics of the one-dimensional slope-three tent map, for which the trajectories diverge to infinity for nearly all the initial conditions after behaving chaotically for a while. This is due to the existence of a nonattractive chaotic set: a chaotic saddle. The partial control allows one to keep the trajectories close to the chaotic saddle, even if the control applied is smaller than the effect of the applied noise, introduced into the system. Furthermore, we also show here that similar results can be implemented on a circuit that simulates a horseshoe-like map, which is a simple extension of the previous one. This encouraging result validates the theory and opens new perspectives for the application of this technique to systems with higher dimensions and continuous time dynamics.

Keywords: Partial control of chaos; analog circuit; transient chaos.

1. Introduction

Although permanent chaos has been thoroughly studied, much less attention has been devoted to the phenomenon of transient chaos [Tél, 1991], which is also a rather common feature in nonlinear dynamical systems. For example, if we have a system with a chaotic attractor, by varying one of the system's parameters, a bifurcation like a boundary crisis [Grebogi *et al.*, 1982] can make it lose its attractive nature. The nonattractive set resulting from this process, typically a zero-measure set with a chaotic dynamics, is called a *chaotic saddle.*

The influence of this chaotic set in the dynamics is typically the appearance of the phenomenon of transient chaos. What happens is that a trajectory which typically passes close to the chaotic saddle, will behave chaotically for a while, after which it will settle into a coexisting (and possibly periodic) attractor. The universality of this type of dynamics

has been underlined by a recent work where the transient chaos can be shown to be formally related with Poincaré recurrences [Altmann & Tél, 2008], shedding thus some light on this important aspect of dynamical systems theory and chaotic dynamics.

As pointed out in [Dhamala & Lai, 1999], transient chaos appears in models related with species extinction, voltage collapse on an electrical power system or undesired bursts on a chemical reaction. Controlling chaos is another interesting concept in nonlinear dynamics. Many different control techniques have been used in order to control permanent chaos, though other control schemes have also been proposed in recent years in order to control transient chaos. By controlling transient chaos it is generally meant to perturb the dynamical system considered in order to keep the trajectories close to the nonattractive chaotic set, far from those undesired attractors. These control schemes, like

the paradigmatic OGY control scheme [Ott *et al.*, 1990], aim to achieve this goal by applying small but accurately chosen perturbations to the system. Some of them aim at the stabilization of the system on any of the periodic orbits embedded in the chaotic saddle [Tél, 1991; Place & Arrowsmith, 2000a, 2000b] or to keep trajectories always close to orbits with long-lived chaotic transients [Dhamala & Lai, 1999; Kapitaniak & Brindley, 1998]. For a description of nonattractive chaotic saddles, how to compute them and their corresponding fractal properties, see [Aguirre *et al.*, 2009].

A major issue in this context is the presence of noise. It is easy to see that if the class of systems described above are under the influence of environmental noise, the process by which the trajectories escape the chaotic saddle and settle into the vicinity of the coexisting attractor will take place faster. This undoubtedly poses a problem from a controlling point of view.

This problem can be stated in a more precise way as follows. We might assume that the dynamics of the system are given by the m-dimensional map $p_{n+1} = f(p_n)$. We suppose that there is a region Q in phase space that encloses the chaotic saddle, and from which all trajectories escape after a finite number of iterations. We consider that there is some noise u_n acting on the system, in such a way that the dynamics become $p_{n+1} = f(p_n) + u_n$. The noise u_n is assumed to be bounded by $0 < |u_n| \leq u_0$, where $|\cdot|$ is the norm of the m-dimensional space considered.

We are interested in keeping trajectories inside Q. In order to achieve this goal, we consider that at each iteration we can apply an adequate control r_n correcting the system's trajectory. This can be expressed mathematically as follows:

$$\begin{cases} q_{n+1} = f(p_n) + u_n \\ p_{n+1} = q_{n+1} + r_n(q_{n+1}), \end{cases} \tag{1}$$

where the control we apply is also bounded, $|r_n| \leq r_0$.

With all these ingredients in mind, we can now consider the possible answers to the following important question: which will be the amplitude r_0 of the control needed to keep the trajectories inside Q? Its value will typically depend on u_0, and one might expect that given the nonattractive nature of the saddle, the fact that nearly all trajectories escape from Q under iterations and the presence of noise, it would be necessary to have $r_0 > u_0$.

However, this is not necessarily true. Recently it has been shown that there is a technique that allows one to keep trajectories close to the saddle even if $r_0 < u_0$: *partial control of chaos*, that is, even if the control applied is smaller than the applied noise. This technique is called *partial* because it allows one to keep trajectories inside Q with $r_0 < u_0$, although it does not determine where the trajectory will *exactly* go inside Q. As we will show shortly, this is possible if the chaotic saddle is a consequence of certain geometrical actions that are quite general. Its first implementation was with the slope-three tent map [Aguirre *et al.*, 2004], the simplest one-dimensional system with a nonattractive chaotic set, and it was extended later to a family of one-dimensional maps with transient chaotic dynamics [Zambrano & Sanjuán, 2008]. Some later research work [Zambrano *et al.*, 2008] showed to be also suitable for the important class of maps with a Smale horseshoe [Smale, 1967], maps that are supposed to arise anytime that we have a transverse homoclinic intersection on a dynamical system. This idea was later generalized and thoroughly analyzed in [Zambrano & Sanjuán, 2009].

However, up to now there are only numerical implementations of this technique. The aim of this paper is to present an electronic circuit implementation of the partial control technique. The simple fact of simulating the equations in a circuit is a solid argument for the robustness of the method. Furthermore, it is a first step toward the possible application to continuous dynamical system in higher dimensions. We present here the first implementation of this technique in an electronic device based on discrete dynamics.

The structure of the paper is as follows: in Sec. 2 we present the system that we will deal with and the control technique. In Sec. 3, we explain how the experimental implementation was performed. In Sec. 4 the experimental results are provided, and in Sec. 5 we explain how an analogous result can be obtained in a circuit simulating a linearly expansive horseshoe-like map. Finally in Sec. 6 we draw the main conclusions of this work.

2. The System and the Partial Control Technique

The system on which we mainly focus in this paper is a paradigmatic example of dynamical system with transient chaos and was already described in
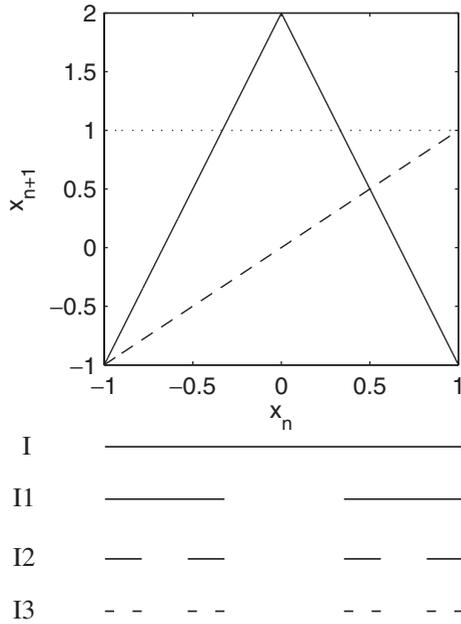
Fig. 1. Inductive construction of the chaotic saddle for the slope-three tent map. The set $I_\infty = \lim_{k \to \infty} I_k$ is a Cantor set, that is, a fractal.

[Aguirre *et al.*, 2004]: The slope-three tent map

$$x_{n+1} = T(x_n) = 3(1 - |x_n|) - 1. \qquad (2)$$

The graph of the map is sketched in Fig. 1. The dynamics of this map is particularly interesting inside the interval $I = [-1, 1]$. Notice that any trajectory starting out of I will never come back to it and diverge very fast to $-\infty$. In order to understand the type of dynamics arising in this system, we can define the intervals I_k as the intervals such that points inside it are mapped outside I after more than k iterations. They are depicted in Fig. 1. It can be easily observed that the length of those intervals decreases exponentially with k. In the limit as $k \to \infty$, we find the set of trajectories that remain bounded forever in the interval, the zero-measure set I_∞, that is the classical Cantor set. The dynamics inside this nonattractive set are chaotic (the set is dense in periodic orbits and it has sensitive dependence on the initial conditions): it is the chaotic saddle for this system.

Moreover, for a typical orbit, if the initial condition is located in the interval I, the numerical value of x_n will escape from I and then diverge to $-\infty$. The number of iterations needed to escape from I will depend on how close the initial condition was to the chaotic saddle. The chaos-like behavior that trajectories have before escaping from

the interval I are what we have called previously transient chaos.

As we said above the purpose of the control technique is to maintain the trajectory in an infinite chaotic transient inside the interval I. However, we will assume that, as stated in Eq. (1), the system is affected by noise u_n bounded by u_0, and that at each iteration we will use a control r_n bounded by r_0 to keep trajectories inside it, so that the equations of the system read

$$\begin{cases} q_{n+1} = T(x_n) + u_n \\ x_{n+1} = q_{n+1} + r_n(q_{n+1}). \end{cases} \qquad (3)$$

Now we can explain how the partial control technique allows one to keep the trajectory inside the interval I even if $r_0 < u_0$. The key idea in order to achieve this goal is to keep the trajectories inside a given *safe set* S^k of a family of safe sets $\{S^j\}$, $j = 0, 1, \dots$ that we describe below. In other words, the control must be such that $x_n \in S^k$ for all n. The geometrical properties of this set allow to keep trajectories inside I with $r_0 < u_0$. Now, we will describe the structure of these safe sets.

The safe sets can be defined as the set of preimages of $x = 0$, which are $S^k \equiv T^{-k}(0)$, so obviously their structure depends basically on the form of the map. We take set S^0 as the point $x = 0$. It can be shown that the safe set S^k is formed by 2^k *safe points* of the form

$$\pm \frac{2}{3} \pm \frac{2}{3^2} \pm \cdots \pm \frac{2}{3^k} \qquad (4)$$

for $k \geq 1$, so the 2^k possible combinations of $+$ and $-$ in the equation above correspond to the 2^k points in the set S^k. A pictorial representation of the safe sets S^0, S^1, S^2 and S^3 is shown in Fig. 2.
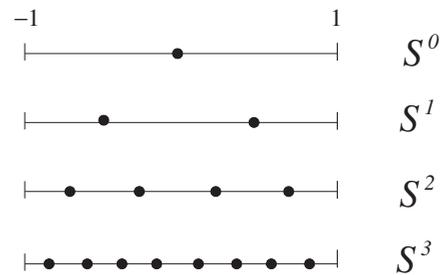
Fig. 2. A pictorial representation of the safe sets S^0, S^1, S^2 and S^3 in the interval $I = [-1, 1]$ for the slope-three tent map. Each point in S^k has two points of S^{k+1}, one to its left and another to its right, that is, closer to it than any other point of S^k.

For a better and a detailed analysis of the structure of these safe sets, it is important to emphasize first that the set S^{k+1} is the preimage of S^k, in other words all points in S^{k+1} are mapped into S^k under the map $x_{n+1} = T(x_n)$. From Eq. (4), it is easy to see that two consecutive points of S^k are separated by a distance equal to $\frac{4}{3^k}$. The second and most important property of these sets, that can also be understood from Eq. (4) and visualized in Fig. 2, is that each point in S^k has two preimages of S^{k+1}, one to its left and another one to its right, closer to it than any other point in S^k, at a distance $\delta^{k+1} = \frac{2}{3^{k+1}}$. The value of δ^k obviously decreases to zero with k.

Considering these geometrical properties, the partial control strategy is then the following. Given the maximum value of the noise u_0, assume that the nth iterate is on a point of the set S^k, where k satisfies $\delta^k < u_0$. The action of the map will take it to a point $T(x_n) \in S^{k-1}$. Noise is also present and will displace the trajectory to a point $q_{n+1} = T(x_n)+u_n$. However, we have noticed that any point in S^{k-1} is surrounded by two points on S^k, one to its left and another to its right at a distance $\delta^k < u_0$. Thus, in all possible cases, it is possible to see that a correction r_n such that $|r_n| \leq r_0 < u_0$ can put the resulting iterate $x_{n+1} = q_{n+1} + r_n(q_{n+1})$ back on S^k, so that the same process can be repeated for x_{n+1}, and this can be repeated *ad infinitum*. This method will be detailed in Sec. 3 and illustrated in Sec. 4.

In order to make the above strategy work, it is clear that the considered trajectory should start at S^k. As in other problems where the aim is to sustain an unstable state, starting somehow close to that unstable state is necessary. However, it can be proved that the technique has a remarkable rate of success. In other words, that it can keep trajectories bounded with $r_0 < u_0$, if the trajectories start sufficiently close to the chaotic saddle.

In the next section we show an experimental implementation of this control technique and some experimental realizations that illustrate clearly all these ideas.

3. Analog Circuit Implementation

Our aim is to build a circuit reproducing the dynamics described by Eq. (3). We must emphasize that the circuits do naturally have noise. However, due to the fact that the tipical voltages in the circuit are much higher than the electronic

Table 1. List of intervals where the safe sets lie.

Interval Where q_n Lies	Safe Point Associated
$-\infty < q_n \leq -\frac{2}{3}$	$-\frac{2}{3} - \frac{2}{3^2}$
$-\frac{2}{3} < q_n \leq 0$	$-\frac{2}{3} + \frac{2}{3^2}$
$0 < q_n \leq \frac{2}{3}$	$\frac{2}{3} - \frac{2}{3^2}$
$\frac{2}{3} < q_n < +\infty$	$\frac{2}{3} + \frac{2}{3^2}$

noise intensity, the signal to noise ratio can exceed 30 dB. Consequently, we opt here to introduce the noise externally. This will allow an analysis of the performance of the control technique.

We are going to use a noise amplitude u_0 so that the adequate safe set will be S^2, that has four safe points. Furthermore, we know that using our control strategy, depending on the value of q_{n+1} we have to steer the trajectory to a different safe point, following the criteria described in Table 1, that gives a systematic way to find the closest safe point.

With these safe points (this safe set), the trajectory can be maintained in a infinite chaotic transient with $r_0 < u_0$ if $\frac{2}{3^2} < u_0$. After each iteration (including the effect of the noise) the system is
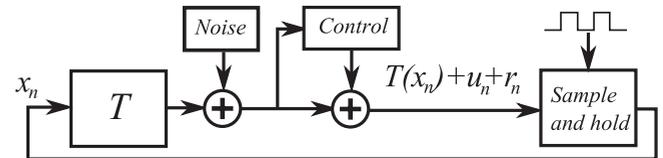
Fig. 3. Block diagram of the circuit. This schematic view of the algorithm shows the relation and the order between each block. The electronic circuit is constructed according to this representation.
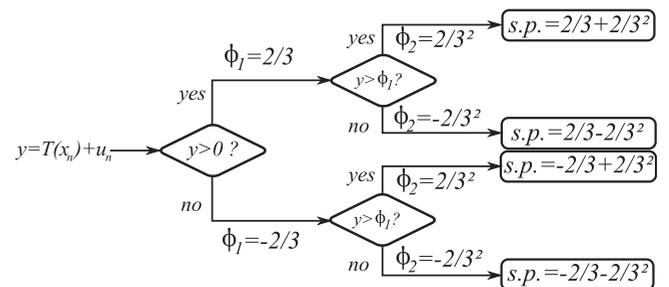
Fig. 4. State flow diagram of the implementation of the control method with the circuit, where "s.p." means safe point.
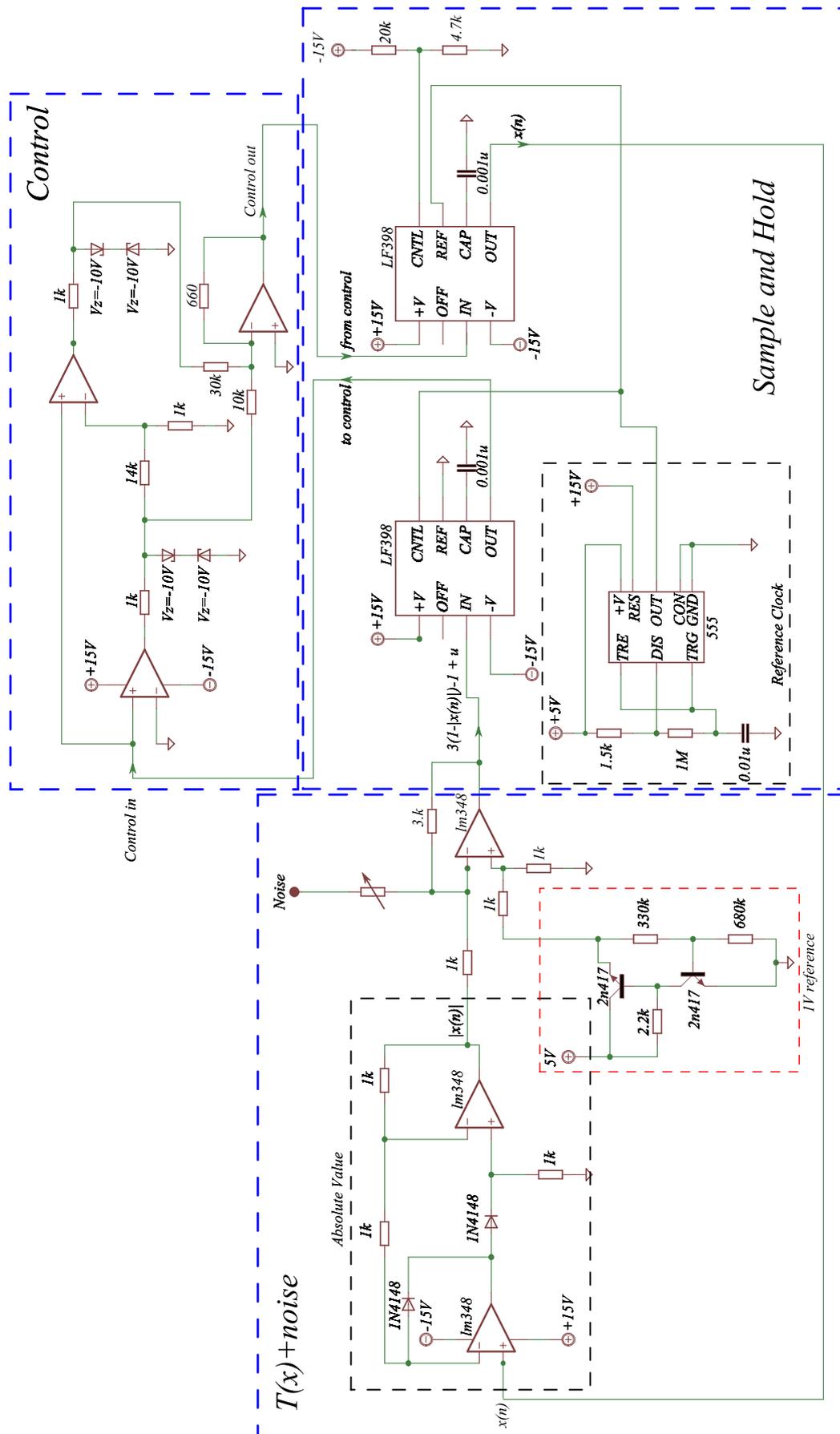
Fig. 5. Schematic representation of the electronic circuit of the controlled unstable tent map.

1250032-5

driven back to the closest safe point according to Table 1.

The block diagram in Fig. 3 represents the system with functional blocks. The implementation of the electronic circuit is based on this sketch. The first block on the left represents the function T with the input x_n on the left side and the output $T(x_n)$ on the right side. In the second block, the noise is summed up to $T(x_n)$ and subsequently the control is applied in the third block. The last block on the right represents the delay unit which implements the recurrence $x_{n+1} = T(x_n) + u_n + r_n$.

The circuit works as follows:

(1) First we consider the value of x_n at the entry of the function T. The value $T(x_n)$ is obtained at the output of the block almost immediately.
(2) The noise is added to $T(x_n)$.
(3) The control is computed and summed up. At this point, we have $T(x_n) + u_n + r_n$. This numerical value is available at the input of the sample and hold block.
(4) The sample and hold block takes at its input $T(x_n) + u_n + r_n$ and holds it until the next clock ticks. The block releases it at the next clock tick so that we obtain at the output $x_{n+1} = T(x_n) + u_n + r_n$ and the cycle repeats.

The electronic circuit is a modification of the logistic map circuit proposed in [Hellen, 2004]. The schematic diagram of the circuit is shown in Fig. 5, where each functional block is delimited by dashed lines.

The tent map block function is based on a simple nonlinear circuit with operational amplifiers (op-amps) and diodes which reproduces the shape of the function. This first block computes the absolute value function with op-amps and diodes so that we obtain $|x_k|$. Then the signal is scaled and we sum up an offset such that the output is $T(x_n)$. The noise is generated with an Agilent function generator and summed up to $T(x_n)$ with an op-amp.

The control block is based on a series of op-amps used as comparators to locate the closest safe point. With two comparators, we can locate four safe points. The number of comparators establishes the number of safe points that can be stored, so that 2^n safe points (and thus the safe set S^n) can be used with n comparators.

The implementation of the algorithm appears in the state flow diagram of Fig. 4. The algorithm operates as follows:

(1) First the input $y_n = T(x_n) + u_n$ is compared to zero such that the result of the comparison is $\phi_1 = \frac{2}{3}$ V when $y_n > 0$ and $\phi_1 = -\frac{2}{3}$ V when $y \leq 0$.
(2) The second comparator takes the input y_n and compares it to ϕ_1 (the result of the previous comparison). The result of this comparison, which we will call ϕ_2 is $\phi_2 = +\frac{2}{3^2}$ V when $y > \phi_1$, else the result is $\phi_2 = -\frac{2}{3^2}$V.
(3) The last op-amp sums up the results of the two comparisons ϕ_1 and ϕ_2 which gives us the safe point.

The control that we will apply to the trajectory is $r_n = \phi_1 + \phi_2 - y_n$ so that the corrected trajectory is on the safe point $\phi_1 + \phi_2$. The control circuit is implemented with op-amps set as comparators. Zener diode clamps regulate the voltage stability for the comparisons. A simple op-amp set as adder sums up the results of the comparisons.

The sample and hold block is based on the LF398 sample and hold circuit. The two sample and hold LF398 circuits act as a delay unit. While the circuit is releasing the value x_n at the output, it samples the value $T(x_n) + u_n + r_n$ at the input in order to release this value at the next cycle. At the next clock tick the numerical value $T(x_n) + u_n + r_n$ becomes x_{n+1} at the output.

We need two additional blocks to make the circuit work properly. The first block is a voltage reference of 1 V necessary to compute the function T. This block is made up of a couple of bipolar transistors and some resistors that set a constant voltage of 1 V at the output of the block. The second block is a clock that sets the pace of the iterations of the circuit. The implementation of the clock is based on a LM555 integrated in a circuit set in multivibrator mode.

4. Experimental Results

In this section, we present some results obtained with the electronic circuit. In Fig. 6 the time series of the iteration of the system without control is shown. For a particular initial condition, the time series tends to -5 V, which is the saturation voltage for this system. It means that the voltage cannot increase or decrease any further than $+5$ V or -5 V. The escape from the interval $[-1, 1]$ V is exactly what could be expected for this type of map.

Once the control is switched on, the trajectory of the system is maintained on a stable chaotic
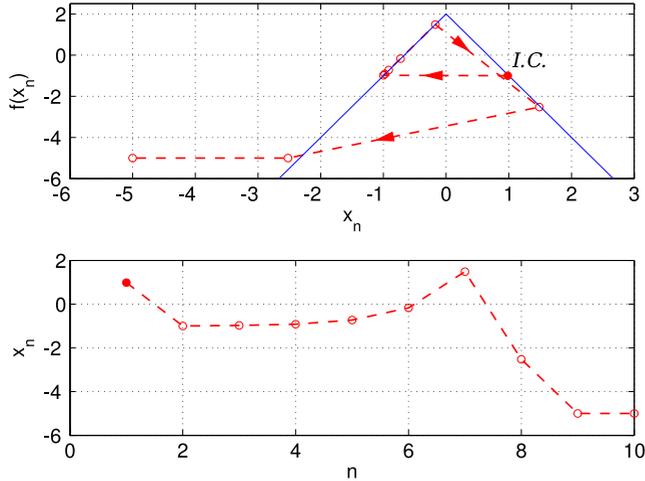
Fig. 6. Time series without control. The trajectory of the time series tends rapidly after a few iterations to $-5\,\text{V}$ which is the saturation voltage of the circuit. The initial condition is displayed as I.C. in the figure and is marked with a red dot. The order of the iterations are marked with an arrow.

transient trajectory. An example of such time series is shown in Fig. 7. We have chosen a value of $u_0 > \frac{2}{3^2}$, and we have considered the initial condition of a point on S^2. The map should take it to a point $T(x_n)$, that is plotted with blue dots. However noise displaces it to $T(x_n) + u_n$ (iterates with noise) which are plotted with red dots. In spite of this, we can see how for each iteration, the trajectories are driven to the closest point on the set
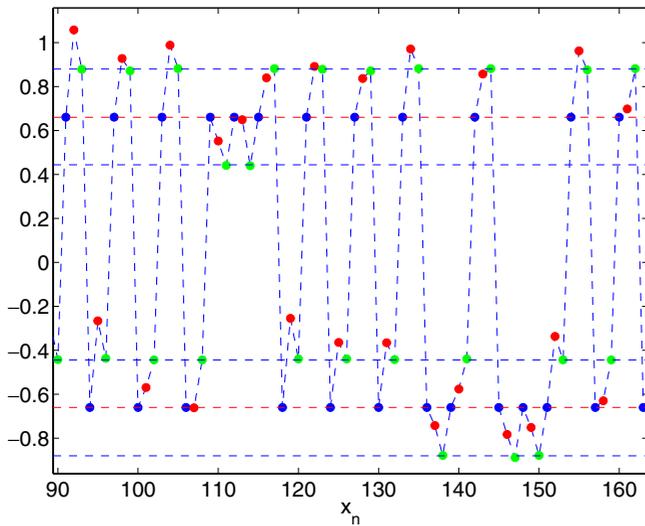
Fig. 7. Time series of the controlled system. Blue dots: iterates, red dots: iterates with noise, green dots: controlled trajectory. If we follow the green dots, we can observe that the trajectory is maintained on the safe points and that the correction needed for this is always smaller than the maximum value of noise, u_0.
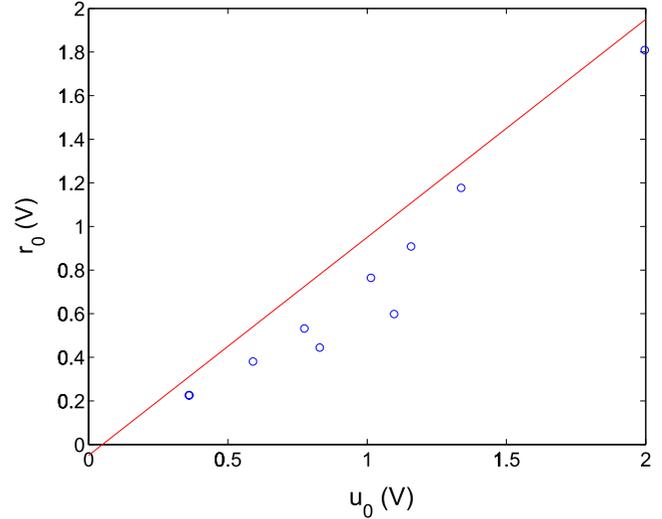
Fig. 8. The figure shows measurements of the maximum noise u_0 against the maximum control r_0. Each point represents the maximum of a time series with different noise standard deviation. The solid line represents the unit line $u_0 = r_0$. All the points below this line satisfy $r_0 < u_0$.

S^2 with a correction smaller than the maximum deviation induced by noise. The controlled iterates $x_{n+1} = T(x_n) + u_n + r_n$, are displayed in green in that figure. This allows to obtain a stabilized trajectory with $r_0 < u_0$.

Provided that $u_0 > \delta^2$, the theoretical results predict that the maximum value of the control r_0 should be lower than the maximum value of the noise u_0. Figure 8 represents the measurements made with the circuit of the maximum noise value against the maximum control applied during the trajectory. Each point in Fig. 8 was obtained with a time series with different noise standard deviation. All the points shown lie under the solid line $r = 0$, so they meet the condition $r_0 \leq u_0$, confirming thus the result mentioned above.

5. Control of a Horseshoe-Like Map

With the circuit shown before, we can extend our work easily to a simple but important type of two-dimensional map: a linear Smale horseshoe-like map [Smale, 1967]. A good description of the dynamics for this type of maps can be found in [Alligood *et al.*, 1996; Yang, 2009]. Furthermore, recent works have shown the existence of horseshoes in hyperchaotic circuits [Yang *et al.*, 2007]. This way, we can achieve experimental confirmation of the results obtained in [Zambrano *et al.*, 2008; Zambrano & Sanjuán, 2009]. The typical Smale horseshoe map, $p_{n+1} = h(p_n)$,
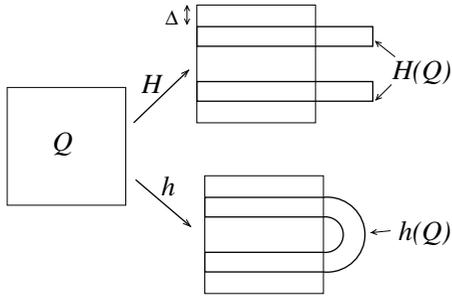
Fig. 9. Geometrical action on a square Q of a typical horseshoe map $p_{n+1} = h(p_n)$ and the horseshoe-like map $p_{n+1} = H(p_n)$ that we consider here. The first one stretches and folds back into itself Q. The second one stretches, cuts and pastes it back into itself. The action of both maps differs just on the mapping of the middle third of the square.

stretches linearly the square and then folds it back into itself, as shown in Fig. 9. This implies that nearly all the trajectories escape from Q under iterations and that there is a nonattractive chaotic set inside Q, where the dynamics is chaotic. A good description of the dynamics for this type of maps can be found in [Alligood *et al.*, 1996]. The geometrical action of the horseshoe map can be observed in Fig. 9.

The horseshoe-like map that we describe here is a map $p_{n+1} = H(p_n)$, where $p_n \equiv (x_n, y_n)$. The equations of the map are the following

$$x_{n+1} = T(x_n) = 3(1 - |x_n|) - 1 \qquad (5)$$

$$y_{n+1} = P(y_n) = \begin{cases} \dfrac{1}{6}y_n + 1 - \Delta & \text{if } x_n \leq 0 \\[2mm] \dfrac{1}{6}y_n - 1 + \Delta & \text{if } x_n > 0 \end{cases}, \qquad (6)$$

with $\Delta = 0.5$. The geometrical action of the map on Q is shown in Fig. 9. Note that the geometrical action of this map in the square $Q \equiv [-1, 1] \times [-1, 1]$ is identical to that of the horseshoe map, except for the middle third ($x \in (-1/3, 1/3)$ of the interval). Of course, it fulfills the general conditions for the application of the partial control technique given in [Zambrano & Sanjuán, 2009].

This system will be particularly easy to implement because we notice that the equation for the x-variable is identical to the one-dimensional map described in the previous section. The second equation takes x as an input. The term Δ gives the distance between the top and bottom sides of Q and $H(Q)$, as shown in Fig. 9.

Moreover, using some of the ideas given for the slope-three tent map, we can sketch the main

characteristics of this map. First, we see that nearly all trajectories inside Q will escape from it under some iterations: only a zero-measure Cantor set of vertical segments will remain inside it forever.

If we add a small amount of noise to the system, this escaping process will be accelerated. As we did before, we assume that the system will be affected by noise u_n bounded by u_0 and a control perturbation r_n bounded by r_0 is applied on each iteration, so that the global dynamics is given by the equations

$$\begin{cases} q_{n+1} = H(p_n) + u_n \\ p_{n+1} = q_{n+1} + r_n(q_{n+1}). \end{cases} \qquad (7)$$

The aim here is to keep the trajectories inside the square Q with $r_0 < u_0$. As with the one-dimensional map, the partial control strategy needed to achieve this goal will consist of maintaining the trajectories on a given safe set S^k.

Considering the analogies between this map and the slope-three tent map, it is not surprising that their safe sets share lots of features. The set S^0 is defined as the vertical segment inside Q placed in the position $x = 0$. The safe sets $\{S^k\}$ are then the preimages inside Q of S^0. It can be verified that the safe set S^k consists of 2^k vertical segments inside Q whose position along the x-axis is equal to the x-value of the safe points of the map $x_{n+1} = T(x_n)$ described above. They are depicted in Fig. 10.

The geometrical properties are then evident: first, by definition each point on S^k is mapped into a point in S^{k-1} under H. Second, each segment of S^k is surrounded by two segments of S^{k+1}, one to its left and another to its right, and the distance to each of them is again $\delta^{k+1} = \frac{2}{3^{k+1}}$.
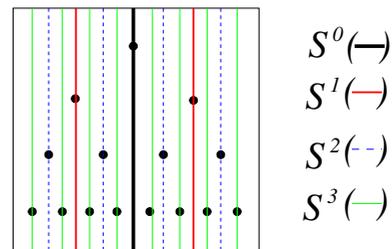
Fig. 10. A pictorial representation of the safe sets S^0, S^1, S^2 and S^3 in the square Q for the horseshoe-like map. The x-coordinate of each segment of each safe set corresponds to the x-value of the safe points of the safe sets for the tent map, which are represented for the sake of clarity. Note that each segment in S^k has segments of S^{k+1}, one to its left and another to its right, that is, closer to it than any other point of S^k.

The strategy that we can follow to keep trajectories bounded is the following. Considering by simplicity that $u_0 < \Delta$, what we have to do is to consider the initial condition on S^k such that $u_0 > \delta^k$. Each iteration will be mapped to a point on S^{k-1} and then deviated by the noise. However, due to the fact that each segment in S^{k-1} is surrounded by two segments of S^k, we can steer the trajectory onto the closest segment of S^k with $r_0 < u_0$ for each iteration [Zambrano *et al.*, 2008].

From an experimental point of view, we notice that the second equation remains stable as long as x_n is driven on a stable trajectory. If we control x_n, the value of y_n is also controlled, which means that we will apply the control only on the first variable. In other words, for this system the only variable physically relevant is the x coordinate.

Figure 11 represents the block diagram of the system. The upper block is identical to the diagram of the one-dimensional system while the inferior part implements the iterations of y_n. The function block P takes as input y_n and x_n to produce the iterates $P(y_n)$. The electronic implementation is similar to Fig. 5 with an additional block for the iterates of y_n. The function P comprises a comparator and op-amps. Since the design is simple and similar to Fig. 5, we do not reproduce the diagram here.

In order to test our results, we use a value of u_0 such that the adequate safe set is S^2, that consists of four vertical segments. In Fig. 12 we show the controlled trajectory of the two dimensional system. In this figure the iterates are plotted with blue dots, the iterates with noise with red dots and the controlled value is displayed with green dots. We notice that no matter how noisy the perturbation

(a)

(b)

Fig. 12. (a) Time series of the two-dimensional system with control. Blue dots: iterates, red dots: iterates with noise, green dots: controlled trajectory. If we follow the green dots, we can observe that the trajectory is maintained always on the safe set S^2 and the correction needed is always smaller than the amplitude of the applied noise u_0. (b) Picture of the oscilloscope with x_n against y_n. We observe how the noise spreads the trajectory around the safe points.

is, the correction needed to take the trajectory back into the set S^2 is always smaller than the maximum value of noise u_0.

In Fig. 13 the time series of the control in absolute value of the variable x_n is shown. The dashed line in the figure represents the maximum absolute value of the noise in the system. The control of the linear horseshoe map remains lower than the maximum as predicted by the theory. In the same figure, the noise trajectory is shown in light color,

Fig. 11. Block diagram of the 2D system. The diagram is similar to the one-dimensional system but with an additional block that implements the function P and samples the variable y_n.
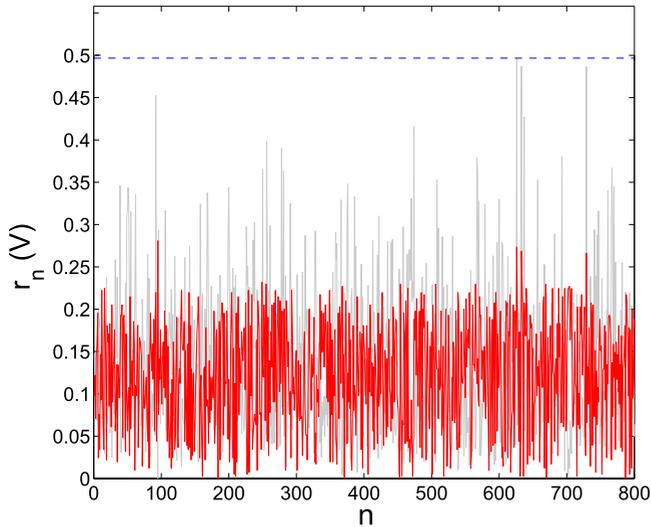
Fig. 13. Time series of the absolute value of the control for a particular trajectory (in solid red). The dashed blue line represents the maximum value of the noise u_0 in the system. The control remains lower than the noise in this situation. In light color the time series of the absolute value of the noise is shown.

the amplitude of the noise is in general higher than the control.

6. Conclusions

We have provided an experimental implementation of the partial control scheme, as described in [Aguirre *et al.*, 2004; Zambrano *et al.*, 2008; Zambrano & Sanjuán, 2009]. We have shown that it can be implemented in electronic circuits simulating the slope-three tent map and a linearly expansive horseshoe map. The main features of this control scheme are recovered, so we expect that this technique might be used in other situations and experimental settings where transient chaotic dynamics exist in a bounded region in phase space in the presence of noise.

Acknowledgments

References

Aguirre, J., d'Ovidio, F. & Sanjuán, M. A. F. [2004] "Controlling chaotic transients: Yorke's Game of Survival," *Phys. Rev. E* **69**, 016203.

Aguirre, J., Viana, R. L. & Sanjuán, M. A. F. [2009] "Fractal structures in nonlinear dynamics," *Rev. Mod. Phys.* **81**, 333–386.

Alligood, K. T., Sauer, T. D. & Yorke, J. A. [1996] *Chaos, An Introduction to Dynamical Systems* (Springer-Verlag, NY).

Altmann, E. G. & Tél, T. [2008] "Poincaré recurrences from the perspective of transient chaos," *Phys. Rev. Lett.* **100**, 174101.

Aziz-Alaoui, M. A. & Bertelle, C. [2009] "Experimental study and characterization of chaos," *Directions in Chaos*, ed. Bai-Lin, H. (World Scientific, Singapore), pp. 149–211.

Dhamala, M. & Lai, Y. C. [1999] "Controlling transient chaos in deterministic flows with applications to electrical power systems and ecology," *Phys. Rev. E* **59**, 1646.

Grebogi, C., Ott, E. & Yorke, J. A. [1982] "Chaotic attractors in crisis," *Phys. Rev. Lett.* **48**, 1507–1510.

Hellen, E. H. [2004] "Real-time finite difference bifurcation diagrams from analog electronic circuits," *Am. J. Phys.* **72**, 499–502.

Kapitaniak, T. & Brindley, J. [1998] "Preserving transient chaos," *Phys. Lett. A* **241**, 41–45.

Ott, E., Grebogi, C. & Yorke, J. A. [1990] "Controlling chaos," *Phys. Rev. Lett.* **64**, 1196.

Place, C. M. & Arrowsmith, D. K. [2000a] "Control of transient chaos in tent maps near crisis. I. Fixed point targeting," *Phys. Rev. E* **61**, 1357.

Place, C. M. & Arrowsmith, D. K. [2000b] "Control of transient chaos in tent maps near crisis. II. Periodic orbit targeting," *Phys. Rev. E* **61**, 1369.

Smale, S. [1967] "Differentiable dynamical systems," *Bull. Amer. Math. Soc.* **73**, 747–817.

Tél, T. [1991] "Controlling transient chaos," *J. Phys. A: Math. Gen.* **24**, 1359–1368.

Yang, F., Li, Q. & Zhou, P. [2007] "Horseshoe in the hyperchaotic MCK circuit," *Int. J. Bifurcation and Chaos* **17**, 4205–4211.

Yang, X.-S. [2009] "Topological horsesheos and computer assisted verification of chaotic dynamics," *Int. J. Bifurcation and Chaos* **19**, 1127–1145.

Zambrano, S. & Sanjuán, M. A. F. [2008] "Control of transient chaos using safe sets," in *Simple Dynamical Systems, Differential Equations, Chaos and Variational Problems*, ed. Staicu, V. (Birkhauser Publishing Ltd.), pp. 425–435.

Zambrano, S., Sanjuán, M. A. F. & Yorke, J. A. [2008] "Partial control of chaotic systems," *Phys. Rev. E* **77**, 055201(R).

Zambrano, S. & Sanjuán, M. A. F. [2009] "Exploring partial control of chaotic systems," *Phys. Rev. E* **79**, 026217.