

Publishers' page

Publishers' page

Publishers' page

Publishers' page

Contents

| | | |
|-------|---|----|
| 1. | Clipping Chaos to Cycles | 1 |
| 1.1 | Application to One-dimensional Maps | 2 |
| 1.1.1 | Analysis | 4 |
| 1.1.2 | Application to Encoding and Information Storage | 7 |
| 1.2 | Application to Multi-dimesnional Systems | 11 |
| 1.2.1 | Controlling the Lorenz System | 11 |
| 1.2.2 | Controlling Neuronal Spikes | 11 |
| 1.2.3 | Smart Matter Application | 13 |
| 1.2.4 | Thresholding at Varying Intervals to obtain different Temporal Patterns | 16 |
| 1.3 | Experimental verification | 18 |
| 1.3.1 | Controlling a Circuit Realisation of Jerk Equations | 18 |
| 1.3.2 | Controlling Chua's Circuit | 20 |
| 1.3.3 | Controlling Hyperchaos | 21 |
| 1.4 | Conclusions | 23 |
| | <i>Bibliography</i> | 29 |

Chapter 1

Clipping Chaos to Cycles

Sudeshna Sinha

*The Institute of Mathematical Sciences, Chennai 600 113, India*¹

In this chapter we will review a powerful control strategy based on the simple and easily implementable *threshold* mechanism [1]. The central idea is as follows: consider a general N -dimensional dynamical system, described by the evolution equation $\frac{d\mathbf{x}}{dt} = F(\mathbf{x}; t)$ where $\mathbf{x} \equiv (x_1, x_2, \dots, x_N)$ are the state variables. Say variable x_T is chosen to be monitored and threshold controlled. The prescription for thresholding in this system is as follows: control will be triggered whenever the value of the monitored variable exceeds the critical threshold x^* (i.e. when $x_T > x^*$) and the variable x_T will then be re-set to x^* . The dynamics continues till the next occurrence of x_T exceeding the threshold, when control resets its value to x^* again. So in this method no parameters are adjusted, and only one state variable is occasionally reset.

This method is not based on stabilising unstable periodic orbits [2]. Rather, the threshold mechanism *clips* the chaotic orbit to periodic time sequences of desired lengths. So the effect of this scheme is to *limit the dynamic range* slightly, i.e. “snip” off small portions of the available phase space, and this action effectively yields a very wide range of stable regular dynamics [3].

In the sections below we will analyse the control achieved by thresholding in different prototypical systems, with varying levels of complexity, including the challenging task of controlling hyperchaos. We will also discuss the experimental implementation of the scheme on a range of strongly nonlinear electronic circuits.

¹email : sudehra@imsc.ernet.in

Table 1.1 Controlling a 1-dimensional chaotic map by thresholding

| Threshold | Nature of Controlled Orbit |
|-----------------------|----------------------------|
| $x^* < 0.75$ | Period 1 (fixed point) |
| $0.75 < x^* < 0.905$ | Period 2 Cycle |
| $x^* \sim 0.965$ | Period 3 Cycle |
| $0.905 < x^* < 0.925$ | Period 4 Cycle |
| $x^* \sim 0.979$ | Period 5 Cycle |
| $x^* \sim 0.93$ | Period 6 Cycle |
| $x^* \sim 0.9355$ | Period 7 Cycle |
| $x^* \sim 0.932$ | Period 8 Cycle |
| $x^* \sim 0.981$ | Period 9 Cycle |
| $x^* \sim 0.95$ | Period 10 Cycle |

Threshold values vs. periodicity, of a few representative controlled cycles, for the chaotic logistic map $x_{n+1} = 4x_n(1 - x_n)$. Note that cycles of the same period, but different geometries, can be obtained in different threshold windows.

1.1 Application to One-dimensional Maps

When the dynamics of the uncontrolled system is given by $x_{n+1} = f(x_n)$ where f is a nonlinear function, the threshold mechanism is simply implemented as the following condition: if variable $x_{n+1} > x^*$ then the variable is adjusted back to x^* , namely the threshold x^* is the critical value the state variable is not allowed to exceed, and control is triggered whenever the variable grows larger than this threshold. The effect of this simple thresholding is dramatic: it yields stable periodic orbits of all orders. See Table 1.1 for a illustrative list of controlled orbits obtained for a range of threshold values.

The threshold controlled chaotic map is effectively a beheaded map (or the “flat top” map), i.e. the unimodal map cut off by the $x_{n+1} = x^*$ line [1],[4]. The level at which the map is chopped off depends on the threshold x^* (see Fig.1.1). This map can yield periodic orbits of various orders under different threshold values. Note also that these orbits are *stable* and the low order cycles have fairly large windows of stability in threshold parameter space. Control latency is very short, and once the system exceeds the critical value it is trapped immediately in a stable cycle whose order is determined by the value of the threshold.

The basis of the marked success of this method is clear for one dimensional maps. It is best rationalised through the fixed points of the effective map obtained from the chaotic unimodal map under threshold mechanism, namely the beheaded map mentioned above. The fixed points of this map under varying heights of truncation (i.e. different thresholds) give the different periods. For instance, Fig.1.1 shows the first iterate x_{n+1} of the thresholded logistic map. One obtains a fixed point from the intersection of this map and the $x_{n+1} = x_n$ line, namely the 45° line. When the threshold is lower than 0.75 one gets an intersection of the flat portion of the map (i.e. $x = x^*$) and the $x_{n+1} = x_n$ line. This fixed point solution at $x = x^*$

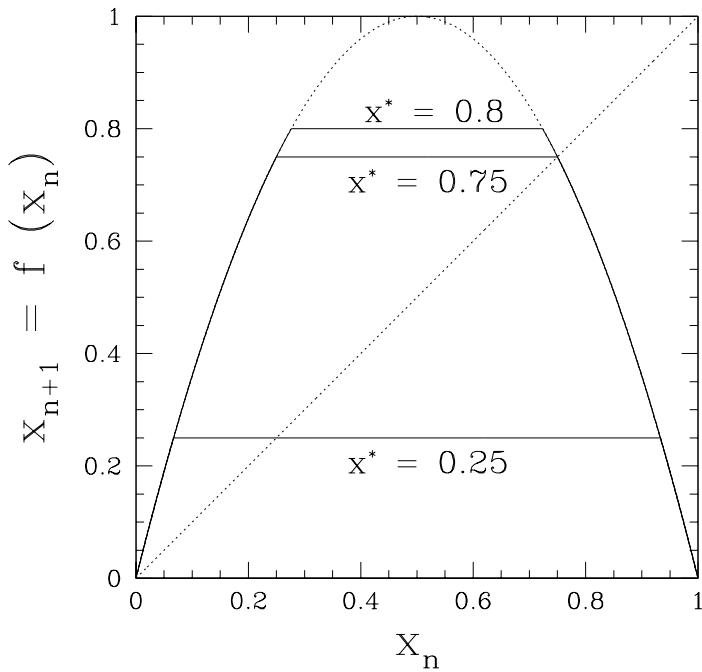


Fig. 1.1 The chaotic logistic map under threshold control, with threshold values 0.25, 0.75 and 0.8. The figure shows the first iterate x_{n+1} (—) of the effective thresholded map, as well as the $x_{n+1} = x_n$ line (i.e. the 45° line (...)). It is clear that the intersection of the flat portion of the map x_{n+1} with the 45° line yields a superstable fixed point of period 1.

has slope zero and thus is *superstable*. This happens for all thresholds $x^* < 3/4$, and one obtains a stable fixed point at x^* for those thresholds. Thus one can control the chaos to a continuous set of fixed points in the range $[0, 3/4]$ by thresholding.

It is also evident from Fig.1.1 that when the threshold exceeds $3/4$ (for example $x^* = 0.8$) the thresholded map does not yield an intersection of the flat portion and the $x_{n+1} = x_n$ line. Rather, it only has the usual fixed point at $x = 0.75$, which is the same as that obtained for the (un-thresholded) chaotic map. This has slope greater than one and is thus unstable. So for thresholds greater than $x^* = 0.75$, the thresholding action does not yield a stable period 1 solution. However *higher order periods* can be obtained. For example Fig.1.2 shows the second iterate of the thresholded map, with threshold $x^* = 0.8$, yielding a fixed point from the intersection of the flat portion of the map and the $x_{n+2} = x_n$ line. This point has slope zero and thus yields a *superstable period 2 orbit*.

Similarly, in Fig.1.3 one sees the first four iterates of the threshold map and the

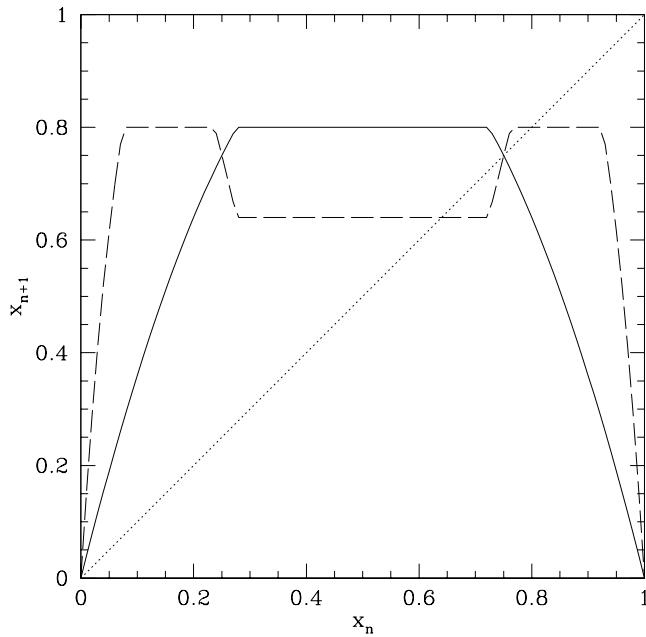


Fig. 1.2 The chaotic logistic map under threshold control, with threshold value 0.8. The figure shows the first iterate x_{n+1} (—) and second iterate x_{n+2} (- - -) of the effective thresholded map, as well as the 45^0 line (...). It is clear that the intersection of the flat portion of the map x_{n+2} with the 45^0 line yields a superstable fixed point of period 2.

fourth iterate yields a superstable fixed point at the intersection of the flat portion of the effective map and $x_{n+4} = x_n$ line. Thus one obtains a stable period 4 orbit at this value of threshold.

In terms of probability densities, the chaotic map under threshold mechanism will map large intervals onto a severely contracting region. Essentially large intervals will get mapped onto a point. This is the reason why the transient control period is so small, and the method is so powerful and stable.

1.1.1 Analysis

For one dimensional maps, the *correspondence of the periodicity of the controlled orbit and the threshold can be obtained exactly*. So one can directly calculate what periodicity will emerge when a certain threshold is set. Further one can obtain the answer to the reverse (important) question as well: what threshold do we need to set in order to obtain a certain period [1].

Now the starting point of the analysis is the fact that the chaotic system is ergodic and thus it is guaranteed to exceed threshold at some point in time. At

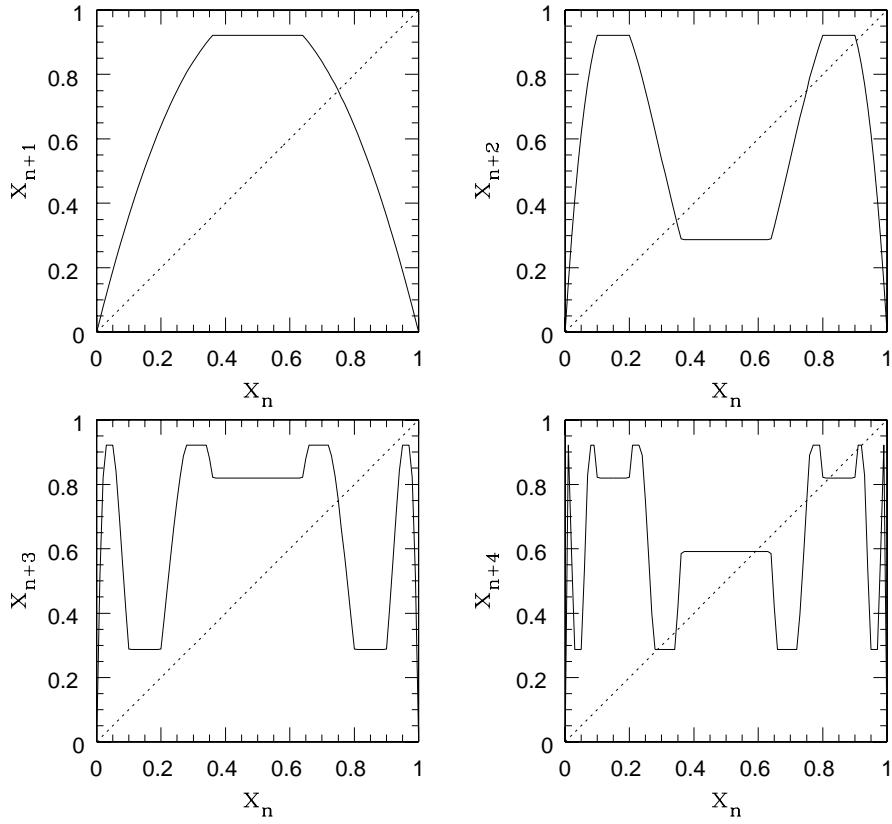


Fig. 1.3 The chaotic logistic map under threshold control, with threshold value 0.922. The figure shows the first four iterates of the effective thresholded map: x_{n+1} , x_{n+2} , x_{n+3} , x_{n+4} , as well as the 45^0 line (...). It is clear that the intersection of the flat portion of the map x_{n+4} with the 45^0 line yields a superstable fixed point of period 4. Note that only for the fourth iterate x_{n+4} does one obtain a solution at the intersection of the flat portion of the effective thresholded map, where slope is zero. So only the period 4 orbit is stable at this threshold.

that point its state is re-set to x^* . One then studies the forward iterations of the map, starting from this state $x = x^*$, i.e. $f_0(x^*)$, $f_1(x^*)$..., where $f_k(x^*)$ is the k^{th} iterate of the map. That is:

1. $k = 0$; $f_0(x^*) = x^*$
 2. $k = 1$; $f_1(x^*) = 4x^*(1 - x^*)$
 3. $k = 2$; $f_2(x^*) = 4(4x^*(1 - x^*))(1 - 4x^*(1 - x^*))$
- and so on. In general

$$f_k(x^*) = f \circ f_{n-1}(x^*) = f \circ f \circ \dots f \circ (x^*)$$

Whenever the $f_k(x^*)$ vs. x^* curve crosses above the $f_0 = x^*$ line (i.e. the 45° line) we have a k cycle, as this implies that the k^{th} iterate exceeds the critical value x^* and thus is adapted back to x^* ($= f_0$, which is the first point in the cycle). For instance, in the range $0 \leq x^* \leq \frac{3}{4}$ the $f_1(x^*)$ curve lie above the f_0 curve (i.e. $f_1(x^*) > x^*$). So the chaotic element is adapted back to x^* at every iterate, yielding a period 1 fixed point. In the range $\frac{3}{4} < x^* < 0.9$ the $f_1(x^*)$ curve dips below the 45° line, but the $f_2(x^*)$ curve lies above the 45° line. This imples that the second iterate of the map (starting from $x = x^*$) exceeds threshold and is adapted back to x^* , thus giving rise to a period 2 cycle. Thus the cycle at each value of threshold is the smallest k such that the k^{th} iterate of the map (starting from $x_0 = x^*$) is greater than x^* , i.e. $f_k(x^*) > x^*$. In this manner the threshold mechanism leads to regular cyclic evolution, whose period depends on the threshold. The chaotic element can then yield a wide variety of dynamical behaviour determined by the threshold.

Thus in threshold parameter space we can find “windows” of various cycles. These are intervals where the following equation is satisfied: Period $P(x^*) = k$ iff $f_k(x^*) \geq x^*$ and $f_l(x^*) < x^*$ for all $l < k$. $P(x^*)$ is a piecewise continuous function of x^* .

For every cycle of periodicity k there will be several windows (with an upper bound of 2^{k-1} windows for period k). The “middle” of the period k windows lies approximately where the curve $f_k(x^*)$ touches 1 (since if it touches 1 it has to have exceeded x^* , as the value of x^* is bounded by 1). Then the solutions of the equation $f_k(x^*) = 1$ gives the x^* values corresponding to a period k . The solutions can be formulated as: $f^{-1} \circ f^{-1} \circ f^{-1} \circ f^{-1}(1)$ where f^{-1} is the (double valued) inverse map. The inverse map: $f^{-1}(y) = \frac{1}{2} \pm \frac{\sqrt{1-y}}{2}$ has two values – one on the right and one on the left of the centre of the interval ($x = \frac{1}{2}$). We will denote these as R and L respectively. Note that for $f^{-1}(1)$ the value of $L(1) = R(1) = \frac{1}{2}$. So the number of distinct values arising from the expression $f^{-1} \circ f^{-1} \dots f^{-1}(1)$ is 2^{k-1} (arising from the 2^{k-1} different possible combinations of R and L).

The evaluation of this algebraic expression for various values of k is simple and direct. Now the existence of a window of period k ($k > 1$) is dependent on the pervious iterates as well, i.e. a solution for period k may be masked by the fact that some iterate l , $l < k$, may have $f_l(x^*) > x^*$. For instance for $k > 1$ all combinations starting with symbol L are masked by period 1 (as the period 1 window extends from 0 to $\frac{3}{4}$ and $L(x) \leq \frac{1}{2}$). So half of the combinations of $f^{-1} \circ f^{-1} \dots f^{-1}(1)$ are swallowed by period 1. One has to examine the remaining 2^{k-2} combinations to check which ones survive swallowing by lower order windows.

However note that one family of windows is guaranteed to exist, namely $RL^{k-1}(1)$, as all iterates leading up to 1 here (i.e. all the subsequences $L(1)$, $L^2(1)$, ... $L^{k-1}(1)$) have value less than $\frac{1}{2}$ (as they are all composed of L). Since

all relevant thresholds for $k > 1$ are greater than $\frac{3}{4}$ it implies that all the iterates leading up to $f_k(x^*)$ have value less than x^* and so this sequence will always yield period k (not any other lower period). *So all possible periods k have atleast one stable window in threshold space.* That is, threshold control for a one dimensional map yields periods of all orders.

Now the analytical results based on symbolic dynamics [1] outlined above are exactly corroborated in circuit realisations of one-dimensional discrete time systems [5] (see Fig.1.4 for traces of representative controlled orbits). Note that chaos is advantageous here, as it possesses a rich range of temporal patterns which can be clipped to wide ranging stable behaviours. This immense variety is not available from thresholding regular systems.

Also note that arrays of thresholded nonlinear elements have been designed, fabricated, and tested in CMOS [6] and such arrays show a wide range of controlled spatiotemporal cycles, consistent with the analytical results.

In marked contrast to many control methods where chaotic trajectories in the vicinity of unstable fixed points are controlled onto these points, in threshold control the system *does not have to be close to any particular unstable fixed point before implementing the control.* Once the trajectory exceeds the threshold it is caught immediately in a stable orbit. So there is no significant interval between the onset of control and the achievement of control, as a wide interval is open to targetting.

Caveat: While threshold control will always yield some regular orbit, it is not clear at the outset exactly what kind of dynamic behaviour will result from a given threshold value [7]. This limitation is overcome easily however through one initial exploratory run over threshold parameter space to map out the dynamic behaviours obtained for different thresholds. Such a “calibration run” once done, makes the scope of the threshold mechanism apparent at the outset, and yields a *look-up table* for the system (relating threshold value to controlled period) for all further applications. The possibility of having this kind of a look-up table to effect control to a “library” of patterns is one of the most powerful features of this method.

1.1.2 Application to Encoding and Information Storage

Information storage is a fundamental function of computing devices. Computer memory is implemented by computer components that retain data for some interval of time. Storage devices have progressed from punch cards and paper tape to magnetic, semiconductor and optical disc storage by exploiting different natural physical phenomena to achieve information storage. For instance, the most prevalent memory element in electronics and digital circuits is the flip-flop or bistable multivibrator which is a pulsed digital circuit capable of serving as a one-bit memory, namely storing value 0 or 1. More meaningful information is obtained by combining consecutive bits into larger units. Here we briefly review a different direction in designing information storage devices: namely we describe schemes to

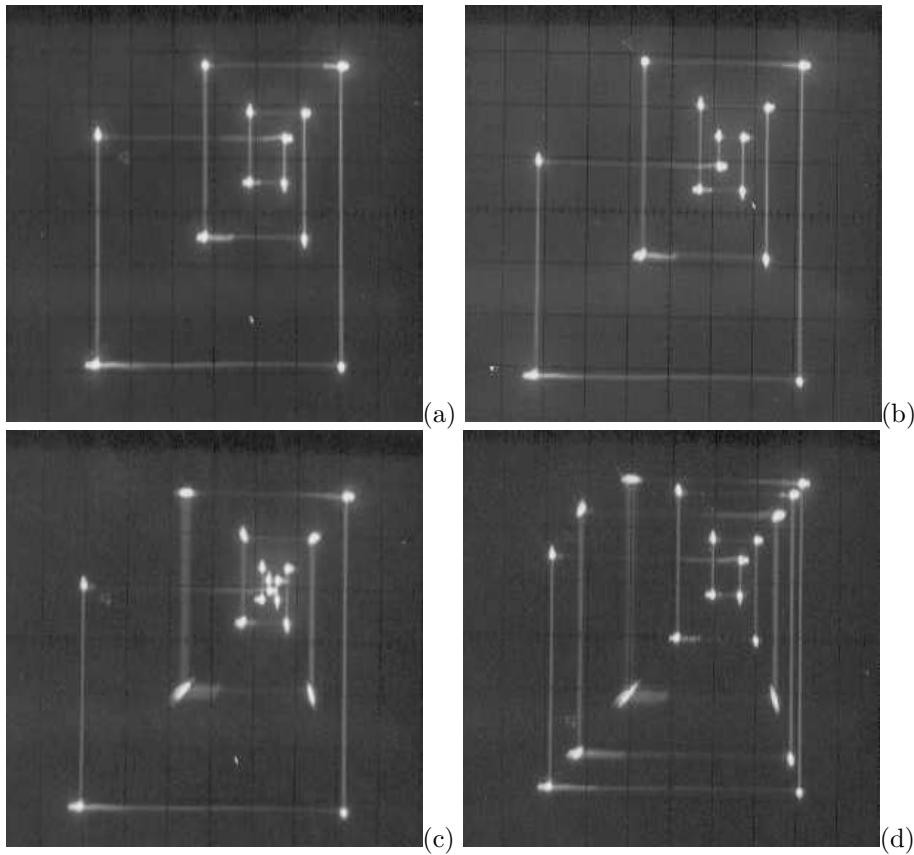


Fig. 1.4 Experimental verification of a range of controlled periods in a circuit realization of the logistic map. The ordinate and abscissa represent traces of x_{n+1} and x_n for (a) period 6 cycle (b) period 7 cycle (c) period 9 cycle with threshold and (d) period 10 cycle. The threshold levels at which these cycles were obtained coincide exactly with those predicted theoretically.

store data using the vast variety of patterns and distinct behaviours that can be extracted by thresholding nonlinear maps.

The aim is to utilize arrays of nonlinear elements to stably encode and store various items of information (such as patterns and strings) to create a database [8]. Further we indicate how this storage method also allows one to utilize the nonlinear dynamics of the array elements in order to determine the number of matches (if any) to specified items of information in the database.

Encoding information: We consider encoding N data elements, each comprised of one of M distinct items. N can be arbitrarily large and M is determined by the kind of data being stored. For instance for storing English text one can consider the letters of the alphabet to be the natural distinct items building the database, namely $M = 26$. Or, for the case of data stored in decimal representation $M = 10$, and for databases in bioinformatics comprised typically of symbols A , T , C , G ,

one has $M = 4$. One can also consider strings and patterns as the items. For instance for English text one can also consider the keywords as the items, and this will necessitate larger M as the set of keywords is large.

Now we demonstrate a method which stores data by utilizing the abundance of distinct stable behaviors obtained by thresholding a chaotic system. This ability of the thresholded chaotic map to be in a variety of fixed states gives it the capacity to represent a large set of items.

A database of size N is stored by N thresholded chaotic elements. Each dynamical element stores one element of the database, encoding any one of the M items comprising our data. Now in order to hold information one must confine the dynamical system to a fixed point behavior, i.e. a state that is stable and constant throughout the dynamical evolution of the system.

As analyzed above, typically a large window of threshold values can be found where the system is confined on fixed points, namely, the state of the element under thresholding is stable at \mathcal{T} i.e. $x = \mathcal{T}$, where \mathcal{T} is the threshold, for all times. So one can choose a large set of distinct thresholds $\mathcal{T}^1, \mathcal{T}^2, \dots, \mathcal{T}^M$, within the fixed point window, with each threshold having a one-to-one correspondence with a distinct item of our data. Thus the number of distinct items that can be stored in a single dynamical element is typically large, with the size of M limited only by the precision of the threshold setting.

In particular, consider a collection of storage elements that evolve in discrete time according to the tent map, $f(x) = 2 \min(x, 1-x)$, with each element storing one element of the given database. Each element can hold any one of the M distinct items. As described above, a threshold will be applied to each dynamical element to confine it to the fixed point corresponding to the item to be stored.

For the tent map, thresholds ranging from 0 to $2/3$ yield fixed points, namely $x = \mathcal{T}$, for all time, when threshold $0 < \mathcal{T} < 2/3$. This can be obtained exactly from the fact that $f(x) > \mathcal{T}$ for all x in the interval $(0, 2/3)$, implying that the subsequent iteration of a state at \mathcal{T} will always exceed \mathcal{T} , and thus get reset to \mathcal{T} . So x will always be held at value \mathcal{T} .

In our encoding, the thresholds are chosen from the interval $(0, 1/2)$, namely a sub-set of the fixed point window $(0, 2/3)$. For specific illustration, with no loss of generality, consider each item to be represented by an integer i , in the range $[1, M]$. Defining a resolution r between each integer as $r = \frac{1}{2} \frac{1}{M+1}$ gives a lookup map from the encoded number to the threshold, namely relating the integers i in the range $[1, M]$, to threshold in the range $[r, 1/2 - r]$, by: $\mathcal{T} = ir$.

Therefore we obtain a direct correspondence between a set of integers ranging from 1 to M , where each integer represents an item, and a set of M threshold values. So we can store N database elements by setting appropriate thresholds on N chaotic maps. Clearly, if the threshold setting has more resolution, namely smaller r , then a larger range of values can be encoded. Note however that precision is not a restrictive issue here, as different representations of data can always be chosen in

order to suit a given precision of the threshold mechanism.

Processing Information: Now we briefly indicate how we can search an arbitrarily large unsorted database set up as above, for the existence of a specific item, by performing just one global operation on the whole array. The basic principle here is that one can construct a single suitable global operation that acts on the nonlinear elements encoding the database such that only elements encoding the matching items yield a prescribed, easily measurable property. This enables the occurrence(s) of matches to be identified easily. We give some details below.

Given a database stored by setting appropriate thresholds on N dynamical elements, we can query for the existence of a specific item in the database by globally shifting the state of all elements of the database up by the amount that represents the item searched for. Specifically the state of all the elements is raised to $x + Q$, where Q is a search key given by: $Q^k = 1/2 - T^k$, where k is the number being queried for. So the value of the search key is simply $1/2$ minus the threshold value corresponding to the item being searched for. This addition shifts the interval that the database elements can span, from $[r, 1/2 - r]$ to $[r + Q^k, 1/2 - r + Q^k]$, where Q^k is the globally applied shift.

Notice that the information item being searched for, is coded in a manner “complimentary” to the encoding of the items in the database (much like a key that fits a particular lock), namely $Q^k + T^k$ adds up to $1/2$. This guarantees that *only* the element matching the item being queried for will have its state shifted to $1/2$. The value of $1/2$ is special in that it is the only state value that on the subsequent update will reach the value of 1.0 , which is the maximum state value for this system. So only the elements holding an item matching the queried item will reach extremal value 1.0 on the dynamical update following a search query. Note that the important feature here is the nonlinear dynamics that maps the state $1/2$ to 1 , while all other states (both higher and lower than $1/2$) get mapped to values lower than 1 .

Basically in unimodal maps, the maximal point can act as a “pivot” for the “folding”. This provides us with a single global monitoring operation to push the state of all the elements matching the queried item to the unique maximal point, in parallel.

To complete the search we now must detect the extremal state. This can be accomplished in a variety of ways. For example, one can simply employ a level detector to register all elements at the maximal state. This will directly give the total number of matches, if any. So the total search process is rendered simpler as the state with the matching pattern is selected out and mapped to the maximal value, allowing easy detection. Further, by relaxing the detection level by a prescribed “tolerance”, we can check for the existence within our database of numbers or patterns that are close to the number or pattern being searched for. So nonlinear dynamics works as a powerful *preprocessing tool*, reducing the determination of matching patterns to the detection of maximal states, an operation that can be

accomplished by simple means, in parallel.

1.2 Application to Multi-dimensional Systems

The action of thresholding on 1-dimensional chaotic maps can be analyzed exactly, as outlined in the section above. However for multi-dimensional systems such an exact calculation is not possible. So one has to rely on numerics and experiments to gauge the scope of threshold control on such systems.

The central issue in multi-dimensional systems is whether or not the thresholded state variable can enslave the rest of the variables to some regular dynamical behaviour. With this in mind we present several examples below, of controlling highly coupled strongly nonlinear high dimensional systems, by thresholding just *one* variable.

1.2.1 Controlling the Lorenz System

First we demonstrate the action of the threshold mechanism on a system of 3 coupled ODE's: the chaotic Lorenz attractor (a system known to be relevant to lasers [9]). It is given by

$$\begin{aligned}\dot{x} &= \sigma(y - x) \\ \dot{y} &= rx - y - xz \\ \dot{z} &= xy - bz\end{aligned}\tag{1.1}$$

with parameters $\sigma = 10$, $r = 28$, $b = 8/3$.

In order to check whether or not *one* thresholded state variable can drag the rest of the multidimensional system to some fixed dynamical behaviour, we impose threshold control on any one of the three variables of the Lorenz system, i.e. one demands that either variable x or y or z must not exceed a prescribed threshold value x^* . Fig.1.5 show some representative results of this thresholding action for a range of threshold values. It is clear that the mechanism successfully controls limit cycles of varying sizes and geometries.

1.2.2 Controlling Neuronal Spikes

In neuronal systems, a wealth of complex patterns have been experimentally observed in a variety of cases [10]. However the mechanisms by which such complex spiking patterns can be manipulated are not well understood. It is thus of considerable interest and potential utility to devise control algorithms capable of achieving the desired type of behaviour in such complex systems.

Here we use threshold control to target desired firing patterns in a prototypical model of a Hippocampal neuron: the Pinsky-Rinzel model [11]. This model neuron

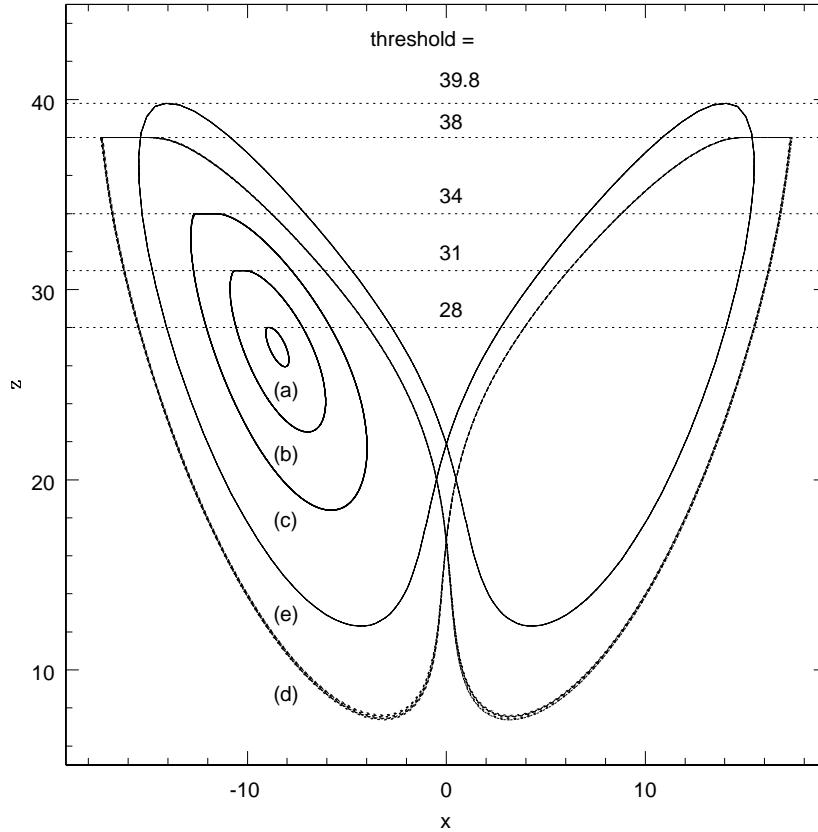


Fig. 1.5 The chaotic Lorenz attractor (with $\sigma = 10, r = 28, b = 8/3$), under threshold control of variable z . The dotted lines indicate the value of the threshold.

consists of somatic and dendritic compartments resistively coupled at different potentials. A patch of the cell membrane is modeled as an equivalent electrical circuit consisting of a resistor and a capacitor in parallel. The current balance equations for the two compartments follow from differentiating the capacitance definition.

The model has 8 variables: the 5 gating variables: h, n, s, c and q , the Ca level and the soma voltage V_s and dendrite voltage V_d . The parameters include the coupling conductance between soma and dendrite g_c , reversal potentials $V_{Na}, V_{Ca}, V_K, V_l, V_{syn}$, ionic conductances $g_l, g_{Na}, g_{KDR}, g_{Ca}, g_{KAHP}, g_{KCa}$, synaptic conductances g_{MDA}, g_{AMPA} , relative area of soma to dendrite p , membrane capacitance c_m , and the applied soma current i_s . The details of the dynamical equations and parameters are given in [11].

In the context of neuronal systems, it is unrealistic to implement the threshold

mechanism on the gating variables or the Ca levels as it is unlikely that one can manipulate these externally with ease. On the other hand, it is natural to try and implement the threshold action on the somatic or dendritic voltages V_s or V_d as they are much more accessible to measurement and monitoring. Thus we demand that variable V_s (or V_d) must not exceed a prescribed threshold value V_* (1 mV $< V^* < 20$ mV), and examine the scope of this mechanism to yield regular firing behaviour. The note-worthy feature here is that *only one variable is thresholded in this strongly nonlinear, highly coupled, 8-dimensional system*.

Fig.1.6 shows a representative result of thresholding the neuronal system. It is clearly evident that the mechanism manages to yield complete regularity, as compared with the very irregular and infrequent firing behavior of the neuron with no thresholding. So the thresholded variable has the ability to drag the rest of this high-dimensional excitable system to regular dynamical behavior (see Fig.1.6). The threshold mechanism typically yields two types of behavior: fixed states and states with regular spiking (with interspike intervals ranging from about 14 msec to 60 msec). Low threshold leads to the first dynamics and higher thresholds leads to regular firing states.

Similar control is achieved by thresholding the somatic voltage V_s . We also checked that the method works for slightly delayed threshold action, that is a scenario where the variable is brought down to the threshold value after a small delay (as is conceivable in real set-ups where there may be a small delay between the detection of the crossing of the threshold condition and the re-setting of the state variable). We find that the method is still as effective [12].

1.2.3 Smart Matter Application

Here we present an application of threshold control to the interesting problem of controlling an unstable elastic array, which has been used as a prototypical model for “smart matter” [13]. It is clear that in such a context, where the system contains many elements, the effectiveness of control algorithms which rely on access to the full state of the system and detailed knowledge of its behaviour, is limited. Hence the present approach, which needs *local information from very few sites* (and no knowledge of the dynamics) in order to implement the necessary control, can prove to be of considerable utility.

For example, consider the general extended system:

$$\frac{d^2\mathbf{x}}{dt^2} = A\mathbf{x} - G\dot{\mathbf{x}} \quad (1.2)$$

where the vector \mathbf{x} gives the displacements of the elements in the array, matrix A contains the system’s coupling parameters and G is the damping. In this array we now implement threshold control on a few selected sites.

Specifically, one can consider a model of buckling instability of beams [13]: an elastic array of N elements coupled to nearest neighbours by springs with spring

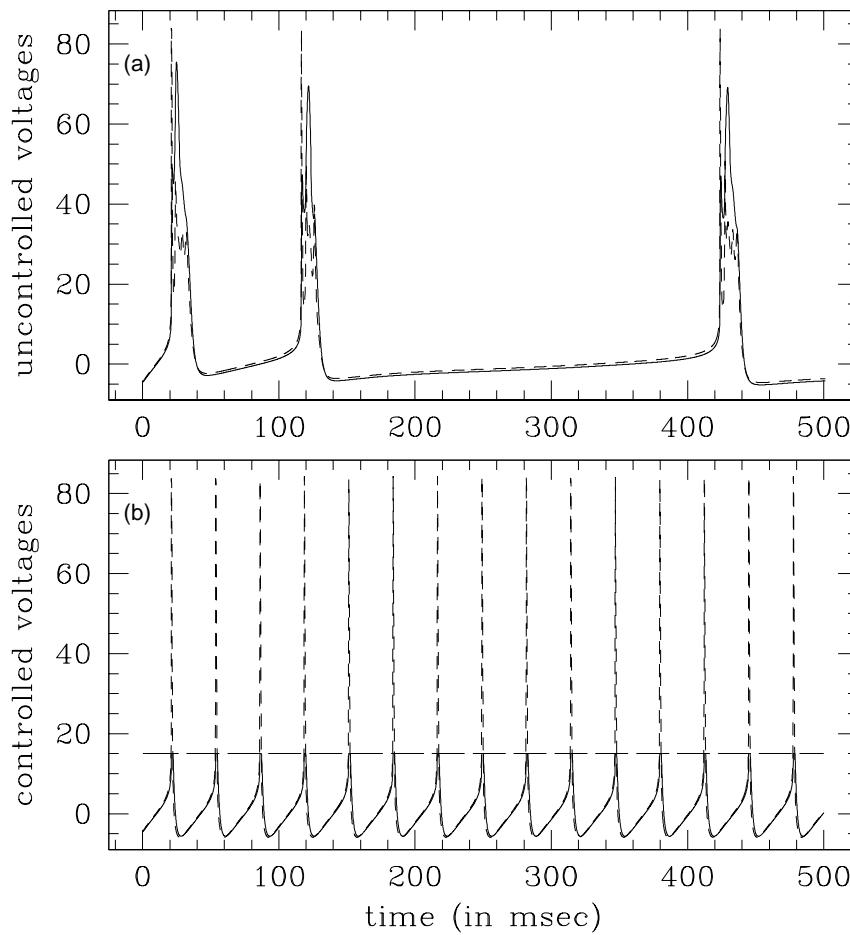


Fig. 1.6 The time evolution of the voltages V_s and V_d (in mV) for Pinsky-Rinzel neuron for the cases of: (a) uncontrolled neuron showing infrequent and irregular spiking behaviour; (b) the same neuron, with voltage V_d under threshold control, with threshold $V^* = 15$ mV. (Here $i_s = 1$ nA) Clearly the controlled neuron spikes at very regular intervals. The solid lines show V_d (—) and the dashed lines show V_s (- - -). The threshold voltage of $V^* = 15$ mV is shown by a dashed line (---).

constants α , and a destabilising force coefficient f . The dynamics of the beam is given by Eqn. 1.2 where the N -dimensional vector $\mathbf{x} = x_1, x_2, \dots, x_N$ gives the displacements of the elements, damping matrix G is gI (where I is the identity matrix) and coupling matrix A has elements $A_{mn} = -2\alpha + f$ for $m = n$, $A_{mn} = \alpha$ for $m = n \pm 1$, and $A_{mn} = 0$ otherwise [13].

In this array we now choose a few sites for threshold control. For effective control of the entire beam one needs to control a minimum of *two* sites [14]. *The amount*

of buckling tolerated determines the threshold \mathcal{T} . In order to effect control in this system we must implement a slight variation of the threshold method, with control action being triggered whenever $|x| > \mathcal{T}$. At the controlled sites here, one then imposes the following control condition: if $x < -\mathcal{T}$, the value of x is re-set to \mathcal{T} , and if $x > \mathcal{T}$, the value of x is re-set to $-\mathcal{T}$. This control action has the effect of “bending” the beam in the direction opposite to that of the buckling, thus having a local “straightening” effect on the array. Note that the value of the threshold \mathcal{T} can be made very small indeed, leading to arrays which are only slightly deformed.

Fig.1.7 shows the displacements of a representative element in an array of size 100, threshold controlled at only 2 sites. This displacement is compared to the uncontrolled situation, where the array deforms exponentially fast. Two cases are presented here: one with allowed buckling tolerance equal to 0.0001, i.e. the threshold $\mathcal{T} = 0.0001$, and the other with lower tolerance, $\mathcal{T} = 0.00001$. It is evident from the figure, that in the absence of control very weak environmental perturbations drive the system exponentially away from the desired configuration, while threshold control manages to achieve the goal, typical in smart matter applications, of preventing the beam from buckling more than the pre-assigned tolerance. Further note, that at the controlled sites the instances of control were quite infrequent. Typically, for an array with 2 controlled sites, with threshold $\mathcal{T} = 0.0001$, the control was ‘triggered on’ approximately one-tenth the number of times the variables were monitored for control.

Now the positions of the controlling sites are crucial for the success of this control. The controlling sites must span the array at roughly equidistant points. For instance, when 2 controllers are used in an array of size 100, one of these should be placed at some site between $i = 30$ and 40 and another between $i = 60$ and 70 . Here we are in fact exploiting the natural coupling of the system to influence a large neighbourhood with only a few sites. Thus very weak and infrequent control at very few sites manages to control the entire array.

It is thus evident that the threshold scheme can successfully control this extended nonlinear system. The present approach needs to implement the control on very few sites (minimum two). It utilizes no knowledge of the dynamics or system parameters, and also does not entail any computation in implementing the control. Further, there is no communication involved, as no site needs to know the state of its neighbours. One only needs to check if the value of the controlled variable at the controlled sites exceeds the user defined critical value or not. Since no communication or computation is involved, the control is very easily and simply implemented.

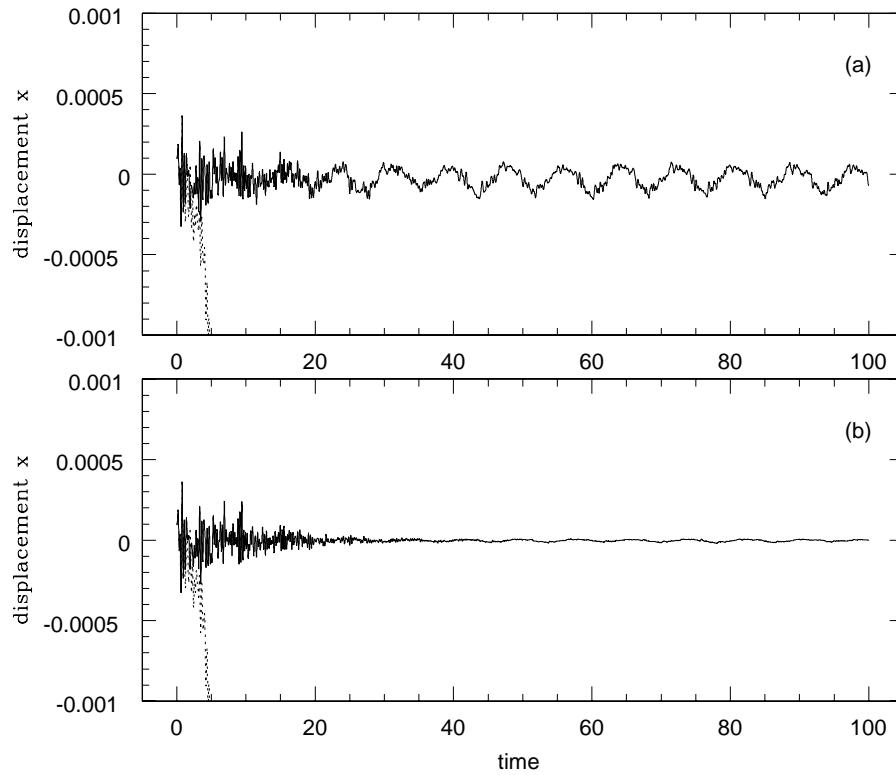


Fig. 1.7 Displacement of a representative element ($i = 20$) in an unstable elastic array w.r.t. time in the controlled (—) and uncontrolled case (....). The number of elements N in the chain is 100. Here we have placed 2 threshold controllers in the array at sites $i = 33$ and $i = 66$. The threshold value T is equal to (a) 0.0001 and (b) 0.00001. In the absence of control the system moves exponentially away from the steady state, while under control it manages to maintain the buckling within prescribed limits (namely, within 0.0001 and 0.00001 respectively).

1.2.4 Thresholding at Varying Intervals to obtain different Temporal Patterns

Now we discuss how stroboscopic threshold mechanisms can be effectively employed to obtain a wide range of stable cyclic behavior from chaotic systems, by simply varying the frequency of control [15]. For instance, consider the action of infrequent threshold control on the chaotic Lorenz-like attractor given by Eqns.1.1, using the three parameters corresponding to the coherently pumped far-infrared ammonia

laser system, obtained by detailed comparison with experiments [9]: $\sigma = 2$, $r = 15$ and $b = 0.25$.

Consider the particular case of the threshold mechanism imposed on the z variable. The stroboscopic threshold action occurs at an interval of $n_c \delta t$. Fig. 1.8 shows the different temporal patterns obtained when the threshold is fixed at $z^* = 1$ and the n_c is increased.

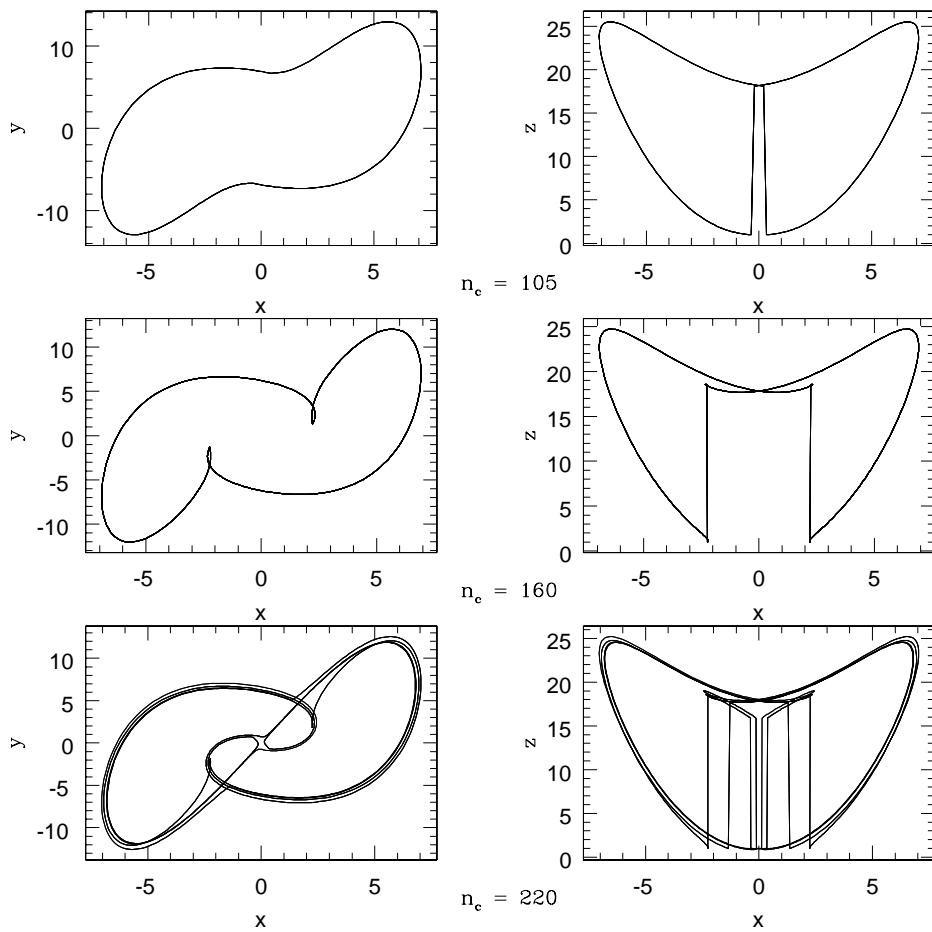


Fig. 1.8 The chaotic laser-Lorenz system under threshold control of variable z , with threshold value $z^* = 1$. The control acts at intervals of $n_c \times \delta t$, with $\delta t = 0.01$. Here $n_c = 105, 160, 220$. The controlled cycles in x - y space and x - z space are displayed. Notice the period doubling of the cycles as control interval n_c is increased.

In [15], a large range of numerical evidence is reported, with the frequency of control spanning three orders of magnitude, to show that stroboscopic threshold action of any variable in this multi-dimensional chaotic system successfully yields

regular temporal patterns, displaying a wide variety of periods and geometries. In fact, the interval of control may be very large in many cases and still lead to very effective control onto simple limit cycles. So varying the interval of threshold control offers flexibility and cost-effectiveness in regulating chaotic systems onto different cyclic patterns.

1.3 Experimental verification

Here we review the experimental verification of thresholding as a versatile tool for efficient and flexible chaos control. We demonstrate the success of the technique in rapidly controlling different chaotic electrical circuits, including a hyperchaotic circuit, onto stable fixed points and limit cycles of different periods, by thresholding just one variable. The simplicity of this controller entailing no run-time computation, and the ease and rapidity of switching between different targets it offers, suggests a potent tool for chaos based applications.

1.3.1 Controlling a Circuit Realisation of Jerk Equations

The first experimental set-up is a realisation of nonlinear third order ordinary differential equations (ODE), a form known in literature as Jerk equations:

$$\frac{d^3x}{dt^3} + A\frac{d^2x}{dt^2} + \frac{dx}{dt} = G(x) \quad (1.3)$$

where $G(x)$ is a piecewise linear function: $G(x) = B|x| - C$ with $B = 1.0, C = 2.0$ and $A = 0.6$ [16]. The circuit realisation of the above uses resistors, capacitors, diodes and operational amplifiers as shown in Fig.1.9. The implementation involves three successive active integrators to generate $\frac{d^2x}{dt^2}$, $\frac{dx}{dt}$ and x from $\frac{d^3x}{dt^3}$, coupled with a nonlinear element that generates $G(x)$ and feeds it back to $\frac{d^3x}{dt^3}$.

Now we implement the threshold mechanism on variable x , i.e. whenever $x > x^*$, x is clipped to x^* . A precision clipping circuit [17] as depicted in the dotted box in Fig.1.9 is employed for threshold control. We have chosen component values for the control circuit to be: [op-amp = μ A 741, diode = IN4148, load resistor = $1k\Omega$ and threshold reference voltage = V , which sets the x^*].

Fig.1.10a displays the uncontrolled attractor and Fig.1.10b-d shows some representative results of the threshold action on this chaotic system for a range of threshold values x^* ($x^* < 2.4$). It is clear that the mechanism manages to yield cycles of varying periodicities. Further, detailed comparison shows *complete agreement between our experimental results and our numerical simulation results*.

So the single thresholded variable x has the ability to drag the rest of this 3-dimensional system to regular dynamical behavior. The characteristics of the controlled states can be easily varied by just changing the threshold x^* (see Table1.2). Also note that simply setting the threshold beyond the bounds of the attractor gives

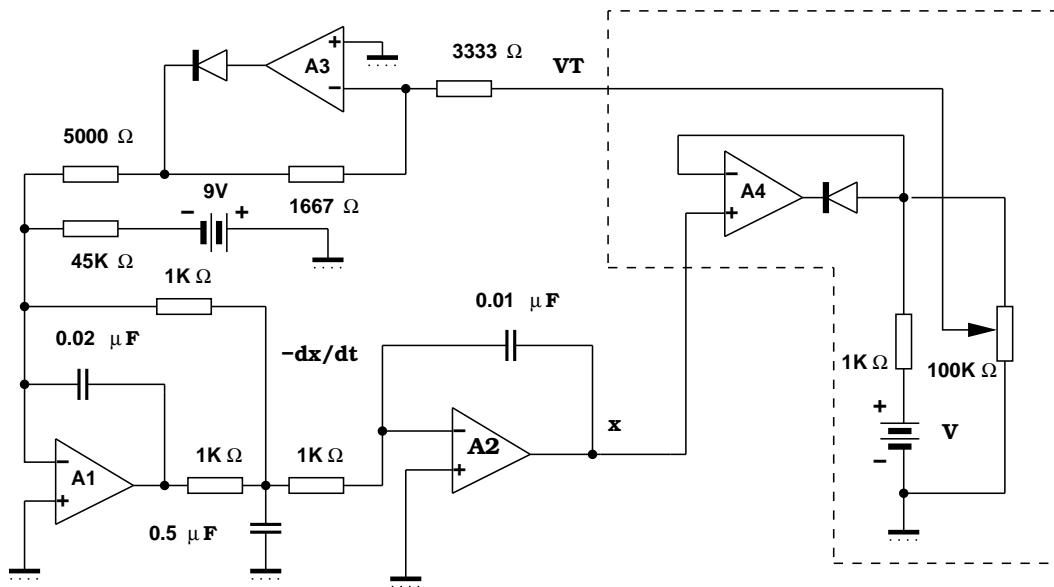


Fig. 1.9 A general circuit for solving Eqn.1.3 using a nonlinear feedback element $G(x) = B|x| - C$. The precision clipping control circuit is shown in the dotted box. Here V_T corresponds to the threshold controlled signal.

Table 1.2 Thresholding a third-order nonlinear system

| Threshold Value | Nature of Controlled Orbit |
|-----------------------|----------------------------|
| $x^* < -2.00$ | fixed point |
| $-2.00 < x^* < 1.477$ | Period 1 Cycle |
| $1.477 < x^* < 2.242$ | Period 2 Cycle |
| $2.242 < x^* < 2.321$ | Period 4 Cycle |
| $2.321 < x^* < 2.325$ | Period 8 Cycle |
| $2.325 < x^* < 2.331$ | Period 16 Cycle |

Threshold ranges (in V) vs. periodicity of the controlled cycle, for the chaotic circuit described by Eqn.1.3.

back the original dynamics.

The control transience is very short here (typically of the order of 10^{-3} times the controlled cycle). This makes the control practically instantaneous. The underlying reason for this is that the system does not have to be close to any particular unstable fixed point, as in OGY based schemes, before implementing control. Once a specified state variable exceeds the threshold it is caught immediately in a stable orbit.

The changes in state effected by thresholding, namely $(x - x^*)$ when $x > x^*$, are typically small (as adjustments are made just after x crosses x^*). Further for higher periods the controlling action is infrequent and occurs for short intervals in

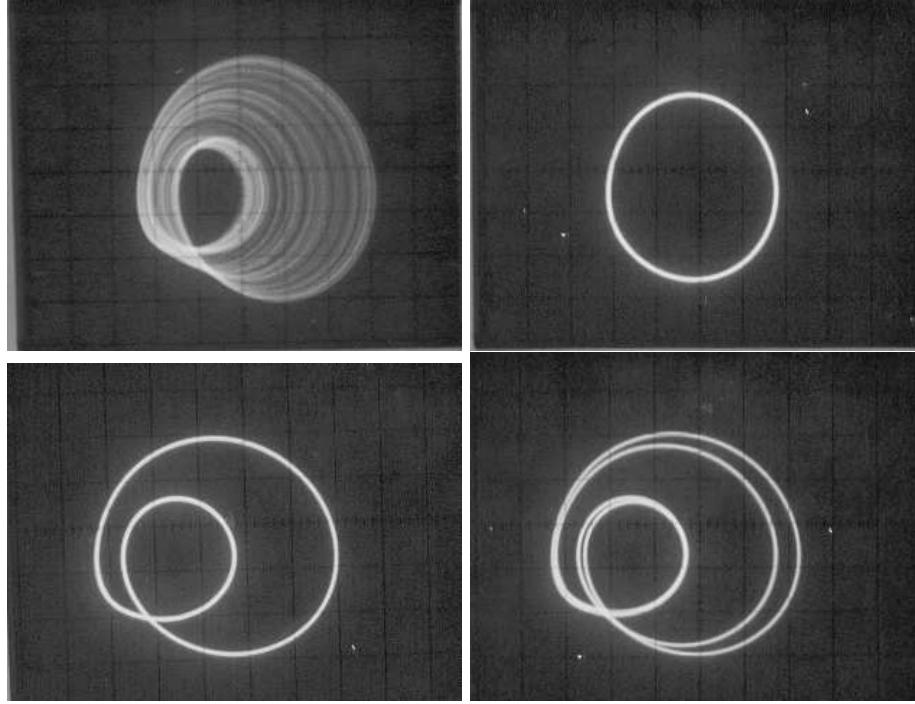


Fig. 1.10 Attractors in the $x - \dot{x}$ plane: (a) the uncontrolled chaotic system obtained from the circuit realisation of Eq.1.3 (upper left box); (b) period 1 cycle obtained when $x^* = 1$ V (upper right box); (c) period 2 cycle obtained when $x^* = 2$ V (lower left box) and (d) period 4 cycle obtained when $x^* = 2.1$ V (lower right box).

every controlled cycle. For instance to control to a 16-cycle with $x^* = 2.327$, the thresholding is operational for only ~ 0.22 msec in an interval of 50 msec.

1.3.2 Controlling Chua's Circuit

Now we consider a realisation of the double scroll chaotic Chua's attractor given by the following set of (rescaled) 3 coupled ODEs [18]

$$\dot{x} = \alpha(y - x - g(x)) \quad (1.4)$$

$$\dot{y} = x - y + z \quad (1.5)$$

$$\dot{z} = -\beta y \quad (1.6)$$

where $\alpha = 10$. and $\beta = 14.87$ and the piecewise linear function $g(x) = bx + \frac{1}{2}(a - b)(|x + 1| - |x - 1|)$ with $a = -1.27$ and $b = -0.68$. The corresponding circuit component values are: [$L = 18mH$, $R = 1710\Omega$, $C_1 = 10nF$, $C_2 = 100nF$, $R_1 = 220\Omega$, $R_2 = 220\Omega$, $R_3 = 2.2k\Omega$, $R_4 = 22k\Omega$, $R_5 = 22k\Omega$, $R_6 = 3.3k\Omega$, $D = \text{IN4148}$, $B_1, B_2 = \text{Buffers}$, $\text{OA1} - \text{OA3} : \text{opamp } \mu\text{A741}$]. Note that the circuit of

Fig.1.11 is the ring structure configuration of the classic Chua's circuit [18]-[19]. The uncontrolled attractor from this system is displayed in Fig.1.12a.

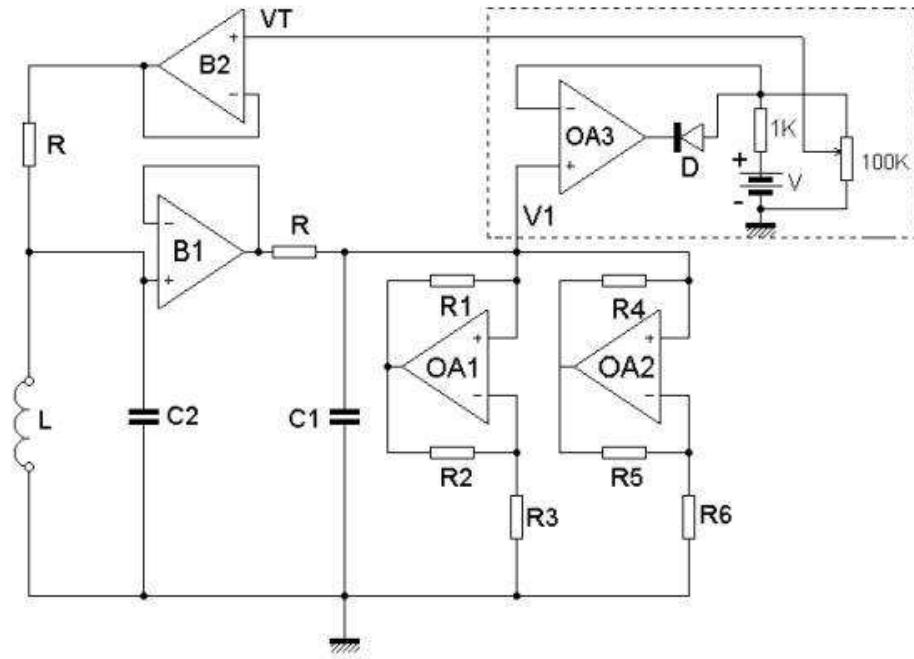


Fig. 1.11 Chua's chaotic circuit with threshold level controlling circuit (shown in the dotted box). Here V_T is the threshold controlled signal.

Now we implement an even more minimal thresholding. Instead of demanding that the x variable be reset to x^* if it exceeds x^* we only demand this in Eqn.1.5. This has very easy implementation, as it avoids modifying the value of x in the nonlinear element $g(x)$, which is harder to do. So then all we do is to implement $\dot{y} = x^* - y + z$ instead of Eqn.1.5, when $x > x^*$, and there is no controlling action if $x \leq x^*$. In the circuit the voltage V_T corresponds to x^* (see Fig.1.12). The resulting controlled orbits with respect to threshold x^* is given in Fig.1.13b-d ($x^* < 2.7$). So the threshold control works on the system rapidly and can control to a wide range of temporal behaviours (see Table1.3).

1.3.3 Controlling Hyperchaos

Now we demonstrate the method on a hyperchaotic electrical circuit. This constitutes a stringent test of the control method [20] since the system possesses more than one positive lyapunov exponent, and so more than one unstable eigendirection has to be reigned in by thresholding a single variable. In particular we consider the

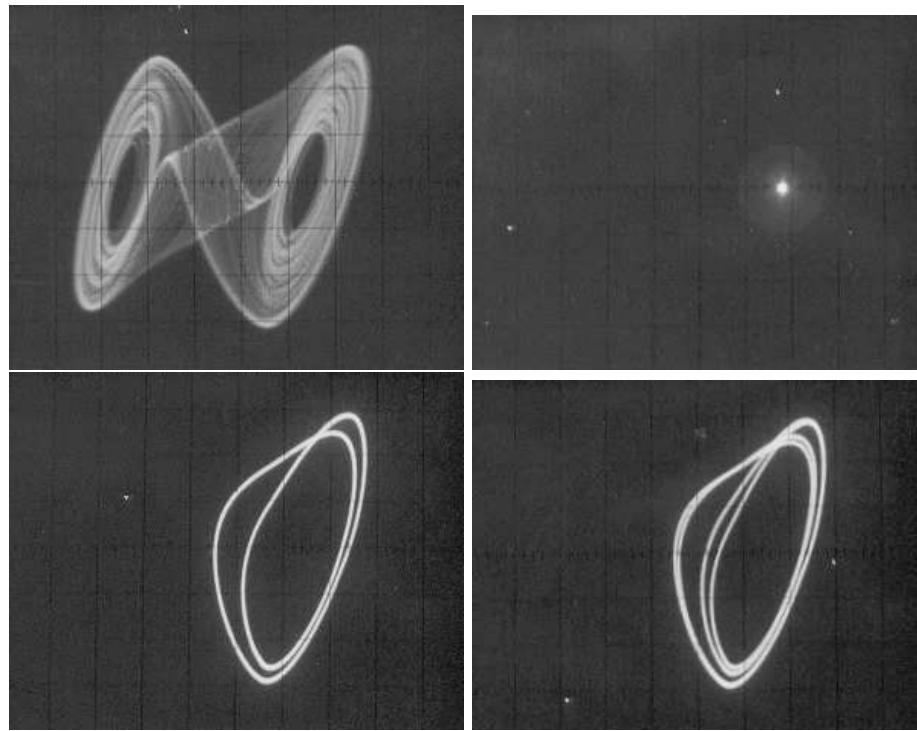


Fig. 1.12 Attractors in the $V_1 - V_2$ plane, corresponding to the $x - y$ plane of Eqns.1.4-1.6: (a) uncontrolled chaotic attractor (upper left box); (b) fixed point obtained when $x^* = 1.8$ V (upper right box); (c) period 2 cycle obtained when $x^* = 2.7$ V (lower left box) and (d) period 4 cycle obtained when $x^* = 2.71$ V (lower right box).

Table 1.3 Thresholding the Chua's circuit

| Threshold Value | Nature of Controlled Orbit |
|-------------------------|----------------------------|
| $x^* < 1.84375$ | fixed point |
| $1.84375 < x^* < 2.235$ | Period 1 Cycle |
| $2.235 < x^* < 2.258$ | Period 2 Cycle |
| $2.258 < x^* < 2.264$ | Period 4 Cycle |
| $2.264 < x^* < 2.265$ | Period 8 Cycle |
| $2.265 < x^* < 2.2653$ | Period 16 Cycle |

Threshold ranges (in V) vs. periodicity of the controlled cycle, for the chaotic system given by Eqns.1.4-1.6.

realisation of four coupled nonlinear (rescaled) ordinary differential equations of the

form

$$\dot{x}_1 = (k-2)x_1 - x_2 - G(x_1 - x_3) \quad (1.7)$$

$$\dot{x}_2 = (k-1)x_1 - x_2 \quad (1.8)$$

$$\dot{x}_3 = -x_4 + G(x_1 - x_3) \quad (1.9)$$

$$\dot{x}_4 = \beta x_3 \quad (1.10)$$

where

$$G(x_1 - x_3) = \frac{1}{2}b[|x_1 - x_3 - 1| + (x_1 - x_3 - 1)]$$

with $k = 3.85$, $b = 88$ and $\beta = 18$ [21]. The circuit realisation of the above is displayed in Fig.1.14, with component values: [$L = 18$ mH, $C_2 = 68$ nF, $R = 1.8k\Omega$, $C = 68$ nF, $R_1 = 2.8k\Omega$, $R_2 = 1k\Omega$, $D_1 = \text{IN4148}$]. Fig.1.15a displays the (uncontrolled) hyperchaotic attractor resulting from this circuit, and it is characterised by two maximal positive lyapunov exponents: $\lambda_1 = 0.13$, $\lambda_2 = 0.05$.

Again we implement a *partial* thresholding on variable x_3 : whenever $x_3 > x^*$ in the system, $G(x_1 - x_3)$ in Eqn.1.7 becomes $G(x_1 - x^*)$, i.e. we have $\dot{x}_1 = (k-2)x_1 - x_2 - G(x_1 - x^*)$, while Eqns.1.8-1.10 are unchanged. When $x_3 \leq x^*$ there is no action at all. A precision clipping circuit [17] as depicted in the dotted box in Fig.1.13 is employed for the above scheme, which is even simpler to implement than thresholding x_3 throughout the system. We have chosen component values for the control circuit to be: [op-amp = μA741 , diode (D) = IN4148 or IN34A, series resistor $R_s = 1k\Omega$ and threshold reference voltage = V , which sets the x^*].

Both our experiments and our numerical simulations (which are in complete agreement) show that this scheme successfully yields regular *stable* cycles under a very wide range of thresholds. A representative example with threshold set at 0 V is displayed in Fig. 1.14b, which shows the controlled cycle in the $V_1 - V_2$ plane, which corresponds to the rescaled $x_1 - x_3$ plane of Eqns.1.7-1.10.

So it is evident that a *single thresholded variable has the ability to clip the full 4-dimensional hyperchaotic system to regular dynamical behavior* (see Figs.1.15-1.16 for some examples of the geometries of the controlled orbits). Thus the period and geometry of the controlled states can be easily varied by setting x^* in different windows. For instance, thresholding at 0 V yields a 1 T attractor (with respect to the x_1 variable), while thresholding at 0.3 V yields period 3 T, 0.32 V yields period 8 T, 0.33 V yields period 5 T and 0.35 V yields period 13 T.

1.4 Conclusions

It is clearly evident then through analytical results, numerics and experiments that threshold control is a powerful, efficient and robust technique to extract a wide range of regular behaviors from a chaotic system. The method involves no adjustment of parameters, but merely the manipulation and re-setting of *one* state variable,

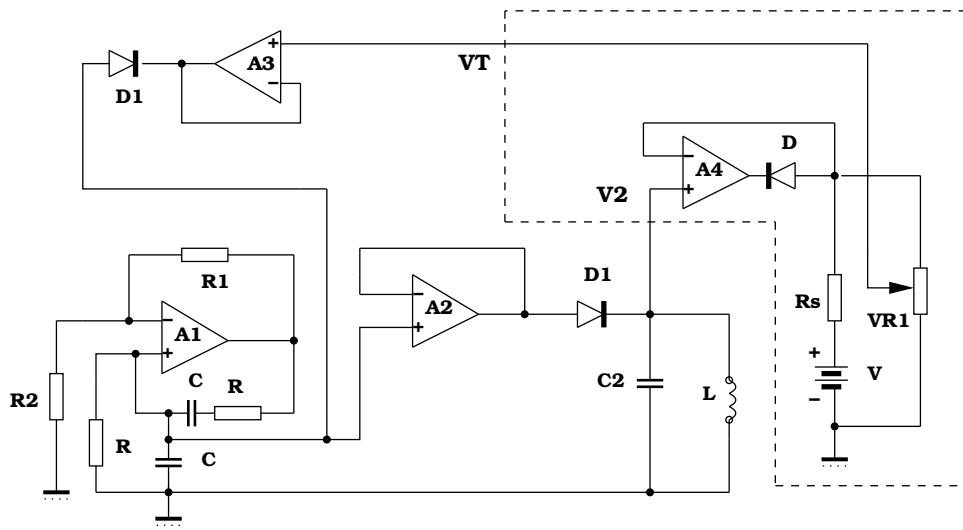
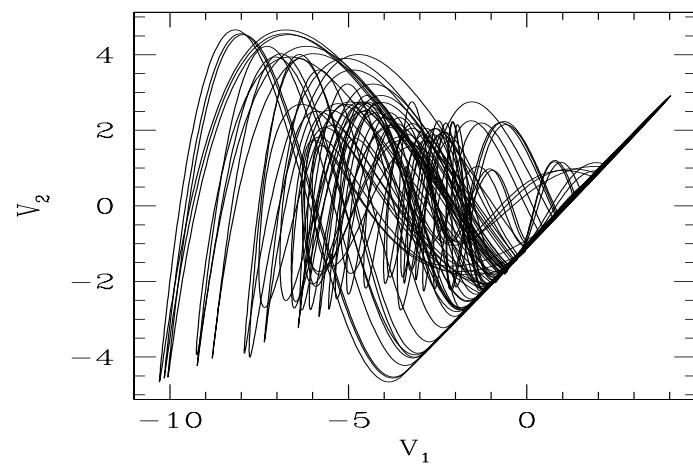


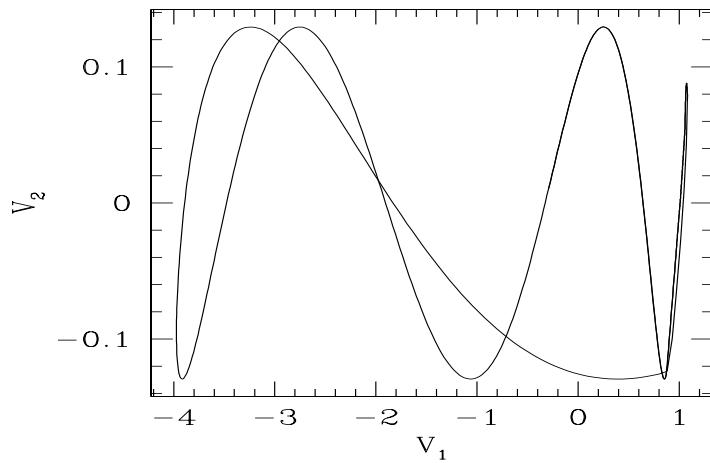
Fig. 1.13 Circuit implementation of Eqns.1.7-1.10, with the precision clipping control circuit in the dotted box. V_T is the threshold controlled signal.

even in hyperchaotic systems possesing more than one unstable eigendirections. The richness of chaotic dynamics is an useful feature here, as it ensures that the dynamics can be “clipped” or “truncated” to many different kinds of patterns, i.e. under threshold mechanism the chaos yields a very wide variety of stable dynamical behaviour. In fact for one-dimensional systems one can obtain exact results for the effect of thresholding, and these theoretical results have been completely verified in electronic circuit experiments, as well as in arrays of 1-d systems implemented in CMOS based VLSI design [5]-[6].

Threshold control is especially useful in the situation where one wishes to design components with the ability to *switch between patterns*, operating as potential *pattern generators*. Basically it can provide a *look up table* for a “*library of patterns*” for very swift control. So the simplicity and versatility of the threshold controller has much potential utility for chaos-based applications, such as chaos computing [22]-[23] and communications [24].



(a)



(b)

Fig. 1.14 (a) Uncontrolled hyperchaotic attractor, (b) controlled attractor for threshold = 0 V, in the $V_1 - V_2$ plane, corresponding to the $x_1 - x_3$ plane of Eqns. 1.7-1.10.

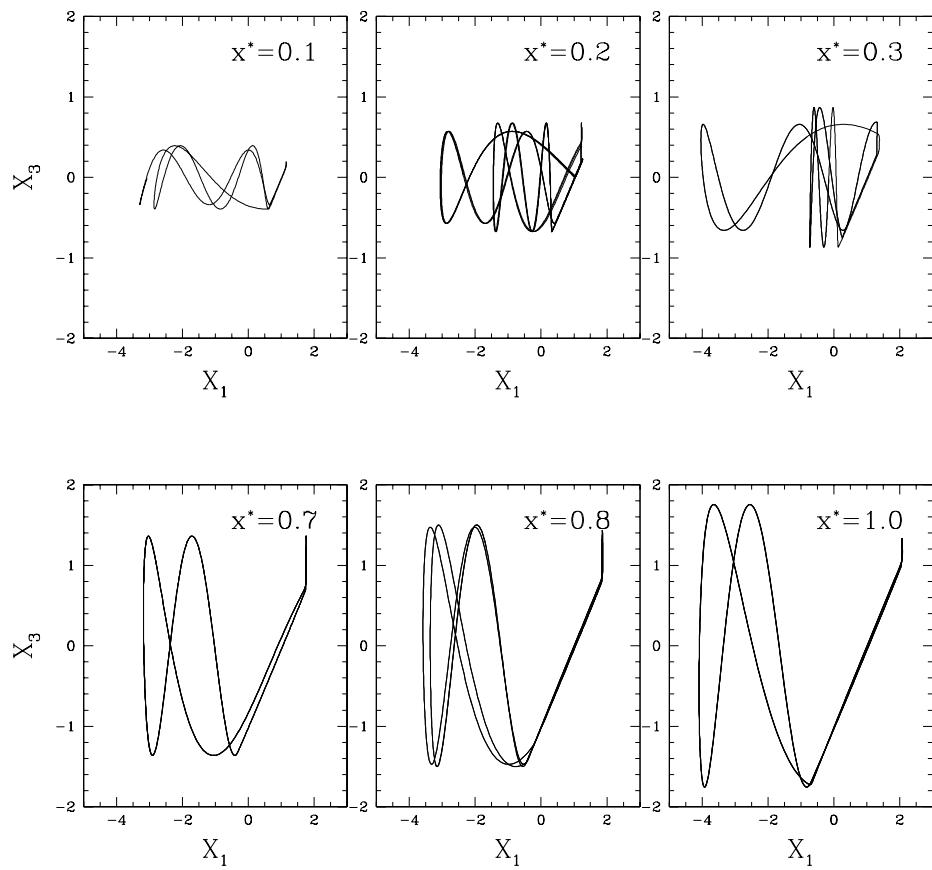


Fig. 1.15 Controlled attractors in the $x_1 - x_3$ plane, obtained from the hyperchaotic system by thresholding the x_3 variable in Eqn.1.7, with threshold values: (i) $x^* = 0.1$, (ii) $x^* = 0.2$, (iii) $x^* = 0.3$, (iv) $x^* = 0.7$, (v) $x^* = 0.8$, and (vi) $x^* = 1.0$,

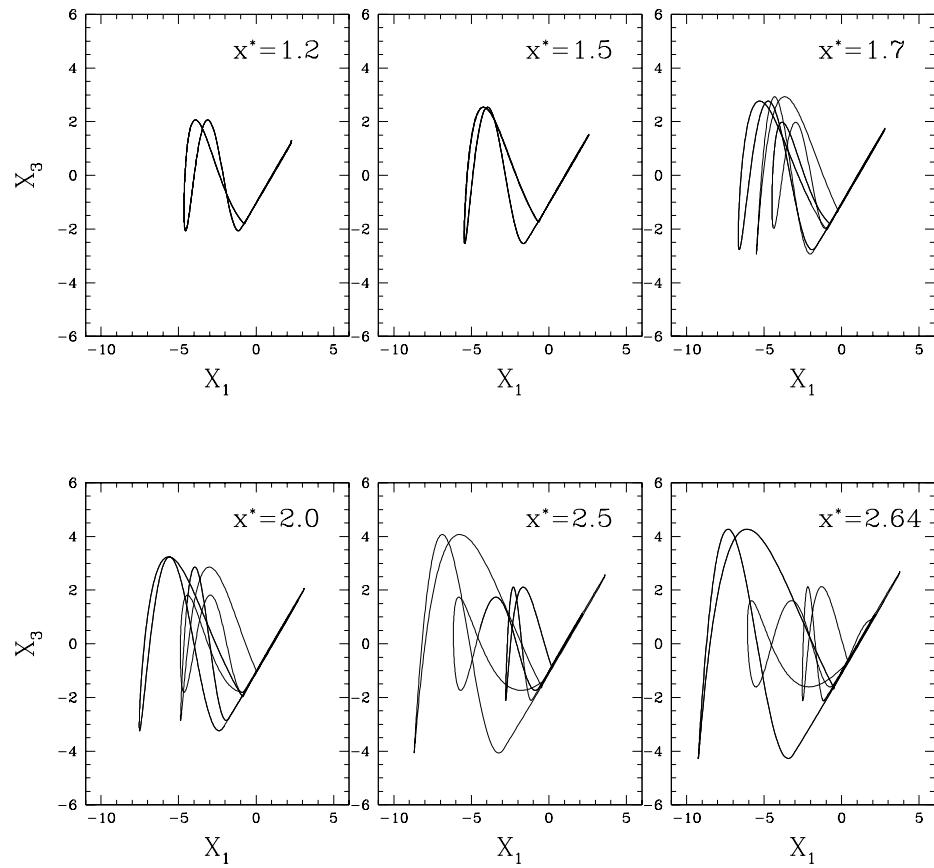


Fig. 1.16 Controlled attractors in the $x_1 - x_3$ plane, obtained from the hyperchaotic system by thresholding the x_3 variable in Eqn.1.7-1.10, with threshold values: (i) $x^* = 1.2$, (ii) $x^* = 1.5$, (iii) $x^* = 1.7$, (iv) $x^* = 2.0$, (v) $x^* = 2.5$ and (vi) $x^* = 2.64$,

Bibliography

- [1] S. Sinha and D. Biswas, Phys. Rev. Letts. **71** (1993) 2010; S. Sinha, Phys. Rev. E, **49** (1994) 4832; Phys. Letts. A, **199** (1995) 365; Int. Jour. Mod. Phys. B **9** (1995) 875; S. Sinha, in *Recent Developments in Nonlinear Systems* (Narosa, 2002).
- [2] E. Ott, C. Grebogi and J. Yorke, Phys. Rev. Letts. **66** (1990) 1196.
- [3] Note that a similar method was developed independently by Wagner and Stoop, and goes by the name *Limiter Control*: see for instance, C. Wagner, R. Stoop, *Phys. Rev. E* **63** 017201 (2001); C. Wagner, R. Stoop, *J. Stat. Phys.*, **106** 97-107 (2002).
- [4] L. Glass and W. Zheng, Int. J. Bif. and Chaos, **4** (1994) 1061.
- [5] K. Murali and S. Sinha, *Phys. Rev. E* **68** (2003) 016210
- [6] D. Atutti, T. Morie, and K. Aihara, *IEEE Int. Workshop on Nonlinear Dynamics of Electronic Systems* (NDES 2007), pp. 145-148 (2007)
- [7] This method may be better suited for optical and electrical systems which lend themselves easily to limiters, while it may be harder to implement limiters in chemical and mechanical systems.
- [8] A. Miliotis, S. Sinha and W.L. Ditto, Proceedings of *IEEE Asia-Pacific Conference on Circuits and Systems* (APCCAS06), Singapore (2006) pp. 1843-1846; A. Miliotis, S. Sinha and W.L. Ditto, *Int. J. of Bif. and Chaos* (to appear)
- [9] U. Hubner, N.B. Abraham and C.O. Weiss, Phys. Rev. A, **40** (1989) 6354 (and references therein)
- [10] *Patterns, Information and Chaos in Neuronal Systems*, ed. B.J. West (World Scientific, 1993) and references therein.
- [11] Pinsky, P. and Rinzel, J., Journ. Comp.Neuroscience, **1** (1994) 39; Lindner, J.F. and Ditto, W.L., Proceedings of the 3rd Technical Conference on Nonlinear Dynamics (Chaos) and Full-Spectrum Processing, Mystic, Connecticut, (AIP Conference Proceedings, **375** (1996) 709.
- [12] S. Sinha and W. Ditto, Phys. Rev. E **63** (2001) 056209.
- [13] T. Hogg and B. Huberman, Smart Materials and Structures, **7** (1998) R1, and references therein.
- [14] The very successful smart matter controls based on markets, i.e. multiagent methods of control [13], depend on the information from many sites, with sensors located at various points.
- [15] S. Sinha, Phys. Rev. E, **63** (2001) 036212
- [16] J.C. Sprott, Phys. Letts. A **266** (2000) 19.
- [17] Maddock, R.J. and Calcutt, D.M., *Electronics: A Course for Engineers*, Addison Wesley Longman Ltd., (1997) p. 542.
- [18] Dimitriev, A.S. et al, J. Comm. Tech. Electronics, **43** (1998) 1038.

- [19] Lakshmanan, M. and Murali, K., *Chaos in Nonlinear Oscillators: Controlling and Synchronisation*, World Scientific, Singapore, 1996.
- [20] M. Ding, *et al*, *Chaos* **7** (1997) 644; L. Yang *et al*, *Phys. Rev. Letts.* **84** (2000) 67.
- [21] Murali, K. *et al*, *Proc. 6th Experimental Chaos Conference*, Potsdam, July 17-22, 2001.
- [22] Sinha, S. and Ditto, W.L. *Phys. Rev. Lett.* **81** (1998) 2156; Sinha, S. and Ditto, W.L. *Phys. Rev. E* **59** (1999) 363; T. Munakata *et al*, *IEEE Trans. Circ. and Systems* **49** (2002) 1629; Sinha, S. *et al*, *Phys. Rev. E* **65** (2002) 036214; Sinha, S. *et al* *Phys. Rev. E* **65** (2002) 036216.
- [23] K. Murali, Sudeshna Sinha and W.L. Ditto, *Phys. Rev. E* **68** (2003) 016205; K. Murali, S. Sinha and W.L. Ditto, *Int. J. Bif. and Chaos (Lett)* **13** (2003) 2669; M.R. Jahed-Motlagh, B. Kia, W.L. Ditto and S. Sinha, *Int. J. Bif. and Chaos* **17** (2007) 1955; W.L. Ditto and S. Sinha, *Phil. Trans. of the Royal Soc. of London* **364 A** (2006) 2483.
- [24] Hayes, S. *et al*, *Phys. Rev. Lett.* **70** (1993) 3031.