

Stabilization of chaotic systems using the BOXES methodology

D.W. Russell

*Penn State Great Valley, Malvern, PA 19355,
USA*

ABSTRACT

The detection and reporting of chaotic tendencies in real-time, real-world systems is increasing as careful analyses of processes are performed. This may well be in part due to the re-emergent interest in chaotic systems and some of the new possibilities that chaotic programming affords. This paper shows first how the algorithm for the measurement of the *information dimension* of a fractal has marked similarity to the BOXES paradigm. Second, the paper illustrates how a BOXES-type automaton can be constructed to attach to a simulation of the well-known chaotic system, described by the Lorentz Equations, and how it can be taught to stabilize the motion.

INTRODUCTION

There is an increase in interest in the whole notion of chaos and how it relates to real-life systems. Schelberg [1] uses the New York City taxi-cab system (or lack of it) as an example of a chaotic system and how one might approach a solution to it that would dictate the cruising and waiting patterns of cab-drivers. There is more than a hint of chaos in Edelman's [2] "rain-forest" approach to modelling the brain. However, in this paper the Lorentz Equations are used to provide the chaotic environment for the study and to demonstrate the principle. The Lorentz System comprises three simultaneous, non-linear, first-order differential equations (1)-(3).

$$dx/dt = \sigma.(y - x) \tag{1}$$

$$dy/dt = \rho.x - y - x.z \tag{2}$$

$$dz/dt = x.y - \beta. z \tag{3}$$

296 Artificial Intelligence in Engineering

σ , ρ and β are parameters that are related to the physical system, for example, fluid flow and x , y and z are the state variables. In the spirit of prior study of the BOXES methodology (Russell [3]) the actual system mathematics are deemed unimportant so the familiar values of 10, 28 and $8/3$ are used throughout for σ , ρ and β respectively. It is also noteworthy that the initial conditions for the state variables are to be randomly selected and that their values do affect the trajectory of the ensuing motion of the system. This is a commonly used criterion for identifying potentially chaotic phenomena. The usual representation of such systems is a phase-plot, in which two of the variables (eg. X & Y) are plotted against each other rather than against time. When the variables are sampled the phase-plot is called a Poincaré map. Figure 1 shows a Z - X Poincaré map for the Lorentz equations described above. The data is generated by numerically integrating Equations (1)-(3), with values sampled every .01 seconds.

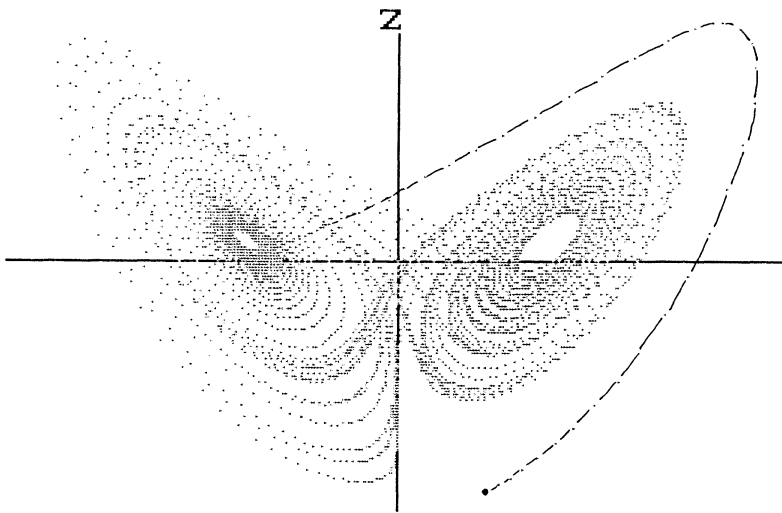


Figure 1: Poincaré Map for the Lorentz Equations

Watching the pattern develop during the simulation, it becomes apparent that the system is hunting (at least) two solutions located at the "eyes" of the "butterfly wings." The state variables seem attracted to the solution space yet when they approach an intersection with it they careen away with toroidal or ellipsoidal orbits. However, the system is seen to remain within distinct boundaries, tracing out an endless "saddle" shape. This type of behavior is also a characteristic of chaotic motion.

STRANGE ATTRACTORS & FRACTAL DIMENSIONS

The solution spaces, in which the system could rest with no motion, are called "Strange Attractors" and are visually obvious in most chaotic system graphics; for example, the "eyes on the butterfly wings" shown in Figure 1. That these areas exist is obvious, but there is a need to express them numerically in such a way as to include not only some notion of size but also confirmation of the expectation of chaotic motion. Moon [4] defines *fractal dimensions* as quantitative measures of the strangeness of the attractor and the following is abstracted from that text.

There are several measures of fractal dimension:

- i. *capacity* -which involves a calculation based on the geometry of the Poincaré map only.
- ii. *pointwise* -in which counts of the number of Poincaré points that lie inside elemental cubes in the phase plane are considered.
- iii. *correlation* -which is similar to the pointwise calculation, but richer in its treatment of every point on the map.
- iv. *information* -in which the frequency of entry into any region of state space is used.

It is curious, but not altogether unexpected, that the four dimensions yield similar values for any given map and that the values are *non-integer* which is an accepted indicator of the existence of fractals.

On inspection of a Poincaré map, it becomes apparent that the chaotic motion, in defiance of its colloquial definition, contains conservative, bounded information. It is because of this that the information dimension was selected for this study. Equations (4) & (5) detail how the information dimension, di , is calculated:

$$Pi = Ni / N \quad (4)$$

$$di = \lim_{\epsilon \rightarrow 0} (\sum Pi \cdot \log Pi) / \log (1/\epsilon) \quad (5)$$

Where, ϵ is a small quantization variable that divides the state space into *small* cubic cells and about which more will be said in the next section. Ni is the number of entries of the system into any given state, i , so Pi is the probability of any of the N points on the map lying within that state. For example, in Figure 1 the calculated value for di was 1.764.

QUANTIZATION & THE BOXES METHODOLOGY

The algorithm that implements the BOXES methodology utilizes has several distinct steps.

The state variables are *first* normalized using known values of the allowed maximum and minimum for each of the n variables. If these values are y_{xi} and y_{mi} respectively, for any variable y_i , Equation (7) yields the normal value, $\hat{y}_i$.

$$\hat{y}_i = (y_i - y_{mi}) / (y_{xi} - y_{mi}) \quad (7)$$

Where, $0 \leq \hat{y}_i \leq 1$.

The *second* step in the process is to allocate an integer region number, r_i , for each of the normalized variables. In traditional BOXES systems, each variable has a set of unique boundary values that delineate these zones, but recalling the quantity ϵ discussed above for this application, the normalized state variables are to be all equally divided into $1/\epsilon$ cells.

Equation (8) is a simple, fast, algorithm for the allocation of r_i values that include both edges of the normal set.

$$\begin{aligned} \text{IF } \hat{y} = 0 \text{ THEN } r_i &= 1 \\ \text{ELSE IF } \hat{y} &= 1 \text{ THEN } r_i = 1/\epsilon \\ \text{ELSE } r_i &= \text{INT} (\hat{y} / \epsilon) \end{aligned} \quad (8)$$

The *third* step is to now combine the n values of r_i into one unique state number, m . It is simple to show by integer arithmetic that Equation (9) performs this function provided all values of r exist between 1 and $1/\epsilon$.

$$m = r_1 + \sum_{k=2}^{k=n} (1/\epsilon)^{k-1} \cdot (r_k - 1) \quad (9)$$

The BOXES methodology uses a signature table, Y , in which to store its decision values. Access to any particular control value, u , is then a simple table look-up process, using the state number, m , as its index (see Equation (10).)

$$u = Y (m) \quad (10)$$

Initially the signature table is randomly seeded with decision values and updated after each and every run by a post-processing, learning, algorithm.

This algorithm is designed to seek to optimize some property or function of the system as a whole and uses a statistical inferencing process. For example, in the *pole and cart* problem (Michie & Chambers (5)), the task is to prolong the pole balancing act before a failure occurs by learning where to place reversals in motor direction in the state space as represented by the signature table. BOXES learning algorithms use data acquired during the most recent run and combine it with historical values at the individual cell and global levels. The algorithm produces two factors that represent the "worth" or "strength" of both decisions for each cell and preferentially updates the stored value before proceeding to the next run. As the system matures, it has been shown (Russell & Rees (6)) that the signature table becomes trained to exhibit intelligent process control. Furthermore, the process is dynamic and adapts to changes in the system, such as wear and failure. It is suggested that the information dimension defined above may contain a measure of the "merit" of a chaotic system in this regard.

Before such a paradigm can be adopted for the Lorentz system, the value of u must be logically connected to the equations and a learning algorithm constructed. The next two sections are devoted to handling these issues. Because this paper is of a forensic nature the practical implementation of such an attachment is omitted and only the mathematics considered.

SIGNATURE TABLE ATTACHMENT

The terms in the Lorentz equations that make the system unstable are the $x.z$ and $x.y$ products in Equations (2) and (3) respectively. These elements may be considered as "virtual driving functions" and as such makes them suitable candidates for a BOXES-type attachment. In a prior study, Russell (7) successfully attached the BOXES-control variable, u , to the PID tuning parameters of a second-order system and given the task of learning to suppress transient behavior without loss of response time or knowledge of the system dynamics.

To affect the motion dramatically, it is suggested that Equation (2) be modified to become Equation (6) in the simulation.

$$dy/dt = \rho x - y - u.x.z \quad (6)$$

Traditionally, the value of the BOXES-variable, u , is binary, but in this case it became evident that a more detailed study of the two possible values in the control matrix was required. When the value of u is unity, the system behaves in its original state, so it became important to design a value for u that would maintain the system integrity, while enforcing the stabilizing regimen.



For the purposes of this paper, the "other" BOXES matrix value, *f*, was selected in the range 0.9-0.98 and produced startling reductions in the chaotic phase orbits, while preserving the basic dynamics of the system.

LEARNING IN THE CHAOTIC ENVIRONMENT

As stated above, the calculation of the information dimension involves the allocation of a state number to each Poincaré point and the keeping of a running count of each entry into that particular state. Intuitively, if the points can be concentrated into a fewer states, then the resultant orbit will be less chaotic. To verify this, the algorithm for fractal dimension was exercised with preset count values. Table I shows the results of these tests for a generated total point count of 10000.

TEST	DESCRIPTION	VALUE of <i>di</i>
1.	Chaotic Motion...Figure 1.	1.764
2.	1 cell with all 10000 points	0
2.	10 cells with 1000 points each	1
3.	100 cells with 100 points each	2
4.	1000 cells with 10 points each	3

Table I. Fractal Dynamics Calculations.

It is apparent that if the data were to be evenly distributed over a few states then the fractal dimension could become an integer and the more points in each cell the closer the dimension is to zero.

The only real measurement of performance available in the Lorentz system is the fractal's information dimension, *di*. Let the run *merit* be expressed as *rdi*, the reciprocal of *di*, and let it be used in the learning process. This procedure is similar to other BOXES automata decision engines.

System Merit Calculation.

When the current run ends, the automaton must incorporate the new data, *rdi* into the historically weighted system merit, *SYM*. Given that: *NUR* is a measure of the automaton's past experience level and *RFD* is a weighted accumulated value of all past reciprocal fractal dimensions, then Equations (11),(12) and (13) show how new values for these global parameters are computed. The "*dk*" term is a forgetfulness factor that serves to age past data.

$RFD = RFD * dk + rdi$ (11)

$NUR = NUR * dk + 1$ (12)

Then, the new system merit is given by

$$SYM = RFD / NUR \quad (13)$$

Equation (14) shows how this can in turn provide a target or desired level of achievement, dla , for the statistical inference process to follow.

$$dla = c0 + c1 * SYM \quad (14)$$

Individual Cell Statistics.

Each cell object has seven attributes;

- m : a unique state number.
- um : a current matrix value (1 or f) = Y
- Nm : the number of times this cell was entered in last run
- $GN1m$: a historical count of cell entries when $u=1$
- $GU1m$: a historical count of runs using cell when $u=1$
- $GNfm$: a historical count of cell entries when $u=f$
- $GUfm$: a historical count of runs using cell when $u=f$

At the end of the run the global counts are updated for each state, m , according to Equation (15):

$$\begin{aligned} \text{IF } um=1 \text{ THEN } GU1m &= GU1m * dk + 1 \text{ \& } GN1m = GN1m * dk + Nm \\ \text{ELSE } GUfm &= GUfm * dk + 1 \text{ \& } GNfm = GNfm * dk + Nm \end{aligned} \quad (15)$$

Inference Parameters.

The fractal decision strengths for each decision value are now computed using Equations (16) & (17):

$$ds1m = p1m.log(p1m) \quad (16)$$

$$dsfm = pfm.log(pfm) \quad (17)$$

where $p1m$ and pfm represent a measure of the information value for either decision and incorporates the value of dla . Based on this value, the stored decision value is updated according to the rule given as Equations (18) & (19): If $ds1m=dsfm$, the stored value is unaltered.

$$\text{IF } ds1m < dsfm \text{ THEN } Y(m) \rightarrow 1 \quad (18)$$

$$\text{IF } dsfm > ds1m \text{ THEN } Y(m) \rightarrow f \quad (19)$$

This introduces an element of learning and provides a mechanism for the control value, attached using Equation (6), to influence the chaotic motion by learning to lower the fractal dimension (ie increase the reciprocal dimension.) The following section illustrates these effects for the same system shown in Figure 1. for which $di=1.764$.

STABILIZATION RESULTS

Figures 2 and 3 show Poincaré maps illustrating a progressive stabilization of the Lorentz system shown as Figure 1 for different BOXES conditions.

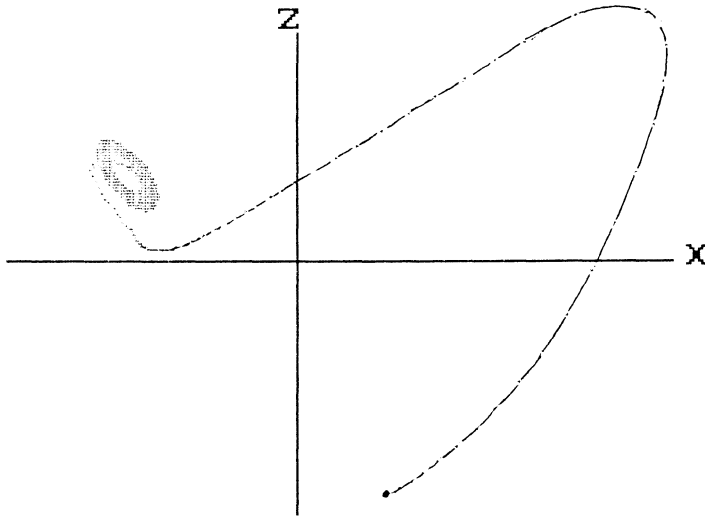


Figure 2: Poincaré Map for the Lorentz Equations with BOXES Stabilization. $di=0.371$

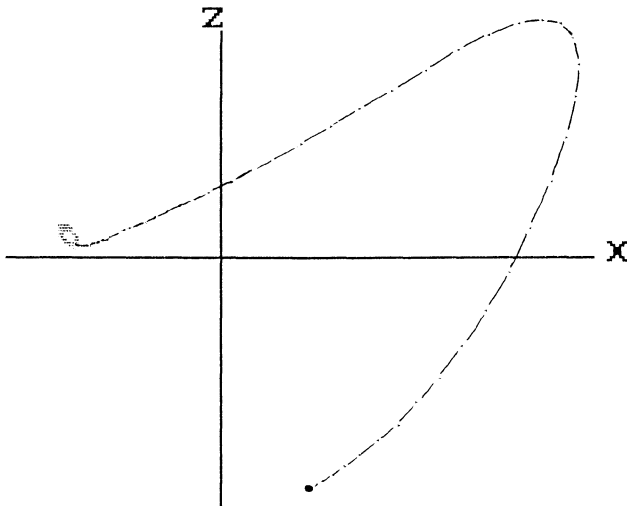


Figure 3: Poincaré Map for the Lorentz Equations with more BOXES Stabilization. $di=0.214$

CONCLUDING COMMENTS & FUTURE DIRECTIONS

This research is in early stages and this paper is of a forensic nature. For example, further work is needed to refine the algorithm that computes the information contribution of both decision options in Equations (16) and (17.) The actual connection of the automaton to a real-time, real-world chaotic system is also the subject of much deliberation.

However, the dramatic stabilizing effects produced by the attachment of a *BOXES*-type algorithm to the well-studied Lorentz equations are most encouraging. Clearly, the calculation of the information dimension requires very similar data capture and processing procedures as those found in the *BOXES* method.

REFERENCES

1. Schelberg, Charles " Parallel Scheduling of Random and Chaotic Processes" in *Dynamic, Genetic and Chaotic Programming--The Sixth Generation* (ed. Soucek et al.) John Wiley & Sons, New York, Chap. 20, 1993
2. Cornwell, John "What we think we are" *The Sunday Times*, Times Newspapers Ltd. London. July 11, 1993
3. Russell, David W. "A Critical Assessment of the *BOXES* Paradigm" *Applied Artificial Intelligence*, Vol 7, No. 4, Taylor & Francis, London, pp. 383-396, 1993.
4. Moon, Francis C. *Chaotic and Fractal Dynamics* John Wiley & Sons, New York, pp. 334 ff, 1992.
5. Michie, Donald and Chambers, R. A. "BOXES-an experiment in adaptive control" *Machine Intelligence II*, Oliver & Boyd, London, pp. 137-152, 1968
6. Russell, David W. & Rees, Steven, J. "System Control: a case study of a statistical learning automaton" *Progress in Cybernetics and Systems Research*. Vol.2 Hemisphere Publishing Corp., New York pp. 114-120, 1974.
7. Russell, David W. "AI Augmented Goal Achievement in Process Control Systems" *Proc. AIRTC'91 3rd. IFAC Intl. Workshop on Artificial Intelligence in Real Time Control*, Sonoma, CA Pergamon Press, New York, pp. 75-79, 1991.