Available online at www.sciencedirect.com
ScienceDirect
Physics Letters A 362 (2007) 407–411
PHYSICS LETTERS A
www.elsevier.com/locate/pla

Triggering crashes in chaotic dynamics

Frank M. Hilker [a,*], Frank H. Westerhoff [b]

[a] *Gulbenkian Institute of Science, Apartado 14, 2781-901 Oeiras, Portugal*
[b] *Department of Economics, University of Osnabrück, 49069 Osnabrück, Germany*

Received 25 July 2006; received in revised form 4 October 2006; accepted 5 October 2006

Available online 19 October 2006

Communicated by C.R. Doering

Abstract

We discuss a method to generate "crashes" in chaotic systems at prespecified time points (or to target trajectories to desirable states). The idea is first to learn typical crash sequences from available time-series and then to implement a single perturbation of the system to force it on a crash path.

PACS: 05.45.Gg

Keywords: Chaos; Crashes; Intervention; Chaos anticontrol; Targeting

1. Introduction

Deterministic chaos is characterized by irregular, aperiodic motion on a bounded and dense attractor. Chaotic systems exhibit extreme sensitivity to initial conditions, i.e., small perturbations substantially alter the state of the system in the course of time. Though chaotic dynamics thus appears to be erratic, its deterministic nature can be used to control chaos towards periodic motion [1–6], achieve synchronization [7,8] and target trajectories [9]. While the original idea of chaos control is to stabilize irregular dynamics, the aim can also be the opposite, namely to impose a desired behaviour on the chaotic system.

For instance, exploiting the standard chaos control technique [1], Mehta and Henderson [10] formulated a related chaotic system and then forced the orbit on this artificially constructed system by small parameter perturbations. Pyragas [11,12] introduced a continuous control method in which a chaotic system can reproduce previous trajectories through linear coupling with a self-generated time series. [13] showed that the driven signal does not need to be related to the system under study, and can be a quite general one. Furthermore, chaos control techniques can also be applied to convert transient chaos into sustained chaos [14,15]. This kind of chaos maintenance (also called chaos anticontrol) is of interest in systems exhibiting deterministic crises and intermittent behaviour [16–18]. In a biomedical context, for example, the loss of chaos and the onset of periodicity may be regarded as pathologic and is often referred to as a dynamical disease [15].

The problem of targeting trajectories has been addressed for two main reasons. First, in chaos control algorithms, it can reduce waiting times considerably. Second, in systems with coexisting attractors, it can drive the orbit into the basin of attraction of a particular attractor that is regarded to be superior. In fact, many targeting methods have dealt with the distinction between smooth and fractal basin boundaries, cf. [19] and references therein. More generally phrasing, targeting of chaos refers to a process in which perturbations are applied to a dynamical system to steer it to (the neighbourhood of) a prespecified state. Actually, the aim of this Letter is to trigger certain single "events" in a chaotic system, such as crashes or peaks. In addition, one may also use the method to direct the trajectory to an unstable steady states for its temporary "stabilization".

* Corresponding author. Present address: Centre for Mathematical Biology, Mathematical and Statistical Sciences, CAB 501, University of Alberta, Edmonton, AB T6G 2G1, Canada. Tel.: +1 780 492 0215; fax: +1 780 492 8373.
E-mail addresses: fhilker@math.ualberta.ca (F.M. Hilker), frank.westerhoff@uos.de (F.H. Westerhoff).

0375-9601/$ – see front matter
doi:10.1016/j.physleta.2006.10.035

Most targeting algorithms focus on small perturbations of a system's parameter, see for example [20–22]. Pyragas [11] was probably the first to propose to perturb instead the state variables of a system. This idea has also been implemented in the targeting algorithms of [23,24]. Boccaletti et al. [25] presented a method that does not require a priori information about the stable and unstable manifolds of the target point. Instead, they constructed goal dynamics by recording preimages of the target. The advantage is that in principle only time-series information is necessary. This can be of importance when the law of motion is not known as for example in ecological systems [26]. The idea to utilize preimages basically goes back to [15,16] and [11].

While targeting problems were so far concerned with applying small perturbations over a longer time-period, there may be situations in which a single, but possibly larger, intervention is more appropriate. In fact, observing the system's state as well as implementing the perturbation can be costly. Think, for instance, of interventions in population dynamics (e.g., by culling or enrichment). A one-time field campaign is obviously easier to realize and less time-consuming than repeated attempts having only a small impact each (cf. [27] for a review of recent experimental large-scale field studies that perturbed ecological communities by trapping, nest removal [28] and treatment with anthelminthics [29]). Here we discuss a method that triggers crashes at prespecified points in time. By a crash we mean that a dynamic variable drops from a relatively large value below a certain threshold. For example, it may be advantageous to minimize the amount of pest species close to the harvest season. Similarly, our method is general enough to be used to induce peaks or to target the trajectory to desirable system states. Imagine, e.g., to improve harvesting yields within a desired time window.

Our method works as follows. In a first step, we learn which preimages may lead to a crash (corresponding to our target). This is done by inspecting time-series information. If potential crash patterns have been identified, then one can disturb the system such that it is put on one of these crash paths. A few time steps later, a crash will be triggered automatically. The advantages of our method are obvious. Only a single intervention is needed to reach our target. Thereafter, as a side effect, the system dynamics remains unaffected, including the maintenance of the original chaotic attractor. The implementation is entirely based on time-series information, i.e. no knowledge of the underlying law of motion is required. The method works well in a deterministic setting and can also cope with substantial noise.

2. The method

We introduce our approach by exemplifying it through a prototype one-dimensional nonlinear map. The quadratic map, which may, for instance, be regarded as a model of density-dependent growth of a population with nonoverlapping generations [30,31], can be expressed as

$$X_{t+1} = r X_t (1 - X_t), \tag{1}$$

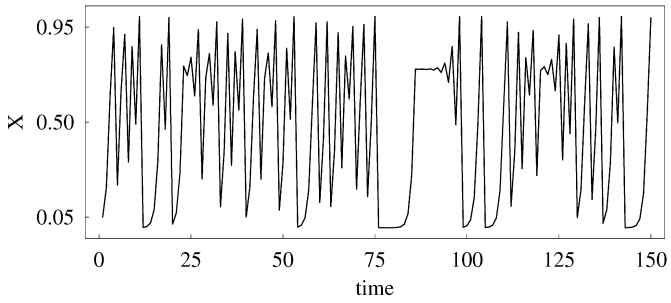

Fig. 1. The dynamics of the quadratic map with $r = 4$ and $X_0 = 0.05$ in the time domain (150 observations).

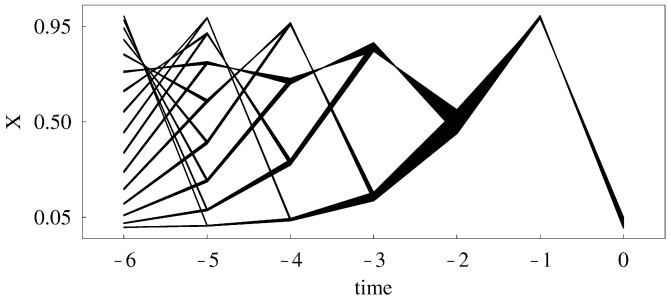

Fig. 2. A collection of 500 crash paths of the quadratic map with $r = 4$ derived from time-series information. A crash is defined as a situation in which X_t drops below $X^C = 0.05$. The data is organized such that a crash occurs at time step zero. Six preimages are plotted.

where the parameter $r \in (0, 4]$ and the initial value $X_0 \in (0, 1)$. Fig. 1 shows a time series of this map with $r = 4$ and $X_0 = 0.05$ in the time domain (150 observations). The system fluctuates in the range $(0, 1)$ and repeatedly displays stronger crashes. For instance, in period 75, the variable decreases from about 0.999 to about 0.001. However, such crashes occur in an unpredictable way. It may be useful to be able to generate crashes at certain future points in time.

The first step of our method is to identify a collection of preimages that lead to a crash. Let us define a crash as a situation in which the variable drops below the threshold value $X^C = 0.05$. Fig. 2 presents 500 crash paths which we have collected from time-series information. A clear "crash pattern" becomes visible. For instance, it can be observed that a crash occurs in period "0" when the trajectory of the system is in one of the following four intervals in period "−4": $[0.033, 0.044]$, $[0.296, 0.322]$, $[0.678, 0.704]$ or $[0.957, 0.967]$. If this is indeed the case, then the system will be in the next time step either in the interval $[0127, 0.167]$ or $[0.833, 0.873]$. In period "−2", the system is located in $[0.444, 0.556]$ from where it will jump into the range $[0.987, 1]$. Finally, the system crashes. As argued by Yang et al. [15], the widths of these intervals tend to shrink in the unstable direction, yet the number of branches approximately double every time step. For instance, in period "−6", we already observe 16 different branches all leading to a crash. Note that these branches are scattered widely over the unit interval, so that in period "−6" the system is not far away from one of the 16 branches.

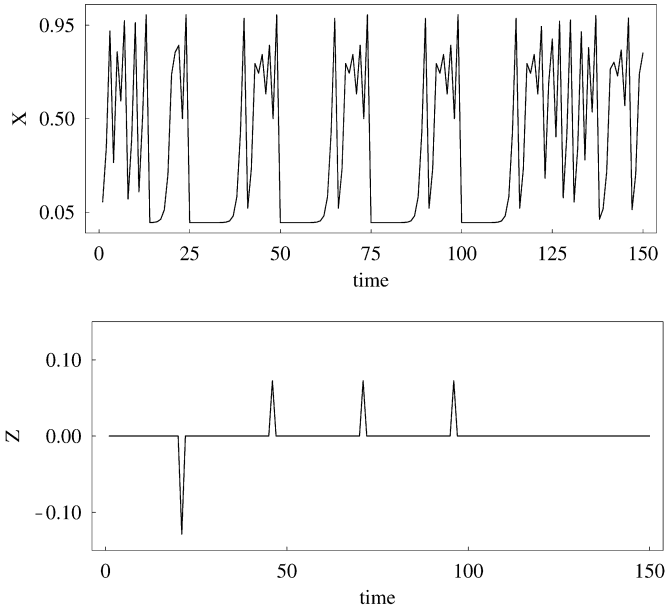

Fig. 3. The top panel shows the dynamics of the manipulated quadratic map (2) and (3) with $r = 4$. The bottom panel depicts the intervention size Z_t. Interventions are executed at time steps 21, 46, 71 and 96, respectively.

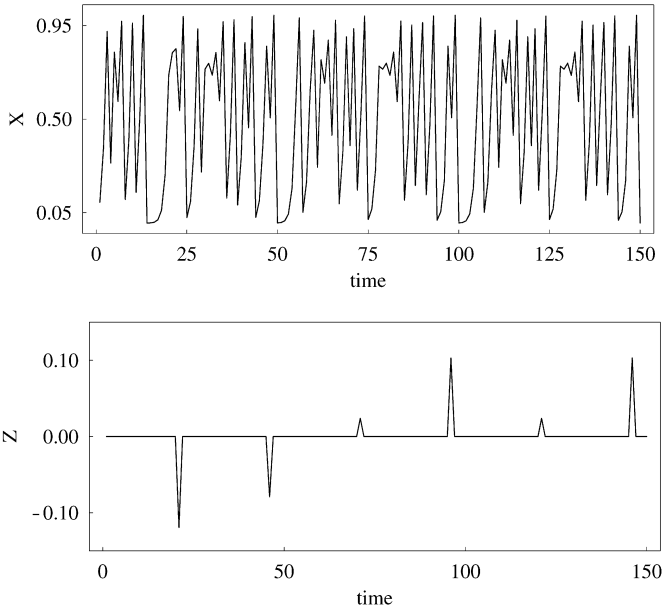

Fig. 4. The same simulation design as in Fig. 3, except that we do not set the system on the midpoint of the closest critical branch, but assume that it drops uniformly into these intervals. Interventions are now executed at time steps 21, 46, 71, 96, 121 and 146, respectively.

The second step of our method consists of an appropriate intervention strategy. For the sake of expository simplicity, we implement the intervention strategy four time steps ahead. The interventions change the law of motion into

$$X_{t+1} = r(X_t + Z_t)\big(1 - (X_t + Z_t)\big), \tag{2}$$

where the (time-dependent) interventions Z_t are executed as

$$Z_t = \begin{cases} (c_j^{\max} + c_j^{\min})/2 - X_t & \text{for } t = \tilde{t}, \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

The intervention time is denoted by $\tilde{t}$ and the lower and upper boundaries of the four aforementioned critical intervals are indicated by $c_j^{\min}$ and $c_j^{\max}$ ($j = 1, \ldots, 4$), respectively (see again Fig. 2). According to this algorithm, we have to intervene only once in the system by placing the trajectory on the midpoint of the closest interval branch c_j. In principle, we can use any of the four branches. However, it may be reasonable to minimize the effort of the intervention.

Fig. 3 depicts a simulation run where we have applied the method in periods 21, 46, 71 and 96. The top panel reveals that the system crashes, as planned, four time periods later, namely in periods 25, 50, 75 and 100. Here, X_t remains below X^C for some time. The bottom panel presents the corresponding interventions Z_t. Note that if we set the trajectory exactly on the midpoint of a crash branch, we may trigger periodic oscillations, because we repeat to reset the system always to the same state. The periodic oscillations, however, immediately vanish if the interventions are stopped (see the last 50 iterations).

It may not always be possible nor advisable to shift the orbit exactly on the midpoint of a branch. Let us now assume that it is possible to force the system onto the closest branch, but that it drops uniformly distributed into the critical region. Fig. 4 illustrates the consequences. Now the intervention dates are 21, 46, 71, 96, as before, and 121 and 146. As is visible, a crash occurs in all six cases four time steps later. Since the interventions have a partially random nature, periodicity cannot be observed anymore. The average intervention size depends on how early we manipulate the system. In general, we should observe that the earlier we intervene the smaller is the necessary intervention size to perturb the orbit on the crash path (since the average distance to the nearest branch decreases backwards in time, cf. Fig. 2).

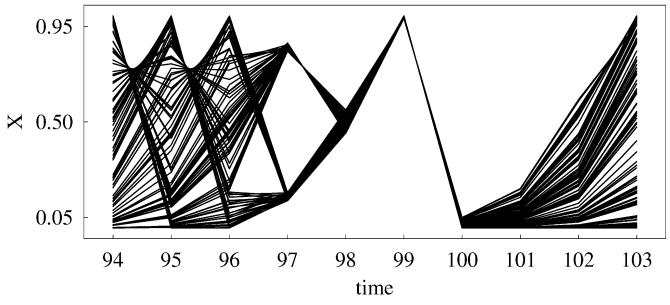

Fig. 5. The dynamics of the manipulated map for 100 different initial values. Interventions are executed at time step 96, at which we force the trajectory randomly in one of the four critical branches.

Fig. 5 is designed to clarify how the intervention method changes the dynamics. Interventions are carried out in period 96 so that crashes occur in period 100. We display 100 simulation runs, generated from randomly drawn initial conditions, for the time window $t \in [94, 103]$. Before the interventions start, the trajectories are scattered over the whole unit interval. Due to the interventions, the system is on the crash path in periods 97, 98, 99, and 100. Afterwards, the system "recovers" rapidly and displays the same (qualitative) motion as before. For instance, in period 103, the trajectories are similarly dispersed as

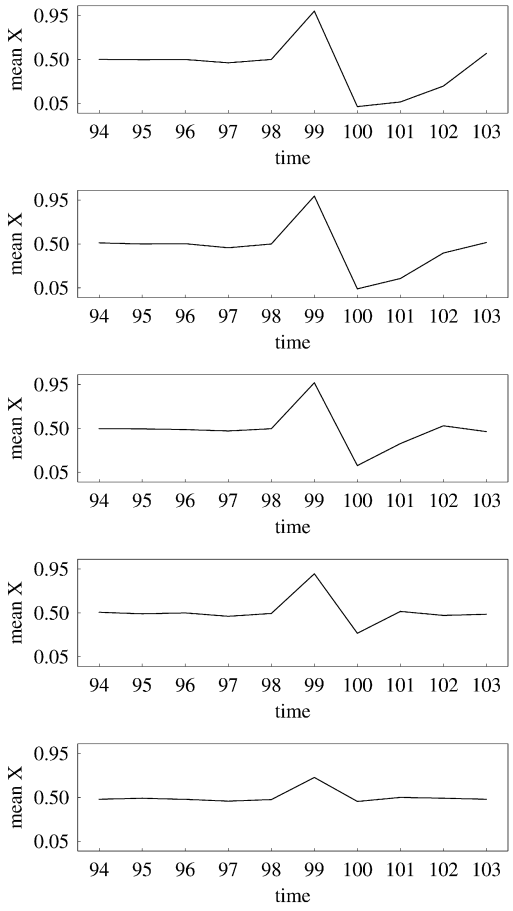

Fig. 6. The panels show the mean dynamics (out of 5000 samples) of the manipulated map with additive noise (4). From top to bottom, we set $\sigma = 0$, $\sigma = 0.01$, $\sigma = 0.025$, $\sigma = 0.05$ and $\sigma = 0.1$, respectively. The interventions are executed in period 96.

in period 94. Hence, the general dynamics of the system is not destroyed; only during the brief intervention and crash period, a temporarily beneficial outcome may be realized.

Finally, we test the effectiveness of the intervention method in the presence of noise. We thus add to the modified map (2) a normally distributed random variable ξ_t with mean zero and constant standard deviation σ:

$$X_{t+1} = r(X_t + Z_t)\big(1 - (X_t + Z_t)\big) + \xi_t. \tag{4}$$

Due to the noise, we furthermore restrict X_{t+1} to remain within the unit interval. The remainder of the simulation design is as in Fig. 5, except that we now display the mean dynamics, which are computed out of 5000 intervention samples. From top to bottom in Fig. 6, we set $\sigma = 0$, $\sigma = 0.01$, $\sigma = 0.025$, $\sigma = 0.05$ and $\sigma = 0.1$, respectively. The method naturally works best in the deterministic case. In period 99, the mean of X_t is clearly above 0.95 and in period 100, the mean of X_t is clearly below 0.05. It takes about two additional time steps until the mean of X_t has recovered and is close to 0.5. Note that the addition of some noise, e.g., $\sigma = 0.01$ or $\sigma = 0.02$, does not weaken the intervention effectiveness much. This is probably because the noisy perturbations are too small to push the system off the crash path. In the presence of noise, it might therefore be advantageous to have a time series with many crashes which can give wide crash pathes. If the noise level gets stronger, the crash effect diminishes. While it may still be possible to raise the trajectory in period 99, the effect for period 100 is basically gone for $\sigma = 0.1$ (given that X_t is bounded between zero and one, a standard deviation of $\sigma = 0.1$ may be considered as quite large).

Recall that we intervene only once, namely in period 96. In the presence of large noise one may improve the effectiveness of the method by repeating interventions until the envisaged crash period is reached. These interventions basically have to counter the exogenous shocks so that the system remains on the crash route. Note also that we still use the crash pattern identified in Fig. 2 which has been detected by observing deterministic crashes. Learning these crash trajectories from noisy time series may blur our results somewhat. On the other hand, if one has some general knowledge of the underlying law of motion, one may even derive the crash trajectories analytically (as demonstrated in [15]).

3. Discussion

We present an approach which may be used to trigger crashes at prespecified periods in time. The method is quite simple. In a first step, one has to identify typical crash paths from time-series information. In a second step, one has to push the trajectory of the system onto a crash route. One advantage of the method is that it requires only a single perturbation. Furthermore, it does not need analytical knowledge of the underlying dynamics. Time-series based approaches often have the serious drawback that they demand a large data acquisition leading to long waiting times. In principle, the method presented here already works when the system orbit visits the target a single time in the available data. This is because our approach does not rely on necessarily small perturbations. Hence, one can choose an appropriate time period for the intervention where the system is relatively close to the identified orbit. In practice, one may, however, prefer to obtain multiple and wider crash paths, in order to reduce the necessary intervention size. This requires either a long time series with a sufficient number of crashes or several time series of the same system. The impact of time series length has been investigated in the context of excluding certain system states such as outbreaks or extinction in population dynamics [26]. Moreover, it should be noted that our algorithm requires the system parameters such as the per-capita growth rate r in Eq. (1) not to be changed over time.

In addition, the method achieves at least partially its goal under noisy conditions. We have illustrated the working of the method using the logistic map. An extension to higher-dimensional systems should be straightforward (cf. the approach in [26]). Since we force the system on the crash path, we will not push it off the chaotic attractor (i.e., the chaotic attractor will remain intact). The embedding theorem actually allows us to reconstruct the full attractor even if we cannot measure all state variables. The only requirement for the method to work in higher-dimensional systems is that we may need to be able to manipulate all dynamic variables of the system. Finally, the focus of the Letter is on triggering crashes. But it is

obvious that the method allows us, at least in a purely deterministic environment, to target also other beneficial states. This may have numerous applications in a wide field of nonlinear systems that need to be casually controlled to exhibit certain dynamical states.

References

[1] E. Ott, C. Grebogi, J.A. Yorke, Phys. Rev. Lett. 64 (1990) 1196.
[2] H.I. McCallum, J. Theor. Biol. 154 (1992) 277.
[3] J. Güémez, M.A. Matías, Phys. Lett. A 181 (1993) 29.
[4] L. Stone, Nature 365 (1993) 617.
[5] S. Parthasarathy, S. Sinha, Phys. Rev. E 51 (1995) 6239.
[6] N.J. Corron, S.D. Pethel, B.A. Hopper, Phys. Rev. Lett. 84 (2000) 3835.
[7] L.M. Pecora, T.L. Carroll, Phys. Rev. Lett. 64 (1990) 821.
[8] Y.-C. Lai, C. Grebogi, Phys. Rev. E 47 (1993) 2357.
[9] T. Shinbrot, E. Ott, C. Grebogi, J.A. Yorke, Phys. Rev. Lett. 65 (1990) 3215.
[10] N.J. Mehta, R.M. Henderson, Phys. Rev. A 44 (1991) 4861.
[11] K. Pyragas, Phys. Lett. A 170 (1992) 421.
[12] K. Pyragas, Phys. Lett. A 181 (1993) 203.
[13] M.A. Matías, J. Güémez, Phys. Lett. A 209 (1995) 48.
[14] Y.-C. Lai, C. Grebogi, Phys. Rev. E 49 (1994) 1094.
[15] W. Yang, M. Ding, A.J. Mandell, E. Ott, Phys. Rev. E 51 (1995) 102.
[16] Y. Nagai, Y.-C. Lai, Phys. Rev. E 51 (1995) 3842.
[17] V. In, S.E. Mahan, W.L. Ditto, M.L. Spano, Phys. Rev. Lett. 74 (1995) 4420.
[18] P. Parmananda, M. Eiswirth, Phys. Rev. E 54 (1996) R1036.
[19] S. Boccaletti, C. Grebogi, Y.-C. Lai, H. Mancini, D. Maza, Phys. Rep. 329 (2000) 103.
[20] E.J. Kostelich, C. Grebogi, E. Ott, J.A. Yorke, Phys. Rev. E 47 (1993) 305.
[21] T. Shinbrot, C. Grebogi, J.A. Yorke, E. Ott, Nature 363 (1993) 411.
[22] T. Shinbrot, Adv. Phys. 44 (1995) 73.
[23] D. Gligoroski, D. Dimovski, V. Urumov, Phys. Rev. E 51 (1995) 1690.
[24] S. Boccaletti, F.T. Arecchi, Physica D 96 (1996) 9.
[25] S. Boccaletti, A. Farini, E.J. Kostelich, F.T. Arecchi, Phys. Rev. E 55 (1997) R4845.
[26] F.M. Hilker, F.H. Westerhoff, q-bio.PE/0606042.
[27] R.M. May, Nature 398 (1999) 371.
[28] E. Korpimäki, K. Norrdahl, Ecology 79 (1998) 2448.
[29] P.J. Hudson, A.P. Dobson, D. Newborn, Science 282 (1998) 2256.
[30] R.M. May, Science 186 (1974) 645.
[31] R.M. May, Nature 261 (1976) 459.