Contributions to the Study of Autonomous Chaotic Circuits and Cellular Automata

Bharathwaj Muthuswamy

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2009-164
http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-164.html

December 9, 2009

Contributions To The Study of Autonomous Chaotic Circuits and Cellular Automata

by

Bharathwaj Muthuswamy

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Engineering-Electrical Engineering and Computer Sciences

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Leon O. Chua, Chair
Professor Pravin P. Varaiya
Professor Andrew Szeri

Fall 2009

Contributions To The Study of Autonomous Chaotic Circuits and Cellular Automata

by

Bharathwaj Muthuswamy

Abstract

Contributions To The Study of Autonomous Chaotic Circuits and Cellular Automata

by

Bharathwaj Muthuswamy

Doctor of Philosophy in Engineering-Electrical Engineering and Computer Sciences

University of California, Berkeley

Professor Leon O. Chua, Chair

This work focuses on autonomous chaotic circuits and cellular automata. In the realm of chaotic systems, it is often difficult to rigorously prove the existence of chaos. For example, the Lorenz system that was discovered in 1963 was rigorously proved to be chaotic only in 1999, after a span of 36 years. Hence, the first part of this thesis concerns rigorous proofs of chaos. The first approach uses a combination of linear dynamics, trajectory analysis in the Jordan space and describing function techniques for period-doubling bifurcations. This approach is applied to a rigorous proof of chaos in the Four-Element Chua's circuit, the simplest chaotic circuit. To our knowledge, this is the first rigorous proof of chaos in the Four-Element Chua's circuit. The second approach involves topological horseshoe theory and is applied to memristor based chaotic systems. This thesis also proposes a realization of a memristor based chaotic system on the breadboard. To our knowledge, this is the first analog realization of a memristor which is not based on designing a mutator for converting a v-i curve into a phi-q curve and also the first rigorous verification of chaos in a memristor based chaotic system. Both these approaches provide the reader with mathematical tools for investigating the behavior of continuous time chaotic systems. In the second part of this thesis, the relationship between bit length and attractor periods of totalistic one dimensional cellular automata are classified. Specifically, the relationship between integer factorization and dynamics of totalistic one dimensional cellular automata is explored for the first time.

The organization of this thesis is: in Chapter 1 we discuss background material necessary for understanding this thesis. Chapter 2 discusses the rigorous proof of chaos in the Four-Element Chua's circuit. Chapter 3 involves memristor based (higher dimensional) chaotic circuits and topological horseshoe theory. Chapter 4 explores the relationship between integer factorization and cellular automata. This is followed by conclusions with suggestions for future work, bibliography and appendices with simulation code.

Dedicated To God

Contents

List of Figures

List of Tables

Acknowledgments

First, I want to thank Ferenc Kovac of the Electronic Support Group and Carl Chun from the Connectivity Lab for funding me all these years. They have also been professional and personal mentors. Many thanks to Hugo Andrade from National Instruments for the additional funding. The EECS department was generous enough to let me be a teaching assistant all these years. Coming to the faculty side, Dr. Pravin Varaiya was instrumental for all my success at the University of California, Berkeley. I owe my PhD and my career to Pravin. My advisor, Dr. Leon Chua, has provided me with the necessary emotional and intellectual support for finishing this thesis. It will always be a pleasure to interact with Leon. I will always be sad that I did not get to work with Leon since the 1960s! Dr. Andrew Szeri, the external committee member and Head of the Graduate Division at UC Berkeley, has taken the time to be on my qualifying exam committee. He was also very helpful when I met him in February 2009 and offered me valuable advice that helped me finish this thesis.

My friends - Sandra, Andrea, Tamara, Pracheta, Alex, Prahalika, Priyanka, and Shimul - where would I be without you? A shout-out to the IEEE lab folks in Spring 2009, my family away from family! Ian, Prashant, Ronnie, Raffi (East LA City College) and Sergei - all those hours of CounterStrike kept me going. The one-two punch of Donovan and Joanna kept me motivated. Sheila Ross and Ashwin Ganesan were there for me during the dark ages. ”Crazy” Joe Makin, Alessandro Abate and John Secord helped me pass my prelims.

My dad, with the help of my mom, made a Michael Jordanesque comeback from a life threatening illness to see me graduate with a PhD. My brother has always been there for me and will always be there for me. Ruth Gjerde helped me navigate the minefield that is graduate bureaucracy. A round of thanks to the Cory hall folks and everyone who has worked with me throughout the years for all their help. Last but not least I want to thank my wife, Deepika, who has been with me always.

Chapter 1

Introduction

In this chapter we will go through the background material required for understanding this dissertation. The first section of this chapter explains the mathematical notation used in the dissertation.

Next we will give an introduction to autonomous chaotic circuits. First we look at a brief history of chaos. We will then discuss the four basic elements that serve as the building blocks of circuits. This is followed by a discussion of the first autonomous chaotic circuit, Chua's circuit [28]. This section will be concluded with a list of references for studying chaos.

The Cellular Automata section of this chapter also begins with a brief history followed by an overview of Wolfram's work on the algebraic properties of Cellular Automata [47]. Next Chua's application of nonlinear dynamics to Cellular Automata is discussed [12]. The section concludes with a list of references for Cellular Automata. The chapter concludes with a summary of the main contributions of this thesis.

1.1 Mathematical Notation

The mathematical notation used in this thesis is standard; nevertheless, this section clarifies the notation used in this thesis [27]. First, crucial terms and ideas in the text are emphasized in *italic* font. Main results are emphasized in ***bold italics font***.

Lowercase letters from the Latin alphabet (a - z) are used to represent variables, with italic script for scalars and bold invariably reserved for vectors. The letter t is of course always reserved for time.

Real-valued functions, whether scalar- or vector-valued, are usually taken (as conventionally) from lowercase Latin letters f through h, plus r and s. Vector-valued functions and vector fields are bolded as well, the difference between the two being indicated by the argument font; hence $\mathbf{f}(x)$ and $\mathbf{f}(\mathbf{x})$ respectively. When these letters have been exhausted, functions resort to the back of the Greek alphabet, capitalization there indicating vector outputs.

Integers are represented by the lowercase Latin letters from i to q - excluding l and o for their likeness to 1 and 0, and k for its use as a generic rate constant - with n usually reserved for the dimension of the state.

(Constant) matrices and vectors are represented with capital and lowercase letters, respectively, from the beginning of the Latin alphabet. Vectors are again bolded. In the context of linear time-invariant systems, the usual conventions are respected: A is the state matrix and B ($\mathbf{b}$) is the input matrix (vector). Constant scalars are usually drawn from the beginning of the Greek alphabet, although the sampling interval is always symbolized by T.

Calligraphic script is reserved for sets, which use capital Latin letters. Elements of sets are then represented with the corresponding lowercase letter. Excepted are the well known number sets, which are rendered in blackboard bold: $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{R}$, and $\mathbb{C}$ for the naturals, integers, reals, and complex numbers, respectively. The natural numbers are taken to include 0; restrictions to the positive or negative subsets are indicated by a superscripted $+$ or $-$. As usual, $\mathcal{C}^0$ is the set of continuous functions.

Subscripts denote the elements of a matrix or vector: $\mathbf{d}_i$ is the i^{th} column of D; x_j is the j^{th} element of $\mathbf{x}$. Plain numerical superscripts on the other hand may indicate exponentiation, a recursive operation, or simply a numbering, depending on context.

Differentiation is expressed as follows. Time derivatives use Liebniz's notation or Newton's notation: one, two, or three dots over a variable for the corresponding number of derivatives, and a parenthetical superscripted numeral for higher derivatives. Leibniz's notation is used explicitly for non-time derivatives.

Finally: $\sum_i$ is used for summations; a superscripted T indicates the matrix transpose; $\in$ denotes set inclusions; I is reserved for the identity matrix. All vectors are assumed to be columns.

1.2 Autonomous Chaotic Circuits

1.2.1 A Brief History of Chaos

In 1889, to commemorate the 60th birthday of King Oscar II of Sweden and Norway, a contest was held to produce the best research in celestial mechanics pertaining to the three-body problem. The problem involved a system of three masses interacting exclusively through gravitational acceleration. The winner was declared to be Henri Poincare, a professor at the University of Paris [1].

Poincare submitted an entry full of seminal ideas. In order to make progress on the problem, he made three simplifying assumptions. First, he assumed that the three bodies were all moving in a plane. Second, he assumed that two of the bodies were massive and that the third had negligable mass, so small that it did not affect the motion of the other two. We can imagine two stars and a small asteroid. In general,

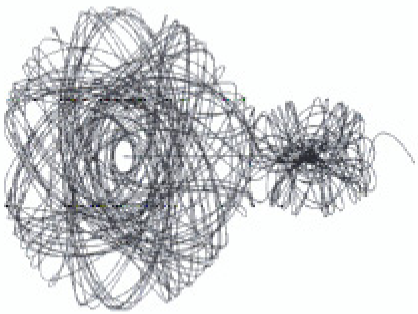


Figure 1.1: A trajectory of a tiny mass in the three-body problem

.

the two large stars would travel in ellipses. But Poincare made another assumption, that the initial condition was chosen such that the two massive bodies moved in circles at a constant speed about their center of mass. Fig. 1.1 shows a typical trajectory of the tiny mass in such a configuration [1]. The two larger bodies are in circular motion around one another. This view is of a rotating coordinate system in which the two larger bodies lie at the left and right ends of the horizontal line segment. The tiny mass is eventually ejected toward the right. Other trajectories starting close to one of the bodies can be forever trapped. Poincare's method of analysis was based on the fact that the motion of the small mass could be studied, in a rather nonobvious manner, by studying the orbit of a plane map. He discovered the crucial ideas of *stable and unstable manifolds* which are special curves in the plane [1]. Poincare's final article *Sur les equations de la dynamique et le problema des trois corps* (on the equations of dynamics and the three-body problem) was published in 1890. In this 270-page work, Poincare established convincingly that no general exact formula exists for making predictions of the positions of the three bodies in the future [1]. However, even after Poincare's seminal work, chaos was largely forgotten till the middle of the 20^{th} century [1].

In the late 1950s, a meteorologist at MIT named Edward Lorenz acquired a Royal-McBee LGP-30 computer. It was the size of a refrigerator carton and contained 16KB of internal memory. It could calculate at the rate of 60 multiplications per second. For the time, it was a staggering cache of computational power to be assigned to a single scientist [1]. Although Lorenz initially started with a system of 12 ordinary differential equations, he eventually simplified the model to a system of three ordinary

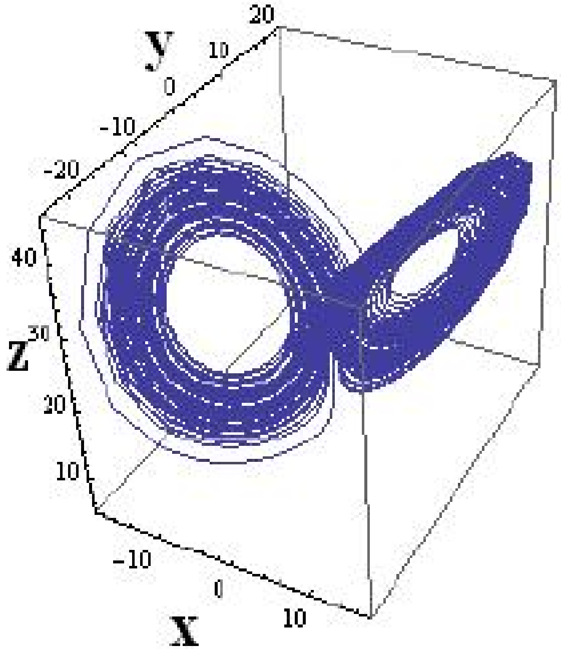


Figure 1.2: The Lorenz butterfly in phase space, observed from simulating Equation 1.1 for 100 seconds. Parameters are $\sigma = 10$, $\rho = 28$, $\beta = \frac{8}{3}$. Initial conditions are $x_0 = 10$, $y_0 = 20$, $z_0 = 30$.

differential equations, the Lorenz equations, shown in Eq. 1.1:

$$\begin{aligned} \dot{x} &= -\sigma \cdot x + \sigma \cdot y \\ \dot{y} &= -x \cdot z + \rho \cdot x - y \\ \dot{z} &= x \cdot y - \beta \cdot z \end{aligned} \tag{1.1}$$

In this highly idealized model of a fluid, the warm fluid below rises and the cool fluid above sinks, setting up a clockwise or counterclockwise current. The Prandtl number σ, the Rayleigh (or Reynolds) number ρ and β are parameters of the system. The variable x is proportional to the circulatory fluid flow velocity. If $x > 0$, the fluid circulates clockwise while $x < 0$ means the fluid circulates counterclockwise. The variable y is proportional to the temperature difference between ascending and descending fluid elements, and z is proportional to the distortion of the vertical temperature profile from its equilibrium [1]. Fig. 1.2 shows the result of simulating the system above. The relevant Mathematica code is in the Appendix. Using such computer simulations, Lorenz identified the presence of *sensitive dependence on initial conditions* and *aperiodicity*, the hall marks of chaos. Lorenz's original paper [26] on this work is deep, prescient and readable.

Since Lorenz observed chaotic behavior via computer simulations, a natural question to ask is whether chaos is an artifact of computer simulation or whether there is an underlying mathematical theory behind the behavior. Although Lorenz used a variety of powerful geometric arguments to explore the trajectories in phase space, a rigorous proof of chaotic behavior was absent. But, researchers found other chaotic systems [33], implying that chaos was not unique to the Lorenz model. Yet, these chaotic systems were difficult to realize using either electronic or mechanical components. Although electronic circuits were easy to construct, both the Lorenz and Rossler systems had product terms. This required the use of analog multipliers, these were expensive and unreliable during the 1970s and 1980s. Hence, although there was mounting evidence that chaos was a field of study in its own right, a system that was easy to build practically and also had an associated rigorous proof of chaos was frustratingly absent. This changed in 1984, when Dr. Leon O. Chua, invented Chua's circuit. Before discussing this circuit, we will briefly go through the four basic circuit elements.

1.2.2 The Four Basic Circuit Elements

From our basic electronics courses, we know that the fundamental variables for circuit theory are flux (denoted by ϕ) and charge (denoted by q). Voltage is defined as:

$$v \equiv \frac{d\phi}{dt} \tag{1.2}$$

Current is defined as

$$i \equiv \frac{dq}{dt} \tag{1.3}$$

Based on the quantities above, we can define the circuit elements shown in Fig. 1.3. Notice that we have a fourth circuit element (in addition to the usual three from basic electronics) that defines a relationship between ϕ and q. This element is the *memristor*, more will be said about the memristor in the chapter on memristor based chaotic circuits.

1.2.3 An Introduction to Chua's Circuit

In 1984 Dr. Leon O. Chua, as a visiting professor in Japan, systematically discovered Chua's circuit [6]. The circuit is shown in Fig. 1.4. The equations associated with the circuit in Fig. 1.4 are:

$$\begin{aligned} C_1\dot{v_{C1}} &= G(v_{C2} - v_{C1}) - g(v_{C1}) \\ C_2\dot{v_{C2}} &= G(v_{C1} - v_{C2}) + i_L \\ L\dot{i_L} &= -v_{C2} \end{aligned} \tag{1.4}$$

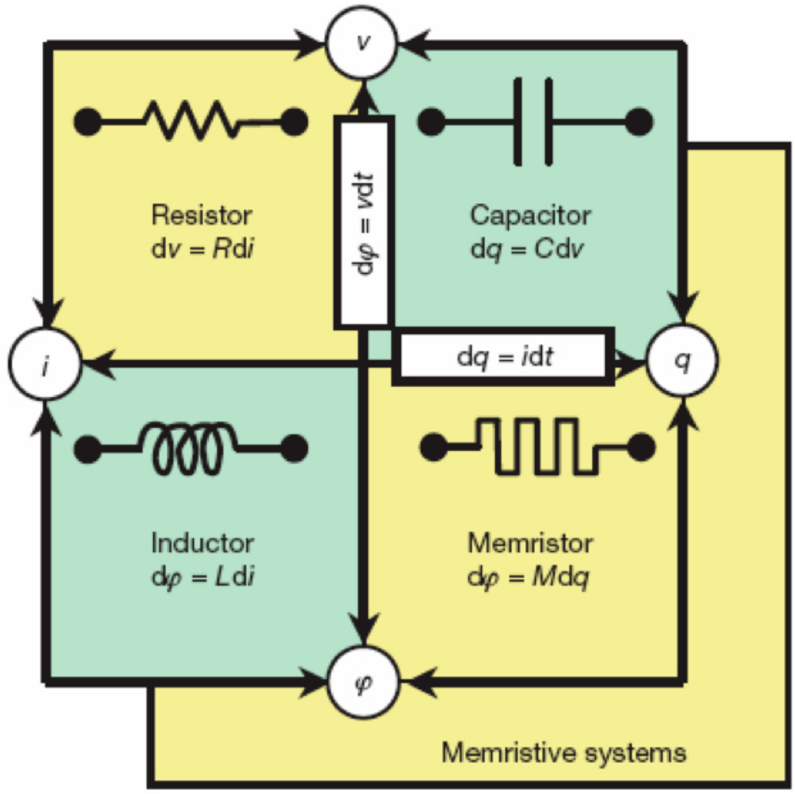


Figure 1.3: The Four Basic Circuit Elements

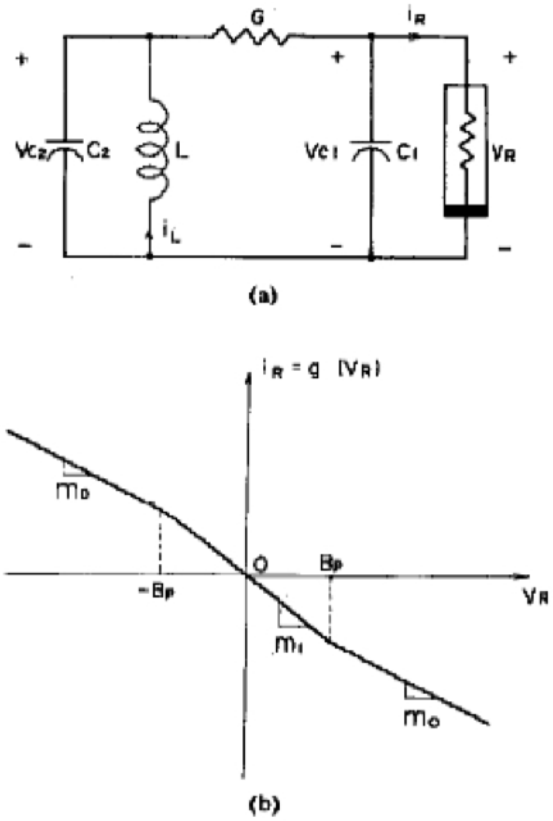


Figure 1.4: Chua's circuit is a simple autonomous circuit with a chaotic attractor.

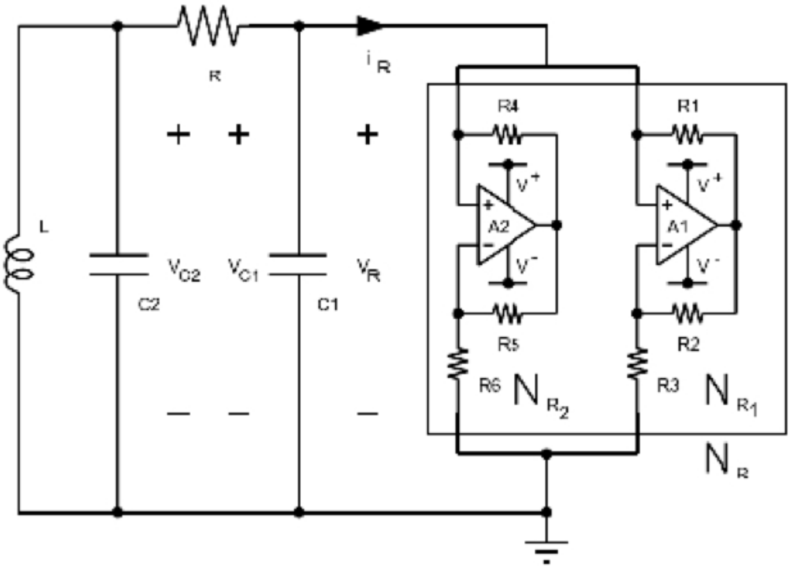


Figure 1.5: Realization of Chua's circuit using two op amps and six linear resistors to implement the nonlinear resistor

Here, v_{C1}, v_{C2} and i_L denote the voltage across C_1, C_2 and the current through L respectively. $g(v_{C1})$ is the piecewise-linear function in Fig. 1.4 given by:

$$g(v_{C1}) = m_0 v_{C1} + \frac{1}{2}(m_1 - m_0)(|v_{C1} + B_p| - |v_{C1} - B_p|) \tag{1.5}$$

It is simple to realize the electronic circuit above. Fig. 1.5 shows the classic two op-amp implementation by M.P. Kennedy [22]. A 2D projection of the "double-scroll" on an oscilloscope screen from the circuit above is shown in Fig. 1.6. The parameters of the circuit used for obtaining Fig. 1.6 are L = 18 mH, C2 = 100 nF, C1 = 10 nF, R = 1.5 kΩ, R1 = R2 = 220 Ω, R3 = 2.2 kΩ, R4 = R5 = 22 kΩ and R6 = 3.3 kΩ. All resistors and capacitors have 5% tolerance, the inductor has 10% tolerance. The op-amps are LMC6482AINs from National Semiconductor. The circuit is powered by two 9 V batteries.

1.2.4 References for Chaotic Systems

A good introduction to chaos is given in the book by Strogatz [40]. Alligood's book [1] is an intermediate introduction to chaos. Wiggins' book [43] covers very advanced topics and is an all-in-one reference. A good introduction to linear and nonlinear circuit theory is [7]. An excellent starting point for further understanding Chua's circuit is [6].

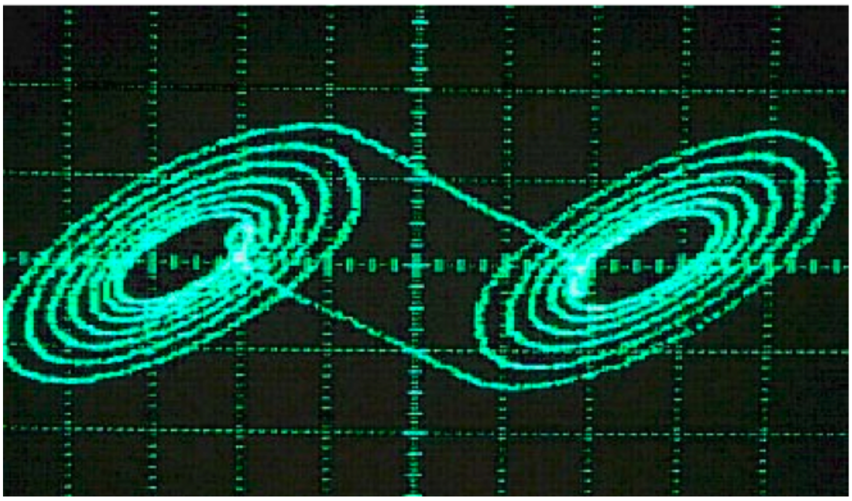


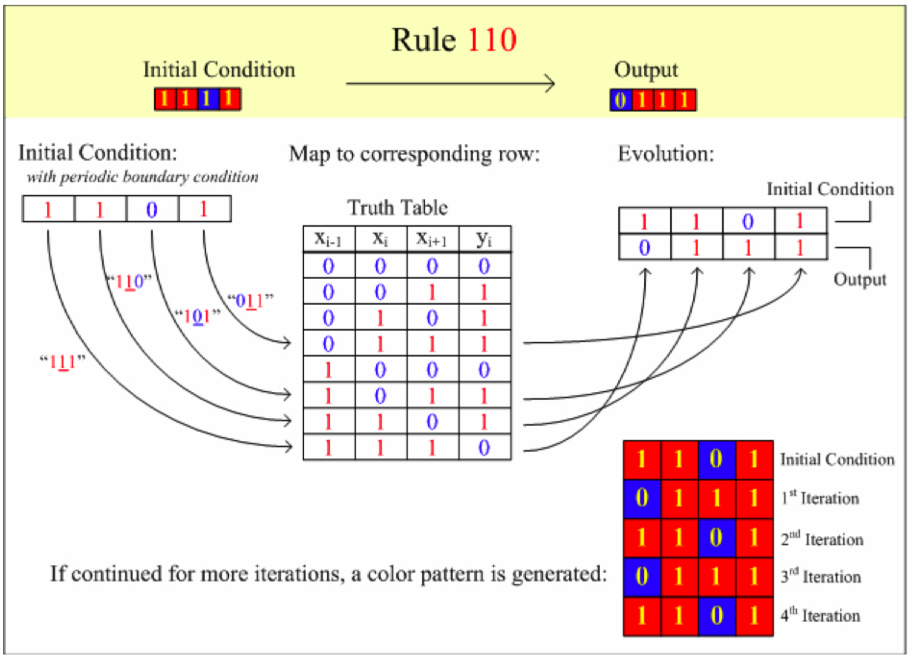
Figure 1.6: The oscilloscope showing the double-scroll, X-axis is measuring v_{C1} (scale is 1.00 V/div) and the Y-axis measuring v_{C2} (scale is 0.5 V/div).

1.3 Cellular Automata

1.3.1 A Brief History of Cellular Automata

Cellular automata (henceforth abbreviated CA) are simple models of computation which exhibit fascinatingly complex behavior [35]. They were originally proposed by John von Neumann as formal models of self-reproducing organisms. The structure studied was mostly on one- and two-dimensional infinite grids, though higher dimensions were also considered. Computation universality and other computation-theoretic questions were considered important.

The simplest description of a CA is a one-dimensional array of infinite cells. Time is discrete, at each time point each cell is in one of a finite set of possible states. The cells change state at each iteration, and the new state is completely determined by the present state of the cell, its left neighbor and right neighbor. The function (called the local rule) which determines this change of state is the same for all cells. The automaton does not have any input, and hence is autonomous. The collection of cell states at any time point is called a configuration or a global state of the CA, and describes the stage of evolution of the CA. At time $t = 0$, the CA is in some initial configuration, and henceforth proceeds deterministically under the effect of the local rule, which is applied to each cell at every iteration [35] (see Fig. 1.7 [31]).



Rule 110

Initial Condition

1	1	1	1

Output

0	1	1	1

Initial Condition:
with periodic boundary condition

1	1	0	1

Map to corresponding row:

Truth Table

x_{i-1}	x_i	x_{i+1}	y_i
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Evolution:

1	1	0	1	Initial Condition
0	1	1	1	Output

If continued for more iterations, a color pattern is generated:

1	1	0	1	Initial Condition
0	1	1	1	1st Iteration
1	1	0	1	2nd Iteration
0	1	1	1	3rd Iteration
1	1	0	1	4th Iteration

Figure 1.7: Example of Cellular Automata evolution, the local rule used is 110. Here 1101 is the initial condition for a 1-D 4-cell CA. The cells can be in state 0 or state 1 (binary CA). The truth table for rule 110 is given. Consider the first cell that contains a 0 (the rightmost cell is numbered 0). The left and right neighbor cells are the second and zero cell, respectively, both containing 1. The output of the first cell would be the output mapped from the 3-tuple (1,0,1) from the truth table of local rule 110. According to the truth table, when $x_{i-1} = 1, x_i = 0$, and $x_{i+1} = 1$, the output $y_i = 1$ and hence the output of the first cell at iteration 1 is "1". The same process is repeated for each cell listed in the initial condition, resulting in the new evolution 0,1,1,1.

1.3.2 Wolfram's work on Cellular Automata

The mid-1980s are an important period in the history of CA, largely due to the work carried out by Wolfram. The nature of his questions represent a paradigm shift in CA research. Wolfram carried out an extensive experimental analysis of the growth patterns of CA. An early paper by Wolfram [47] discusses several statistical parameters of the space-time patterns of CA evolution. Later work extended and clarified much of the intuition in several directions. The approach taken is to consider CA as models of complex systems, in the sense that very simple CA rules can give rise to extremely complex patterns. Wolfram published a conglomerate of his ideas in a 2002 book [46].

1.3.3 Nonlinear Dynamics Perspective of Cellular Automata

Starting in 2002, Dr. Leon. Chua published a series of papers ([12], [9], [10], [11]) that explained Wolfram's observations in [46] using a nonlinear dynamics perspective. In particular, Chua developed a geometrical approach for defining an integer characterization of all Boolean functions arising from binary 1D CA with nearest neighbors.

1.3.4 References for Cellular Automata

Wolfram's collection of papers in his 1986 book [45] serves as a good starting point for understanding the basics of Cellular Automata. Dr. Chua's paper series: [12], [9], [10], [11] connects nonlinear dynamics to Cellular Automata.

1.4 A Summary of Thesis Contributions

Ever since Lorenz's seminal paper [26], many more chaotic systems have been discovered [6]. However providing a rigorous proof of chaos in such systems is difficult [28]. Chua's circuit (from Fig. 1.4) was the first system that was proved to be chaotic rigorously [28]. But, a consequence of the Poincare-Bendixson theorem [23] is that for an autonomous continuous-time dynamical system to be chaotic, we need 3 or more dimensions. In the case of circuits, we need three *independent* energy storage elements. Therefore if we try and simplify Chua's circuit, the only element we can replace is the linear resistor, leading to Fig. 1.8. Notice that we no longer have a 3 dimensional system since the two capacitors can be combined in parallel.

Nevertheless, Barboza [2] in 2008 derived the circuit shown in Fig. 1.9. This circuit is linearly conjugate to the original Chua's circuit.

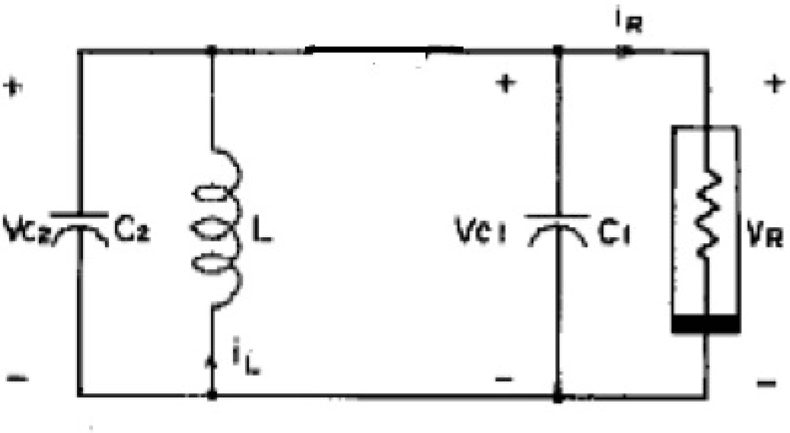


Figure 1.8: Removing the linear resistor (i.e..replacing it with a short circuit) in Chua's circuit leads to a two dimensional system because C_1 and C_2 can be combined in parallel

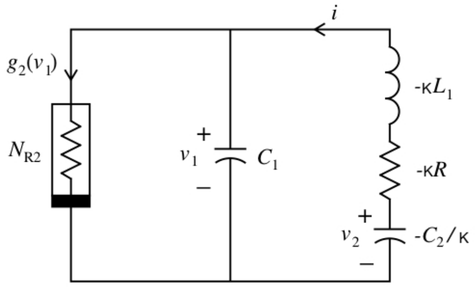


Figure 1.9: Alternate version of Chua's Circuit

But notice that in the limiting case of $R \to 0$, we get the circuit shown in Fig. 1.10. Note that this circuit has only four elements, yet is modeled by three differential equations. Therefore, this is the simplest possible autonomous chaotic circuit, the Four-Element Chua's circuit.

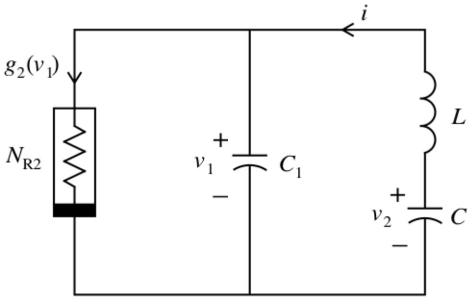


Figure 1.10: A chaotic circuit with only four elements

However a rigorous proof of chaos in the circuit above was still lacking in [2]. Hence the first contribution of this work is to provide a rigorous proof of chaos in this circuit. This proof utilizes the simplified Jordan form and coordinate transform approach proposed in [28]. Also bifurcation analysis is done using describing function methods [17] since we have a one-parameter bifurcation family for the Four-Element Chua's circuit (for details, please refer to the chapter on Rigorous Proof of Chaos in the Four-Element Chua's circuit).

Looking back at our review of the four fundamental circuit elements, a natural question to ask is whether memristors can be used to build chaotic circuits. The answer is yes, this thesis proposes memristor based (four and higher dimensional) chaotic circuits (systems). We first systematically derive a memristor based chaotic circuit from the canonical Chua's circuit. We next demonstrate that the four dimensional chaotic circuit can be linearly extended to five dimensions by simply adding an inductor. We also use topological horseshoe theory to rigorously verify the existence of chaos in the memristor based chaotic circuits. We also implement a memristor based chaotic circuit on the breadboard. In a nutshell, the second contribution of this thesis is design and implementation of memristor based chaotic systems.

From the perspective of Cellular Automata, this thesis explores the relationship between integer factorization and cellular automata.

Part I

Autonomous Chaotic Circuits

Chapter 2

Rigorous Proof of Chaos in the Four-Element Chua's Circuit

2.1 Introduction and Chapter Organization

In this chapter, a mathematically rigorous proof of chaos is provided for the Four-Element Chua's circuit. Although Chua et. al. provided a rigorous proof of chaos for the canonical Chua's circuit [8], the Four-Element Chua's circuit [2] is a limiting case of the classic Chua's circuit. Thus the circuit requires a separate rigorous justification of chaos. In the next section, the Four-Element Chua's circuit is introduced and the main theorem to be proved in this chapter is stated. Note that we actually prove chaos in a general class of alternative Chua's circuit of which the Four-Element Chua's circuit is a special case. Section three gives the intuitive ideas behind the proof. The subject of the fourth section is the rigorous proof of chaos. The final section uses describing function methods to verify period-doubling route to chaos in the Four-Element Chua's circuit [17], [20], [39].

2.2 Introduction to the Four-Element Chua's circuit and Main Result

In this section, the circuit diagram and equations are given. For details on the linear conjugacy between this circuit and the canonical Chua's circuit, please refer to [2].

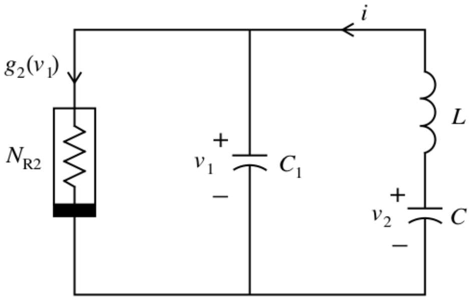


Figure 2.1: The Four-Element Chua's Circuit

Figure 2.1 is the circuit diagram, the equations for the circuit are given in Eq. 2.1.

$$\begin{aligned} C_1\frac{dv_1}{dt} &= i - g_2(v_1) \\ \kappa L_1\frac{di}{dt} &= v_1 - v_2 \\ C_2\frac{dv_2}{dt} &= \kappa i \end{aligned} \tag{2.1}$$

The nonlinear function $g_2(v_1)$ is given by:

$$g_2(v_1) = 9.333 \cdot 10^{-4} v_1 + (\frac{-5 \cdot 10^{-4} - 9.333 \cdot 10^{-4}}{2})(|v_1 + 1| - |v_1 - 1|) \tag{2.2}$$

The parameter values that gives rise to the strange attractor in Fig. 2.2 and the time domain waveform in Fig. 2.3 are: $\kappa = 8.33$, $C_1 = 33 \cdot 10^{-9}$, $C_2 = 100 \cdot 10^{-9}$, $L_1 = 10 \cdot 10^{-3}$. The relevant Mathematica code is in the Appendix.

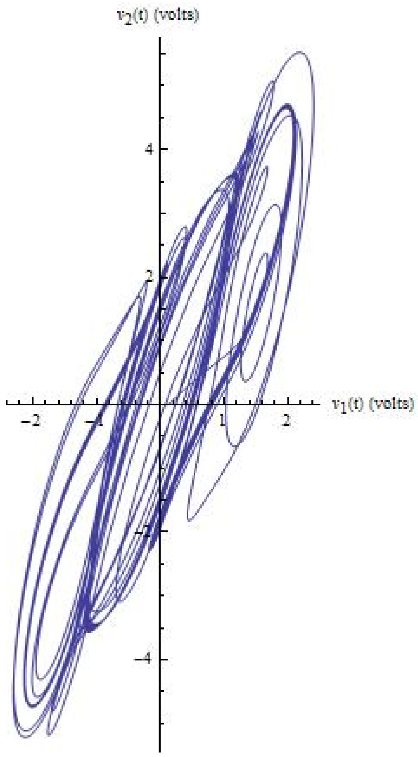


Figure 2.2: A plot of $v_2(t)$ vs. $v_1(t)$ for the Four-Element Chua's circuit

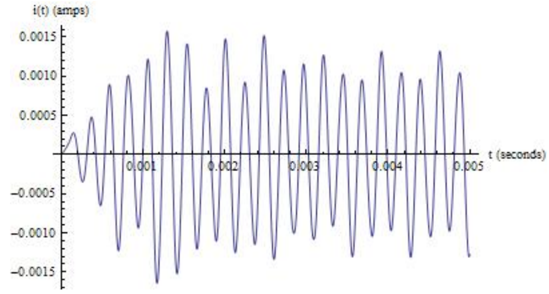


Figure 2.3: Plot of $i(t)$ for the first 5 ms

Fig. 2.4 shows a screen shot of an oscilloscope in XY mode, sampling $v_2(t)$ and $v_1(t)$ from a physical implementation of Chua's circuit.

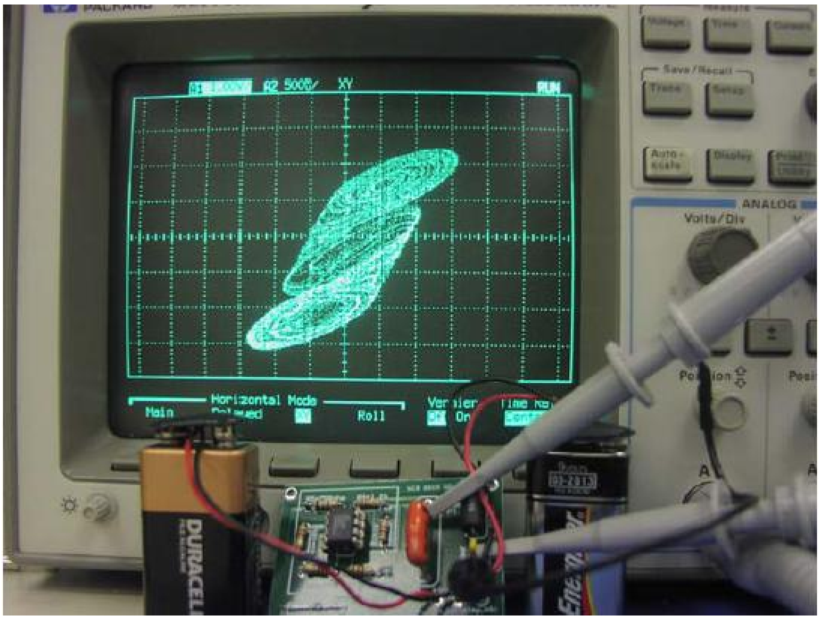


Figure 2.4: $v_2(t)$ vs. $v_1(t)$ from an oscilloscope. The Printed Circuit Board realization of the Four-Element Chua's circuit is shown.

This chapter actually provides a proof of chaos in the more general circuit shown in Fig. 2.5. Notice that we get the Four-Element Chua's circuit when $R \to 0$.

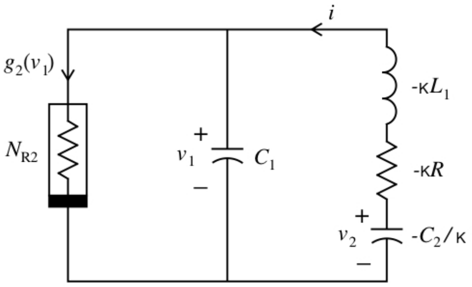


Figure 2.5: Alternate version of Chua's Circuit

First it would be useful to transform the circuit equations representing Fig. 2.5 to dimensionless form with the following change of variables:

$$\begin{aligned}
\kappa &= \frac{R_1}{R_2} \\
\tau &= \frac{t}{\kappa \cdot \sqrt{L_1 C_2}} \\
\alpha &= \kappa \frac{C_2}{C_1} \\
\beta &= \kappa^2 \\
\gamma^* &= \rho\kappa \\
x &= \frac{v_1}{B_p} \\
y &= \sqrt{\frac{L_1}{C_2}} \frac{i}{B_p} \\
z &= \frac{-v_2}{B_p} \\
f(x) &= \sqrt{\frac{L_1}{C_2}} \frac{g_2(B_p x)}{B_p}
\end{aligned} \tag{2.3}$$

In the$f(x)$ above, $B_p = 1$. The dimensionless form of Chua's circuit is thus:

$$\begin{aligned}
\frac{dx}{d\tau} &= \alpha(y - f(x)) \\
\frac{dy}{d\tau} &= x - \gamma^* y + z \\
\frac{dz}{d\tau} &= -\beta y \\
f(x) &= m_1 x + \frac{1}{2}(m_0 - m_1)(|x+1| - |x-1|)
\end{aligned} \tag{2.4}$$

The theorem below is the main result that will be proved in this chapter. The methodology used in the proof is general and can be applied to prove chaos in a three-dimensional autonomous piecewise-linear system.

Result 1. ***The alternative Chua's system in Eq. 2.4 is chaotic in the sense of Shilnikov if*** $\alpha = 25.24, \beta = 69.39, m_0 = -0.16, m_1 = 0.29, \gamma^* \in J \overset{\triangle}{=} [0, 0.75]$***. Notice that*** $\gamma^* = 0$ ***is the Four-Element Chua's circuit. This circuit also undergoes a period-doubling route to chaos, as*** γ^* ***decreases from*** 1 *to* 0***.***

Before giving a formal proof of chaos, we will give some intuitive ideas behind the proof. This is important to gain physical insights into the problem and also appreciate the intricacies of the mathematics involved.

2.3 Intuitive Ideas Behind the Proof

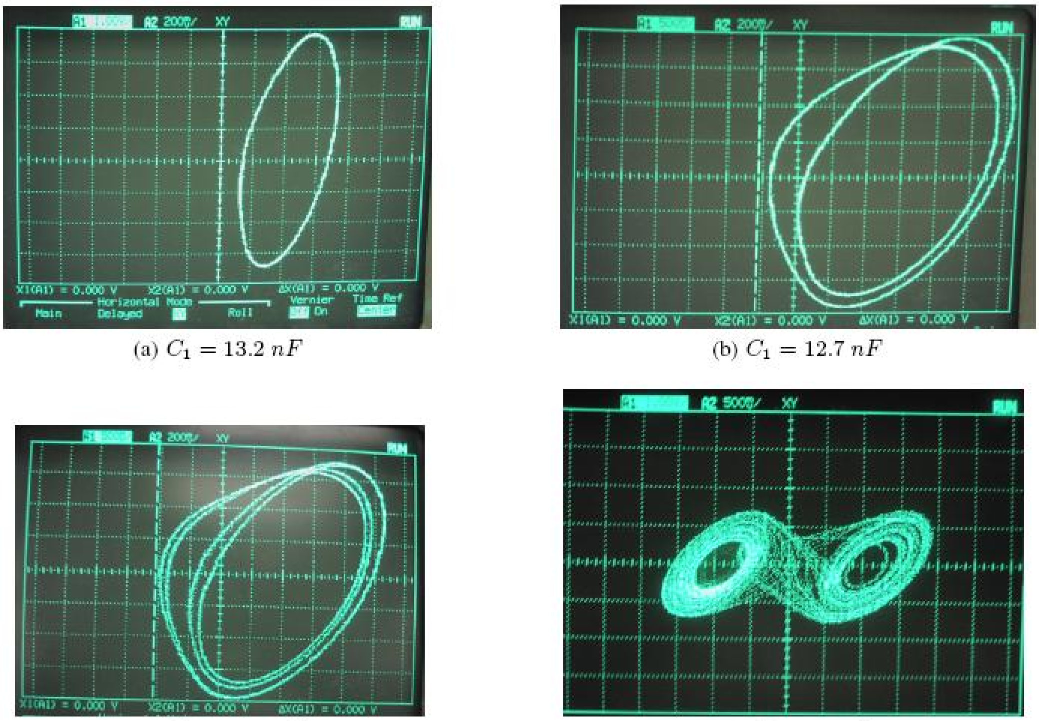
As the study of chaos evolved, a working definition was developed that is used to this day [37]. A dynamical system is informally called chaotic if it exhibits the following features:

1. *A basically continuous and possibly banded, Fourier or power spectrum.* This property indicates that the motion is nonperiodic and justifies the often made analogy of chaos with noise.

2. *Nearby orbits that diverge exponentially fast, thus causing extreme sensitivity to initial conditions.* Although this attribute is shared by many dynamical systems (for example, linear time-invariant dynamical systems), this alone does not gaurantee chaotic behavior.

3. *State trajectories are bounded.* This third property is probably the most important. This property seemingly contradicts the two properties above. How can we have aperiodic trajectories that are sensitive to initial conditions, yet bounded? The answer to this can be seen in the phase space of a chaotic system. For the *correct choice of parameters* we will get a *fractal structure in phase space called a strange attractor or chaotic attractor.*

Notice that the correct choice of parameters is important. All chaotic systems exhibit complex behavior only in a certain range of system parameters. The system also follows a *specific route to chaos* like *period-doubling*, *intermittency* etc. For example, period-doubling route to chaos is illustrated in Fig. 2.6.



(a) $C_1 = 13.2\ nF$

(b) $C_1 = 12.7\ nF$

(c) $C_1 = 12.2\ nF$

(d) $C_1 = 10\ nF$

Figure 2.6: Period doubling route to chaos in a synthetic inductor version of Chua's circuit. The bifurcation parameter is a capacitance. The oscilloscope is setup in X-Y mode and is probing the voltages across two capacitors.

Notice that we get the strange attractor in (d) after the state goes through a period-1, period-2 and period-4 limit cycle in (a) through (c) respectively. The system goes through an infinite period-doubling cascade but only three are shown here since it is impossible to visually distinguish limit cycles with higher periods.

Thus, any proof of chaos and bifurcation analysis of chaos requires two steps: a rigorous proof that a strange attractor exists in a given parameter space and the route to chaos (or *bifurcation analysis*). But why do we even need a rigorous proof? The reason is that computer simulations have finite precision and experimental measurements have finite ranges (e.g., time or frequency) [37]. The behavior witnessed might either be an artifact of the observation device or it might actually be regular with a period or bandwidth beyond the limits capturable by the device. To allay these anxieties, we need to rigorously prove the existence of chaos.

One of the most useful tools for studying chaos in autonomous systems is based on the fundamental work of Shilnikov [36]. The formal statement of Shilnikov's theorem and application is given later in this chapter. In this section, we will give a self-contained description of the basic concepts and terminology needed to understand the results of Shilnikov theory. We need to understand three concepts before using Shilnikov's theorem.

Fig. 2.7 presents a very special orbit that lies at the heart of the Shilnikov approach - the *homoclinic orbit*. This means a bounded dynamical trajectory of our system that is doubly asymptotic to an equilibrium point. That is, at time approaches ±∞ the system approaches the equilibrium point [29].

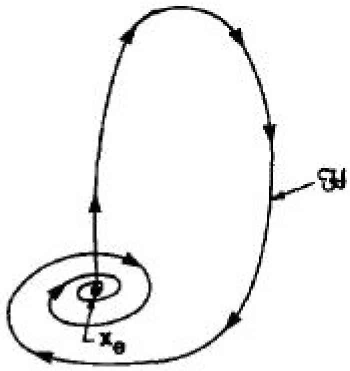


Figure 2.7: Illustration of a typical homoclinic orbit [37] in $\mathbb{R}^3$. The arrowheads indicate forward evolution of time. The homoclinic orbit $\mathbb{H}$ is based at a hyperbolic saddle focus x_e having a positive real equilibrium eigenvalue.

The next concept needed is that of a *Poincare map*, which is a stroboscopic means of analyzing the dynamics of nonlinear systems (refer to Fig. 2.8). For a three-dimensional system (like the Four-Element Chua's circuit), this technique amounts to using a plane $\Sigma \subset \mathbb{R}^2$ to cut traversely across recurrent behavior (as occurs local to a homoclinic orbit); this in turn defines a 2-D map $P : U \subset \Sigma \rightarrow \Sigma$, called the Poincare map. Here, the neighborhood U designates those points that return to Σ at least once under the flow. This map takes a point x_0 in U to the first intersection $P(x_0) = \phi^{t_r}(x_0)$ of the dynamical orbit from x_0 with the Σ-plane (t_r is the transit time for this orbit). Fig. 2.8 depicts a case in which the Poincare map is constructed in the neighborhood of a periodic orbit Δ (which itself is seen to correspond to a first-order fixed point of P, that is, $P(x_*) = x_*$).

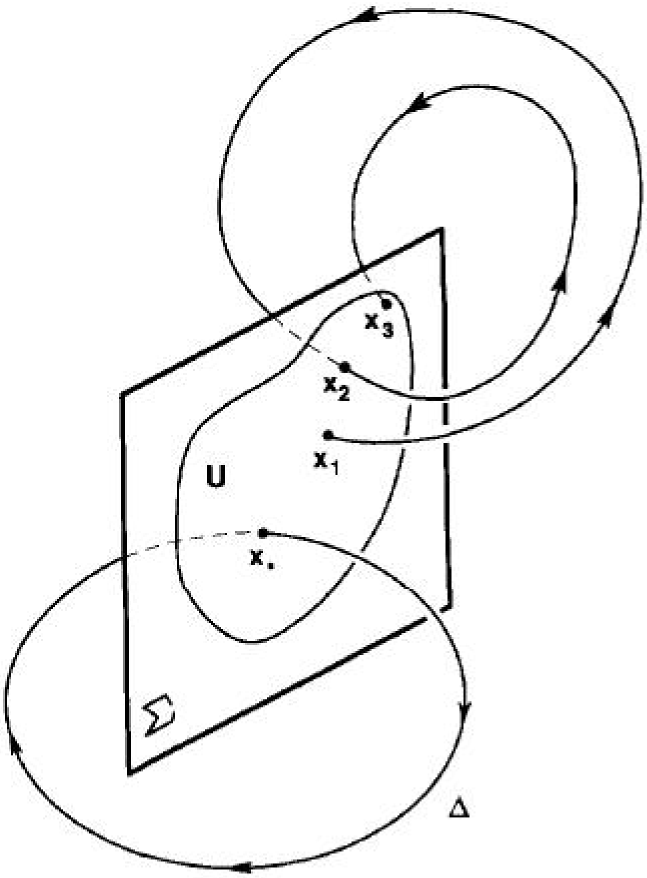


Figure 2.8: Illustration of Poincare map $P : U \to \Sigma$ local to a periodic orbit Δ [37]. We call the plane Σ a *local cross section* at x_*. The transversality of Σ to the flow means that $\xi(\mathbf{x})$ is not parallel to Σ for all x in Σ. A sample orbit initiating at $x_1 \in \Sigma$ and intersecting Σ twice more at x_2 and x_3 is shown. In terms of P, this implies that $x_2 = P(x_1), x_3 = P(x_2)$ and hence, $x_3 = P^{(2)}(x_1)$.

Observe that P defines a 2-D discrete dynamical system

$$x_k + 1 = P(x_k),\ k = 0, 1, \cdots \tag{2.5}$$

Thus, with the Poincare map approach one may study the reduced system above instead of the original 3-D system. For the case of a homoclinic orbit a characteristic local Poincare map P (called the Shilnikov map) can be constructed from two constituent maps: ψ_e, which corresponds to the linearized flow near the equilibrium point while the second one ψ_h, describes the behavior in a neighborhood of the homoclinic orbit away from the equilibrium point (see Fig. 2.9).

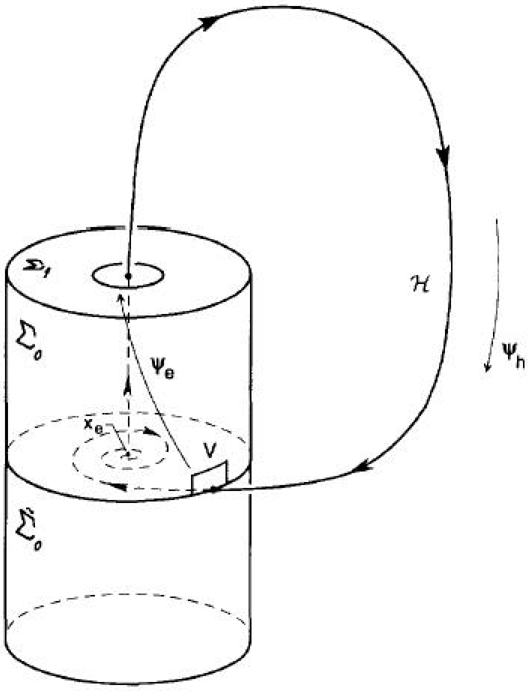


Figure 2.9: The Shilnikov map $P : V \to \Sigma_0 \cup \tilde{\Sigma_0}$ for the case of a homoclinic orbit $\mathbb{H}$ based at the saddle focus x_e [37]. Here, V is a small section of the cylindrical surface Σ_0. The map $\psi_e : \Sigma_0 \to \Sigma_1$ characterizes the behavior local to x_e whereas $\psi_h : \Sigma_1 \to \Sigma_0 \cup \tilde{\Sigma_0}$ takes care of the behavior local to the portion of $\mathbb{H}$ that is not in the neighborhood of x_e.

But, what is the point of computing a Shilnikov map? The answer lies in the third important concept - *the Smale horseshoe.* The Smale horseshoe [37] is the set analytically detected by the Shilnikov method in the discrete dynamics generated by the Shilnikov map (that is, in Eq. 2.5). In fact, it can be shown that when the Smale horseshoe is embedded in Eq. 2.5, then there exist orbits in the original dynamical system that typify chaotic behavior.

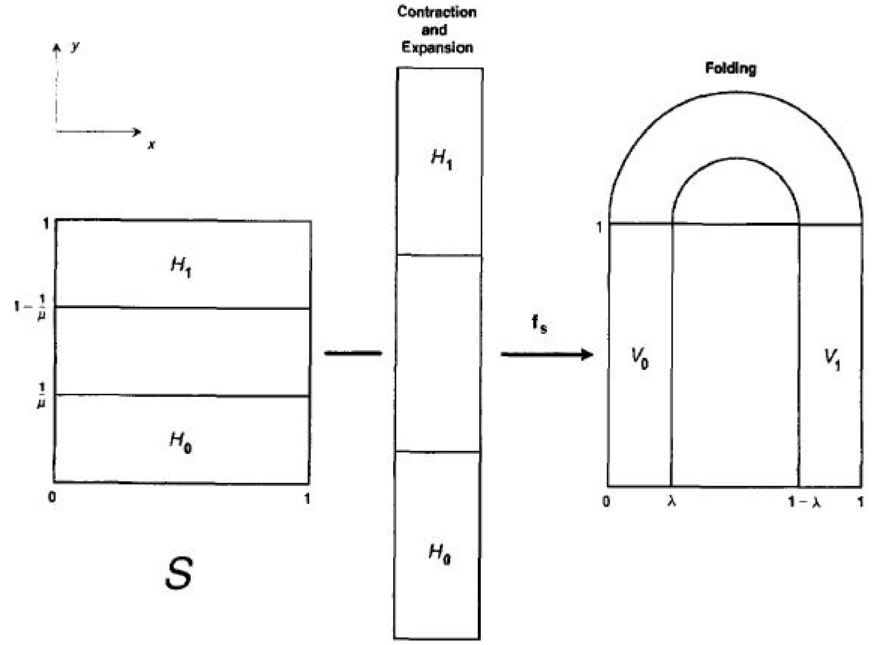


Figure 2.10: Geometric illustration of the simplified Smale horseshoe map f_s [37]. Its basic characteristics of 1) mapping disjoint regions (such as H_0 and H_1) over themselves and 2) "strong" stretching and contraction in complementary directions are representative of the Poincare maps for many of the continuous dynamical systems that exhibit chaotic behavior.

In its simplest form, the Smale horseshoe map can be written as $f_s : S \rightarrow \mathbb{R}^2$, where S is the unit square in $\mathbb{R}^2$. Its basic operation (see Fig. 2.10) is that of contracting S in the x-direction, expanding it in the y-direction, folding the result (which is in the shape of a horseshoe) and placing this result back over S. Note how pieces of S fall outside of S under the action of f_s and how the horizontal rectangles H_0 and H_1 become the vertical ones V_0 and V_1 respectively. By repeating iterations under f_s, which corresponds to evolving the discrete dynamics generated by f_s and retaining only those points in S that remain invariant under f_s , one arrives at a very complex set of points in S (in the limit of an infinite number of iterations) that is the Smale horseshoe. This set is reminiscent of a 2-D version of the familiar Cantor middle-thirds set.

Thus, the basic intuitive idea behind the Shilnikov method is then to show that the Shilnikov map behaves qualitatively the same as the map f_s, thereby ensuring the existence of the Smale horseshoe in the map's discrete dynamics - and hence finally *chaos* in the original third-order continuous dynamical system. We will now state Shilnikov's theorem and prove Theorem 1.

2.4 Rigorous proof of chaos via Shilnikov's Theorem

We will provide a rigorous proof of chaos in this section and then show the bifurcationan analysis in the next section. Shilnikov's theorem [8] is stated below:

Theorem 1 (Shilnikov's Theorem). *Consider* $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$*,* $\mathbf{x} \in \mathbb{R}^3$ *where* $\mathbf{f} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$*. If the following conditions are satisfied:*

1. *The origin is an equilibrium point* x_e *and* x_e *is a hyperbolic saddle.i.e..the eigenvalues of the Jacobian are of the form* γ*,* $\sigma \pm j\omega$*,* $\sigma\gamma < 0$*,* $\omega \neq 0$*,* $\gamma > |\sigma| > 0$

2. *There exists a homoclinic orbit based at* x_e*.*

Then, an infinitesimal perturbation of this homoclinic orbit will lead to a countable set of horseshoes on that orbit.

Our approach in using Shilnikov's theorem is quite general and can be applied to a much larger class of piecewise-linear differential equations (of which Eq. 2.4 is a special case). We first define a set of *normalized eigenvalue parameters* in terms of the eigenvalues of the corresponding linear system in Eq. 2.4. Next, we obtain *symbolic expressions* for the normalized eigenvalue parameters in terms of system parameters $(\alpha, \beta, \gamma^*, m_1, m_0)$ from Eq. 2.4. These symbolic expressions will then be used to prove condition 1 in Shilnikov's theorem. Next, we will use the normalized eigenvalue parameters along with a transformation of Eq. 2.4 to the Jordan form to derive *exact parametric equations* for Poincare map(s). These Poincare map(s) will be used to show the existence of homoclinic orbits in Eq. 2.4 thereby satisfying condition 2 of Shilnikov's theorem. After the rigorous proof of chaos, bifurcation analysis will be carried out using the describing function method [3].

First, definitions related to the proof of condition 1 in Shilnikov's theorem are summarized next.

2.4.1 Definitions for proving that x_e is a hyperbolic saddle

Please refer to Fig. 2.11 for a pictorial representation of the definitions below. Note that we have computed expressions for the definitions below relevant to Fig. 2.11 using the parameters for the Four-Element Chua's circuit from Theorem 1.

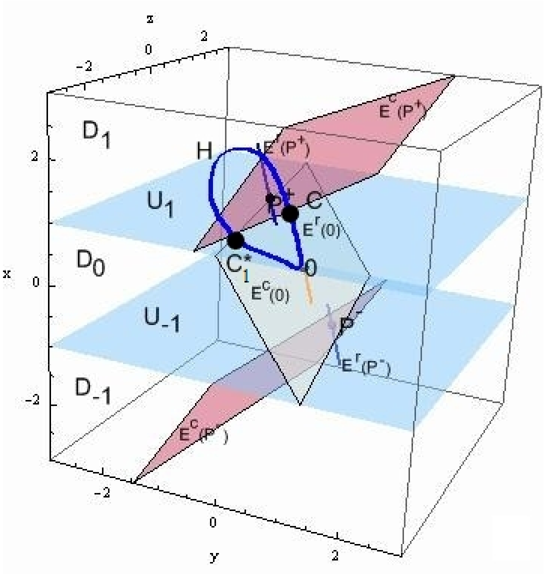


Figure 2.11: Plot of stable and unstable eigenspaces associated with the Jacobian in each region for the Four-Element Chua's circuit. Also shown is a homoclinic orbit H. This will be discussed in the section on homoclinic orbits at x_e.

Definition 1. *There are two planes U_1 and U_{-1} which are symmetric with respect to the origin and they partition $\mathbb{R}^3$ into three closed regions D_1, D_0 and D_{-1} as shown in Fig. 2.11.*

Definition 2. *Let ξ be the continuous vector field:*

$$\xi(\mathbf{x}) \triangleq \begin{bmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \\ f_3(\mathbf{x}) \end{bmatrix} \triangleq \begin{bmatrix} \alpha(y - f(x)) \\ x - \gamma^* y + z \\ -\beta y \end{bmatrix} \tag{2.6}$$

From Eq. 2.4, $f(x) = m_1 x + \frac{1}{2}(m_0 - m_1)(|x+1| - |x-1|)$.

Definition 3. *ξ has three equilibrium points, one at the origin O, one in the interior of D_1 (labeled P^+) and one in the interior of D_{-1} (labeled P^-). For $\gamma^* = 0$ (the Four-Element Chua's circuit), O is (0,0,0), P^+ is $(1.16, 0, -1.16)$ and P^- is $(-1.16, 0, 1.16)$.*

Definition 4. *In each region D_i ($i = -1, 0, 1$), the vector field (ξ) is affine, i.e.,*

$$D\xi(x, y, z) = M_i \; for \; (x, y, z) \in D_i \tag{2.7}$$

where $D\xi$ deontes the Jacobian matrix of $\xi(\mathbf{x})$ and M_i denotes 3x3 real constant matrix.

Definition 5. *Each matrix M_i has a pair of complex conjugate eigenvalues (labeled $\tilde{\sigma_0} \pm j\tilde{\omega_0}$ for M_0 and $\tilde{\sigma_1} \pm \tilde{\omega_1}$ for M_{-1} and M_1 where $\tilde{\omega_0} > 0$ and $\tilde{\omega_1} > 0$) and a real eigenvalue (labeled $\tilde{\gamma_0}$ for M_1 and $\tilde{\gamma_1}$ for M_{-1} and M_1, where $\tilde{\gamma_0} \neq 0$ and $\tilde{\gamma_1} \neq 0$).*

Definition 6. $E^c(0) \stackrel{\triangle}{=}$ *2-D eigenspace cooresponding to complex eigenvalue $\tilde{\sigma_0} \pm j\tilde{\omega_0}$ at 0. In Fig. 2.11, $E^c(0) \stackrel{\triangle}{=} 9.3x + 12.9y + 2.4z = 0$.*
$E^r(0) \stackrel{\triangle}{=}$ *1-D eigenspace corresponding to real eigenvalue $\tilde{\gamma_0}$ at 0. In Fig. 2.11, $E^r(0) \stackrel{\triangle}{=} t \cdot (-0.817, -0.045, 0.57), t \in \mathbb{R}$.*
$E^c(P^+) \stackrel{\triangle}{=}$ *2-D eigenspace corresponding to complex eigenvalue $\tilde{\sigma_1} \pm j\tilde{\omega_1}$ at P^+. In Fig. 2.11, $E^c(P^+) \stackrel{\triangle}{=} 0.1x - 0.153y + 0.017z = 0.129$.*
$E^r(P^+) \stackrel{\triangle}{=}$ *1-D eigenspace corresponding to real eigenvalue $\tilde{\gamma_1}$ at P^+. In Fig. 2.11, $E^r(P^+) \stackrel{\triangle}{=} (1.16, 0, -1.16) + t \cdot (-0.903, 0.054, 0.425), t \in \mathbb{R}$.*

Definition 7. *We define the following normalized eigenvalue parameters $(\sigma_0, \gamma_0, \sigma_1, \gamma_1, \kappa)$:*

$$\sigma_0 \stackrel{\triangle}{=} \frac{\tilde{\sigma_0}}{\tilde{\omega_0}}, \gamma_0 \stackrel{\triangle}{=} \frac{\tilde{\gamma_0}}{\tilde{\omega_0}}, \sigma_1 \stackrel{\triangle}{=} \frac{\tilde{\sigma_1}}{\tilde{\omega_1}}, \gamma_1 \stackrel{\triangle}{=} \frac{\tilde{\gamma_1}}{\tilde{\omega_1}}, \kappa \stackrel{\triangle}{=} \frac{\tilde{-\gamma_0}}{\tilde{\gamma_1}} \tag{2.8}$$

Definition 8. *Some important line segments (not shown in Fig. 2.11 for clarity purposes):*

$$L_0 \stackrel{\triangle}{=} U_1 \cap E^c(0), L_1 \stackrel{\triangle}{=} U_1 \cap E^c(P^+), L_2 \stackrel{\triangle}{=} \{\mathbf{x} \in U_1 : \xi(\mathbf{x}) \mid\mid U_1\} \tag{2.9}$$

Definition 9. *The following points called fundamental points of ξ will play an important role in the proof of chaos:*

$$\begin{aligned} A \stackrel{\triangle}{=} L_0 \cap L_1, B \stackrel{\triangle}{=} L_1 \cap L_2, C \stackrel{\triangle}{=} U_1 \cap E^r(0), D \stackrel{\triangle}{=} U_1 \cap E^r(P^+), \\ E \stackrel{\triangle}{=} L_0 \cap L_2, F \stackrel{\triangle}{=} \{\mathbf{x} \in L_2 : \xi(\mathbf{x}) \mid\mid L_2\} \end{aligned} \tag{2.10}$$

Now that we have these definitions in hand, we can prove Theorem 1 stated in this chapter. We will first prove condition 1 of Shilnikov's theorem (Theorem 1) using our eigenvalues and normalized eigenvalue parameters.

2.4.2 Proving that x_e is a hyperbolic saddle

We will use the abbreviated notation:

$$\lambda \uparrow a \leq \lambda \leq b \; (resp., \; \lambda \downarrow in \; b \geq \lambda \geq a) \tag{2.11}$$

to mean the variable $\lambda = \lambda(\gamma^*)$ increases (resp., decreases) monotonically and satisfies $a \leq min(\lambda) \leq max(\lambda) \leq b$ as γ^* increases monotonically in the range J in Theorem 1.

Lemma 1. *As γ^* increases monotonically from $\gamma_1^* = 0$ to $\gamma_2^* = 0.75$ in Theorem 1, the following parameters also vary monotonically as indicated:*

$$\begin{aligned} \tilde{\sigma_0} \downarrow \; in \; & -0.69275 \geq \tilde{\sigma_0} \geq -1.037 \\ \tilde{\omega_0} \downarrow \; in \; & 7.15435 \geq \tilde{\omega_0} \geq 6.8972 \\ \tilde{\gamma_0} \downarrow \; in \; & 5.4239 \geq \tilde{\gamma_0} \geq 5.36324 \end{aligned} \tag{2.12}$$

$$\begin{aligned} \tilde{\sigma_1} \downarrow \; in \; & 0.7562 \geq \tilde{\sigma_1} \geq 0.416375 \\ \tilde{\omega_1} \uparrow \; in \; & 7.5456 \leq \tilde{\omega_1} \leq 7.92795 \\ \tilde{\gamma_1} \downarrow \; in \; & -8.832 \geq \tilde{\gamma_1} \geq -8.90235 \end{aligned} \tag{2.13}$$

$$\begin{aligned} \sigma_0 \downarrow \; in \; & -0.0968298 \geq \sigma_0 \geq -0.15047 \\ \sigma_1 \downarrow \; in \; & 0.100217 \geq \sigma_1 \geq 0.0525199 \\ \gamma_0 \uparrow \; in \; & 0.758126 \leq \gamma_0 \leq 0.778215 \\ \gamma_1 \uparrow \; in \; & -1.17048 \leq \gamma_1 \leq -1.12291 \end{aligned} \tag{2.14}$$

$$\kappa \downarrow \; in \; 0.614119 \geq \kappa \geq 0.602452 \tag{2.15}$$

Moreover, the above bounds can be calculated to any desired accuracy.

Proof. It follows from Eq. 2.4 that the real eigenvalue $\tilde{\gamma_i}$ corresponding to $m = m_i$ $(i = 0, 1)$ is a real root of the characteristic polynomial equation:

$$s^3 + (\alpha \cdot m + \gamma^*) \cdot s^2 + (\alpha \cdot m \cdot \gamma^* - \alpha + \beta) \cdot s + \alpha\beta m = 0 \tag{2.16}$$

Solving Eq. 2.16 for γ^*, we obtain:

$$\gamma^* = \gamma^*(s) \overset{\triangle}{=} -s + \frac{1}{s + \alpha \cdot m}\left(\alpha - \beta - \frac{\alpha\beta m}{s}\right) \tag{2.17}$$

For the parameter range from Theorem 1 we find from Eq. 2.17 above that $\tilde{\gamma_0}$, $\tilde{\gamma_1}$ decrease and they satisfy:

$$\begin{aligned} \tilde{\gamma_0} \downarrow \; in \; & 5.4239 \geq \tilde{\gamma_0} \geq 5.36324 \\ \tilde{\gamma_1} \downarrow \; in \; & -8.832 \geq \tilde{\gamma_1} \geq -8.90235 \end{aligned} \tag{2.18}$$

Now the solutions of Eq. 2.16 (i.e., the eigenvalues in each region) are related to its coefficients as follows:

$$\begin{aligned} 2\tilde{\sigma_i} + \tilde{\gamma_i} &= -(\alpha m_i + \gamma^*) \\ \tilde{\sigma_i}^2 + \tilde{\omega_i}^2 + 2\tilde{\sigma_i}\tilde{\gamma_i} &= \alpha(m_i\gamma^* - 1) + \beta \\ \tilde{\gamma_i}(\tilde{\sigma_i}^2 + \tilde{\omega_i}^2) &= -\alpha\beta m_i \end{aligned} \tag{2.19}$$

Solving for $\tilde{\sigma_i}$ and $\tilde{\omega_i}^2$ from the equation above we obtain for $i = 0, 1$

$$\begin{aligned} \tilde{\sigma_i} &= \frac{-1}{2}(\alpha m_i + \gamma^* + \tilde{\gamma_i}) \\ \tilde{\omega_i}^2 &= \frac{-1}{4}(\alpha m_i - \gamma^* - \tilde{\gamma_i})^2 - \frac{\alpha^2 m_i}{\tilde{\gamma_i} + \alpha m_i} \end{aligned} \tag{2.20}$$

Combining Eq. 2.18 and Eq. 2.20 we obtain properties 2.12 and 2.13. Properties 2.14 and 2.15 follow from properties 2.12, 2.13 and the definition of normalized eigenvalues in Eq. 2.8. Finally note that the bounds in properties 2.12 to 2.15 can be calculated to be exact to any number of digits because Eqs. 2.17 and 2.20 are rational expressions. □

Lemma 2. *Theorem 1 satisfies hypothesis 1 of Shilnikov's theorem 1.*

Proof. Property 2.12 (proved above) implies:

$$-3.75741 \geq \tilde{\sigma_0}\tilde{\gamma_0} \geq -5.56168 \tag{2.21}$$

Also, $\tilde{\gamma_0} > |\tilde{\sigma_0}| > 0$. Thus, hypothesis 1 of Shilnikov's Theorem is satisfied by Theorem 1. □

2.4.3 Proving that a homoclinic orbit exists at x_e and related definitions

Consider trajectory H shown in Fig. 2.11 through the origin and moving upward along the unstable eigenvector $E^r(0)$ until it hits U_1 at point C. Under the dynamics in the D_1 region, C is taken to a point C_1^*. If we can prove that C_1^* (which would be the first return point of the map ψ_H discussed at the beginning of this chapter) lies on the stable eigenspace $E^c(0)$ of the origin, we have found a homoclinic orbit [8]. Unfortunately, the coordinates of the return point C_1^* cannot be calculated explicitly because it would involve solving a pair of transcendental equations [8]. But by working in the Jordan space, the coordinates of C_1^* (and other necessary points) can be calculated. Computing these points and the necessary return maps are the main topics of this section.

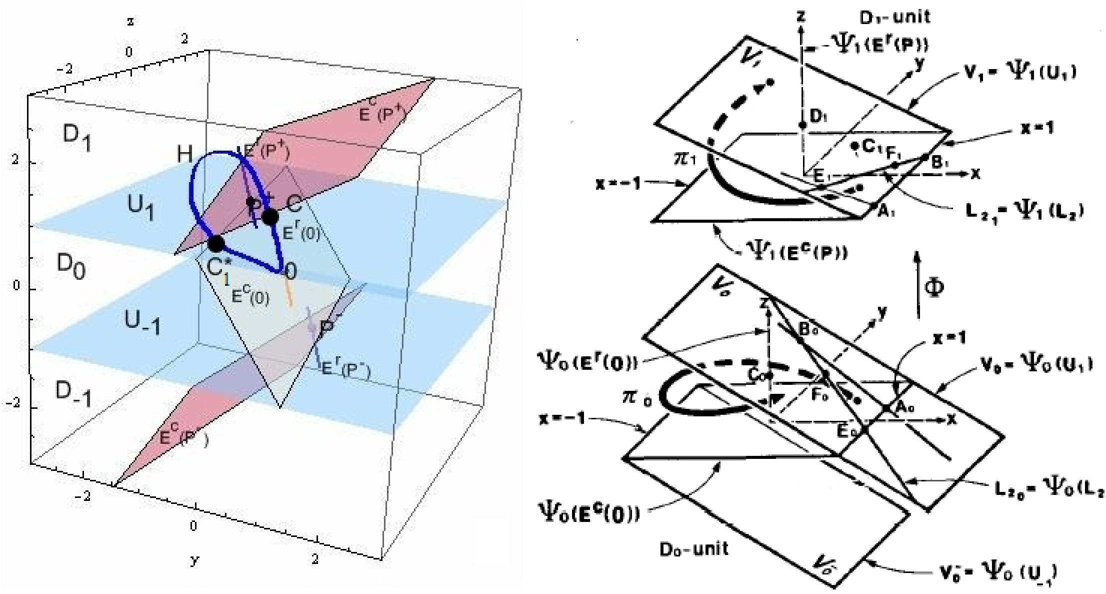


Figure 2.12: Geometrical structure and typical trajectories of our system in the Jordan space. Half-return maps are also shown. Note that for simplicity, for the Jordan space, we have reused the figure from [8].

Further definitions are necessary for this section, refer to Fig. 2.12 as we go through the definitions below.

Definition 10. *Let $\psi_{\mathbf{0}}$ and $\psi_{\mathbf{1}}$ be the appropriate affine transformations that reduce the matrix representation of ξ in D_0 and D_1 to the Jordan form. Specifically,*

1. $\psi_{\mathbf{0}} : D_0 \to \mathbb{R}^3$. *Thus, we have the following transformations:*

   (a) $\psi_{\mathbf{0}}(0) = 0,\ \psi_{\mathbf{0}}(U_1) = V_0 \triangleq \{(x, y, z) : x + z = 1\}$

   (b) $\psi_{\mathbf{0}}(U_{-1}) = V_0^- \triangleq \{(x, y, z) : x + z = -1\}$

   (c) $\frac{1}{\tilde{\omega}_0} D\psi_{\mathbf{0}}\left(\xi(\psi_{\mathbf{0}}^{-1}(\mathbf{x}))\right) = \xi_{\mathbf{0}}(\mathbf{x}) \triangleq \begin{pmatrix} \sigma_0 & -1 & 0 \\ 1 & \sigma_0 & 0 \\ 0 & 0 & \gamma_0 \end{pmatrix} \cdot \mathbf{x}$. *Note that the matrix above is M_0, the Jordan form of ξ in D_0.*

2. $\psi_{\mathbf{1}} : D_1 \to \mathbb{R}^3$. *Thus, we have the following transformations:*

   (a) $\psi_{\mathbf{1}}(P) = 0,\ \psi_{\mathbf{1}}(U_1) = V_1 \triangleq \{(x, y, z) : x + z = 1\}$

(b) $\frac{1}{\tilde{\omega_1}} D\psi_{\mathbf{1}}\left(\xi(\psi_{\mathbf{1}}^{\mathbf{-1}}(\mathbf{x}))\right) = \xi_{\mathbf{1}}(\mathbf{x}) \overset{\triangle}{=} \begin{pmatrix} \sigma_1 & -1 & 0 \\ 1 & \sigma_1 & 0 \\ 0 & 0 & \gamma_1 \end{pmatrix} \cdot \mathbf{x}$. *Note that the matrix above is M_1, the Jordan form of ξ in D_1.*

We will call $\xi_{\mathbf{0}}(\mathbf{x})$ and $\xi_{\mathbf{1}}(\mathbf{x})$ as the normalized Jordan forms of M_0 and M_1 respectively. In other words, via this definition, we have defined two local coordinate systems (the D_0 unit and D_1 unit in Fig. 2.12). We will abuse notation and label both coordinate systems as (x, y, z).

Definition 11. *Now the images of the fundamental points defined earlier will be labeled in an obvious way:*

$$\begin{aligned} D_0 : A_0 = \psi_0(A),\ B_0 = \psi_0(B),\ C_0 = \psi_0(C) \\ D_0 = \psi_0(D),\ E_0 = \psi_0(E),\ F_0 = \psi_0(F) \end{aligned} \tag{2.22}$$

$$\begin{aligned} D_1 : A_1 = \psi_1(A),\ B_1 = \psi_1(B),\ C_1 = \psi_1(C) \\ D_1 = \psi_0(D),\ E_1 = \psi_1(E),\ F_1 = \psi_1(F) \end{aligned} \tag{2.23}$$

Now we have to compute the coordinates of the points above. Since the points A through F are located on the intersections of various lines (Definition 8 and Definition 9), their images must lie on the corresponding lines in the new reference frame. Computing the coordinates of these points is an exercise in linear algebra, here we list only the coordinates of these points.

Definition 12. *Points in the D_0 unit:*

$$A_0 = (1, p_0, 0),\ p_0 \overset{\triangle}{=} \sigma_0 + \frac{\kappa}{\gamma_0}(\sigma_0^2 + 1) \tag{2.24}$$

$$B_0 = \left(\frac{\gamma_0(\gamma_0 - \sigma_0 - p_0)}{Q_0}, \frac{\gamma_0(1 - p_0(\sigma_0 - \gamma_0))}{Q_0}, \frac{1 - \gamma_0(\gamma_0 - \sigma_0 - p_0)}{Q_0}\right)$$
$$Q_0 = (\sigma_0 - \gamma_0)^2 + 1 \tag{2.25}$$

$$C_0 = (0, 0, 1) \tag{2.26}$$

$$E_0 = (1, \sigma_0, 0) \tag{2.27}$$

$$F_0 = \left(\frac{\gamma_0(\gamma_0 - 2\sigma_0)}{Q_0}, \frac{\gamma_0(1 - \sigma_0(\sigma_0 - \gamma_0))}{Q_0}, \frac{\sigma_0^2 + 1}{Q_0}\right) \tag{2.28}$$

Definition 13. *Strategic points in the D_1 unit:*

$$A_1 = (1, p_1, 0),\ p_1 \overset{\triangle}{=} \sigma_1 + \frac{1}{\kappa\gamma_1}(\sigma_1^2 + 1) \tag{2.29}$$

$$B_1 = (1, \sigma_1, 0) \tag{2.30}$$

$$\begin{aligned} C_1 &= (x_c, y_c, 0) \\ x_c &= \frac{1 - (\sigma_1^2 + 1)((\sigma_0 + \frac{\gamma_0}{\kappa}) + 1)}{(\sigma_0^2 + 1)Q_1} \\ y_c &= \frac{\gamma_1(1 - \sigma_1(\sigma_1 - \gamma_1))}{Q_1} - \frac{\frac{(\sigma_1^2+1)\gamma_0}{\kappa}}{(\sigma_0^2 + 1)\gamma_1 Q_1} \end{aligned} \tag{2.31}$$

$$E_1 = \left(\frac{\gamma_1(\gamma_1 - \sigma_1 - p_1)}{Q_1}, \frac{\gamma_1(1 - p_1(\sigma_1 - \gamma_1))}{Q_1}, \frac{1 - \gamma_1(\gamma_1 - \sigma_1 - p_1)}{Q_1}\right)$$
$$Q_1 = (\sigma_1 - \gamma_1)^2 + 1 \tag{2.32}$$

$$F_1 = \left(\frac{\gamma_1(\gamma_1 - 2\sigma_1)}{Q_1}, \frac{\gamma_1(1 - \sigma_1(\sigma_1 - \gamma_1))}{Q_1}, \frac{\sigma_1^2 + 1}{Q_1}\right) \tag{2.33}$$

Notice that some of the points in the D_0 *unit* and D_1 *unit* have $z = 0$. This means that we actually have projected the point onto the $x - y$ plane in Jordan space, this obviously simplifies computations. We will also be able to write expressions for the Poincare maps π_1 and π_0 in Fig. 2.12 (these correspond to ψ_h and ψ_e in Fig. 2.9 respectively) in the Jordan space. To prove the existence of a homoclinic orbit we apply π_1^{-1} to the line segment E_1A_1 since this segment corresponds to the intersection of the stable eigenspace at the origin with the D_1 *unit* ($\psi_1(C_1^*)$ in Fig. 2.12 is a point on E_1A_1). That is, we need $C_1 \in \pi_1^{-1}(E_1A_1)$ (C_1 corredponds to point C in the D_1 unit) for some $\gamma^* \in [0, 0.75]$. Finally, we prove that the trajectory does not leave the D_0 *unit* with the π_0 map. First, the definitions for π_1 and π_0:

Definition 14. *Given* $\mathbf{x_1} \overset{\triangle}{=} (x_1, y_1)^T \in \Delta A_1B_1E_1 \rightarrow V_1$*, the half-return map* $\pi_1(\mathbf{x_1})$ *is given by:*

$$\pi_1(\mathbf{x_1}) = e^{\sigma_1\tau_1}\begin{pmatrix} cos(\tau_1) & -sin(\tau_1) \\ sin(\tau_1) & cos(\tau_1) \end{pmatrix} \cdot \mathbf{x_1} \tag{2.34}$$

Here τ_1 is the first return time for π_1

Definition 15. *Similarly, given* $\mathbf{x_0} \overset{\triangle}{=} (x_0, y_0)^T \in \Delta A_0B_0E_0 \rightarrow V_0$*, the half-return map* $\pi_0(\mathbf{x_0})$ *is given by:*

$$\pi_0(\mathbf{x_0}) = e^{\sigma_0\tau_0}\begin{pmatrix} cos(\tau_0) & -sin(\tau_0) \\ sin(\tau_0) & cos(\tau_0) \end{pmatrix} \cdot \mathbf{x_0} \tag{2.35}$$

Here τ_0 is the first return time for π_0

Lemma 3. *C_1 defined in Eq. 2.31 is a continuous function of γ^* for the range given in Theorem 1.*

Proof. Since $\tilde{\gamma_i}$ is a continuous function of γ^* in view of Eq. 2.17, it follows from Eq. 2.20 that $\tilde{\sigma_i}$, $\tilde{\omega_i}$ and κ are also continuous functions of γ^* for $i = 0, 1$. Since $C_1 = (x_c, y_c)$ is given in Eq. 2.31, C_1 is also a continuous function of γ^*. □

Lemma 4. *$C_1 \in \pi_1^{-1}(E_1A_1)$ for some $\gamma^* \in [0, 0.75]$.*

Proof. The coordinates of E_1 and A_1 can be obtained from Eq. 2.32 and Eq. 2.29, since we have already defined and computed the range of the normalized eigenvalues. Using these ranges and the definition of π_1 above, we find that the magnitude of $\pi_1^{-1}(E_1A_1)$ is bounded between:

$$1.021826e^{-0.100217\tau_1} \leq |\pi_1^{-1}(E_1A_1)| \leq 1.13213 \cdot e^{-0.0525199\tau_1} \tag{2.36}$$

The left hand side of the inequality above corresponds to σ_1 from Eq. 2.13 when $\gamma^* = 0$ and the right hand side corresponds to σ_1 from Eq. 2.13 when $\gamma^* = 0.75$. Now, from the definition of C_1 in Eq. 2.31, the range for C_1 is for $\gamma^* \in [0, 0.75]$:

$$0.312167 \leq C_1 \leq 0.394493 \tag{2.37}$$

But,C_1 is continuous from Lemma 3. Therefore, from Eqns. 2.36 and 2.37, we have the necessary result. Notice that we did not have to explicitly calculate the value of τ_1, the first return time. Of course if we need to determine the exact γ^* for the homoclinic orbit, we need the first return time. □

Lemma 5. *For the parameter range in Theorem 1, in the D_0 unit, no trajectory starting from points in the line segment E_0A_0 in the eigenspace $z = 0$ intersects the boundary line $x = -1$.*

Proof. Similar to Lemma 4, we can compute the bounds on $|\pi_0(x)|$, $x \in E_0A_0$ (for $\gamma^* \in [0, 0.75]$) to be:

$$0.81775e^{-0.0968298\tau_0} \geq |\pi_0(x)| \geq 0.794243 \cdot e^{-0.15047\tau_0} \tag{2.38}$$

□

Hence, Eq. 2.38 implies that points on E_0A_0 will never leave the D_0 *unit*.

Lemma 6. *Theorem 1 satisfies hypothesis 2 of Shilnikov's Theorem 1.*

Proof. Lemmas 4 and 5 imply that there exists a $\gamma^* \in [0, 0.75]$ such that the second condition of Shilnikov's theorem is satisfied. □

Hence, Lemma 2 and 6 imply that Shilnikov's theorem is satisfied by Theorem 1.

2.5 Bifurcation analysis using Describing Functions

2.5.1 One Parameter Bifurcation Diagram for the Four-Element Chua's circuit

A bifurcation diagram obtained from Mathematica is shown below. Refer to the appendix for the corresponding Mathematica code.

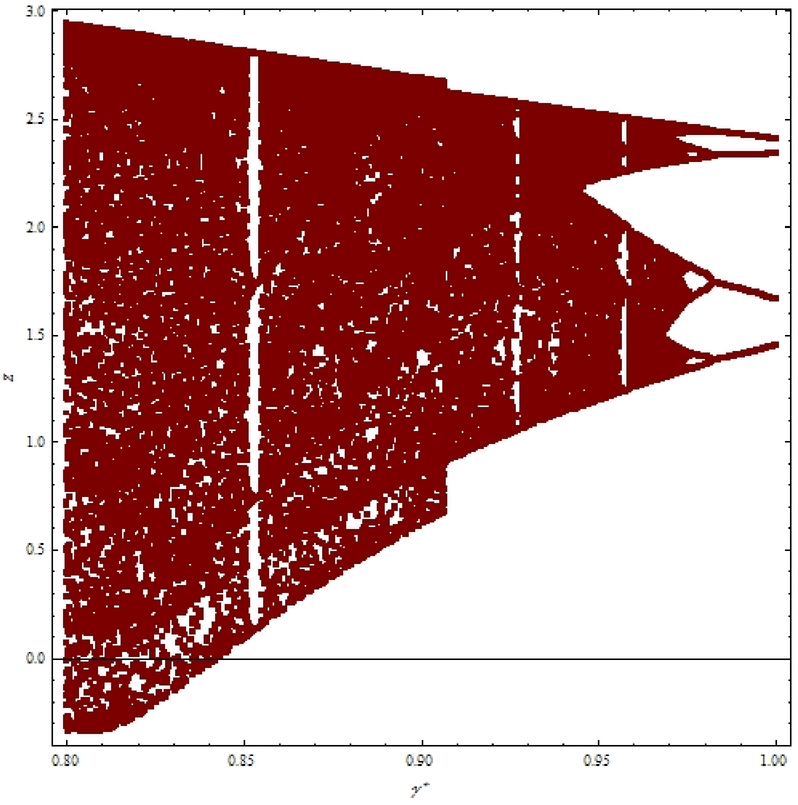


Figure 2.13: The bifurcation diagram showing the onset of period doubling route to chaos as γ^* decreases from 1 to 0.8. Notice the onset of period-doubling bifurcation at $\gamma^* = 1$. This property will be proved rigorously in this section.

Of course, the bifurcation diagram above has not been rigorously derived. The aim of this section is to rigorously justify the period doubling bifurcation above via describing functions [17], [20], [3].

2.5.2 Introduction to Describing Functions

Thorough description of describing functions can be found in [39] and [16]. A quick overview is given here. Let $G(s)$ deonte the laplace transform of the linear element(s) in our system and $N(a)$ be the describing function of the nonlinear element. The definition of the describing function is the complex ratio of the fundamental component of the output of the nonlinear element to the input sinusoid. A block diagram view of a system represented in this form is shown in Fig. 2.14.

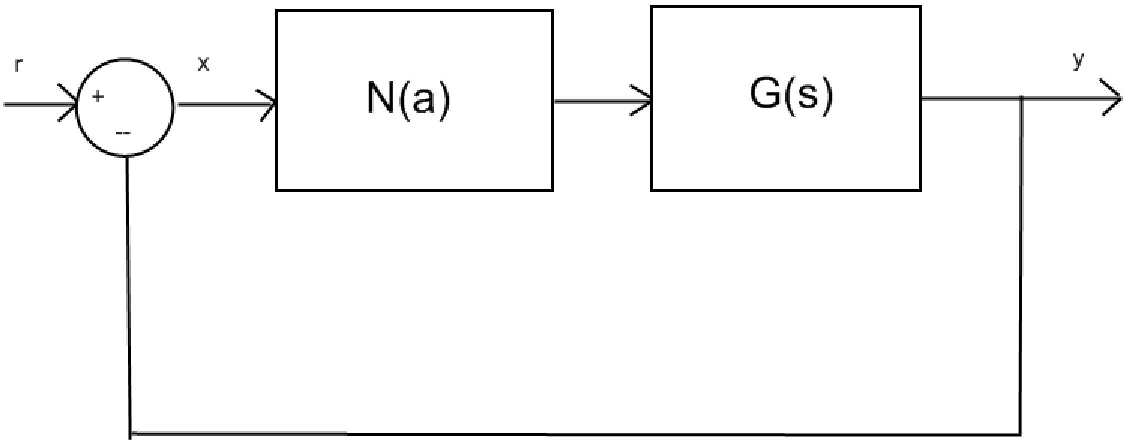


Figure 2.14: Lure form, for autonomous systems $r = 0$.

The above form of the system is called the *Lure Form*. The advantage of representing a system in Lure form is that the following expression is true ($s = j\omega$):

$$1 + N(a)G(j\omega) = 0 \tag{2.39}$$

In other words, we have the following expression involving the describing function and our linear transfer function:

$$G(j\omega) = \frac{-1}{N(a)} \tag{2.40}$$

2.5.3 Period-Doubling Criterion Based on the Describing Function

In order to derive the period-doubling criterion, we will relate the concept of the Loeb Criterion [39] to the concept of a harmonically linearized Poincare map [20]. Suppose a limit-cycle solution $a(t) = a_0 e^{j\omega_0 t}$ exists in Eq. 2.40. Then, the classic Loeb Criterion says that if Δa and $\Delta\sigma$ are perturbations to a_0 and ω_0 resp.,

$$\frac{\Delta a}{\Delta\sigma} = \frac{\left(\frac{\partial U}{\partial\omega}\right)^2 + \left(\frac{\partial V}{\partial\omega}\right)^2}{\frac{U\partial N\partial V}{N\partial a\partial\omega}} \tag{2.41}$$

In Eq. 2.41, $U(\omega_0) + jV(\omega_0) \triangleq G(j\omega_0)$. The describing function N is evaluated at a_0. [20] relates the harmonically linearized Poincare map $P(a_n)$ to the Loeb Criterion:

$$\frac{\partial P}{\partial a} = 1 + \frac{\frac{2\pi a_0 \partial N \partial V}{\omega_0 N^2 \partial a \partial \omega}}{\left(\frac{\partial U}{\partial \omega}\right)^2 + \left(\frac{\partial V}{\partial \omega}\right)^2} \tag{2.42}$$

Feigenbaum's criterion for period-doubling route to chaos in harmonically linearized Poincare maps is [15]:

$$\frac{\partial P}{\partial a} < -1 \; when \; P(a) \; has \; a \; hump \; characteristic \tag{2.43}$$

Now, it turns out that the hump characteristic for $P(a)$ can be replaced by a constraint of piecewise-linearity on $N(a)$ [20]. Substituting Eq. 2.43 in Eq. 2.42 for $\frac{\partial P}{\partial a}$, we get the following period-doubling criterion in terms of $G(j\omega_0$ and $N(a_0)$:

$$\frac{\partial V}{\partial \omega}\frac{\partial N}{\partial a} < -\left(\frac{\left(\frac{\partial U}{\partial \omega}\right)^2 + \left(\frac{\partial V}{\partial \omega}\right)^2}{\frac{\pi}{\omega_0}\frac{a_0}{N^2}}\right) \tag{2.44}$$

2.5.4 Period-Doubling Route to Chaos in the Four-Element Chua's circuit

Proof. Based on Eq. 2.44, the following steps are necessary to confirm period-doubling route to chaos in the Four-Element Chua's circuit.

1. Rewrite the Four-Element Chua's circuit in Lure form so that we can obtain $N(a)$ and $G(j\omega)$ in Eq. 2.40.

2. Determine a period-1 limit cycle for the given choice of parameters.

3. Check if Eq. 2.44 is satisfied for (1) and (2) above.

Rewriting the Four-Element Chua's circuit in Lure form, we obtain the following expression for $G(j\omega)$:

$$G(s) \mid_{s=j\omega} = \frac{\alpha(s^2 + \gamma^* s + \beta)}{s^3 + (\alpha + 1)s^2 + \beta\gamma^* s + \alpha\beta} \tag{2.45}$$

Fortunately, $N(a)$ for the nonlinearity in the Four-Element Chua's circuit can be easily determined from [16]:

$$N(a) = \begin{cases} m_0 & |a| \leq 1 \\ 2\left(\frac{m_0 - m1}{\pi}\right) \cdot \left(\sin^{-1}\left(\frac{1}{a}\right) + \frac{1}{a}\sqrt{1 + \left(\frac{1}{a}\right)^2}\right) & |a| > 1 \end{cases} \tag{2.46}$$

Notice that our describing function is purely real, this is a consequence of our nonlinearity [39]. The first step is to find a_0 and ω_0. This can be done by equating the real and imaginary parts of the expressions in Eqs. 2.45 and 2.46. The parameters are given in Result 1. Although the expressions above are *exact*, we can only solve them numerically. Doing so, we get $a_0 \approx 1.4$ and $\omega_0 \approx 8.149\ rad/s$. Using these values, the LHS of Eq. 2.44 becomes approximately 16.998 and the RHS of Eq. 2.44 becomes approx 37.63. Since $16.998 < 37.63$, the period-doubling criterion is satisfied. □

2.6 Conclusion

In this chapter, we studied the Four-Element Chua's circuit in rigorous detail. We particularly provided a rigorous proof of chaos via circuit analysis in the Jordan space. Bifurcation analysis was carried out using describing function techniques.

The next chapter moves on to higher dimensional chaotic circuits using the memristor, which are derived from the classical Chua's circuit.

Chapter 3

Memristor Based Chaotic Circuits and Topological Horseshoe Theory

3.1 Introduction and Chapter Organization

In this chapter, we will go through higher dimensional chaotic circuits based on the memristor. We will also supply rigorous verification of chaos in such systems via topological horseshoe theory.

An example of a memristor based chaotic circuit can be found in [21] . However, although this circuit is *based* on the canonical Chua's circuit, it does not have the classic three-segment piecewise-linear nonlinearity with a negative slope in each segment. In contrast, this thesis *derives* a four dimensional memristor based chaotic circuit from the classical Chua's circuit (in section three). Another feature of this system is that very few parameters are changed between the classical Chua's circuit and the four dimensional memristor based chaotic circuit. We also derive three other circuits from the four dimensional memristor based chaotic circuit, these circuits are the subject of sections four, five and six. Empirical evidence of chaos is provided via Lyapunov exponents in section seven. Section eight provides a rigorous verification of chaos in the canonical circuit via topological horseshoe theory. The chapter concludes by proposing a breadboard implementation of this circuit.

3.2 Memristor Based Chaotic Circuits and Main Result

The memristor was postulated as the fourth circuit element by Leon O. Chua in 1971 [5]. It thus took its place along side the rest of the more familiar circuit elements such as the resistor, capacitor, and inductor. The common thread that binds these four elements together as the four basic elements of circuit theory is the fact that the characteristics of these elements relate the four variables in electrical engineering (voltage, current, flux and charge) intimately. Fig. 3.1 (reproduced from Chapter 1) shows this relationship graphically [41].

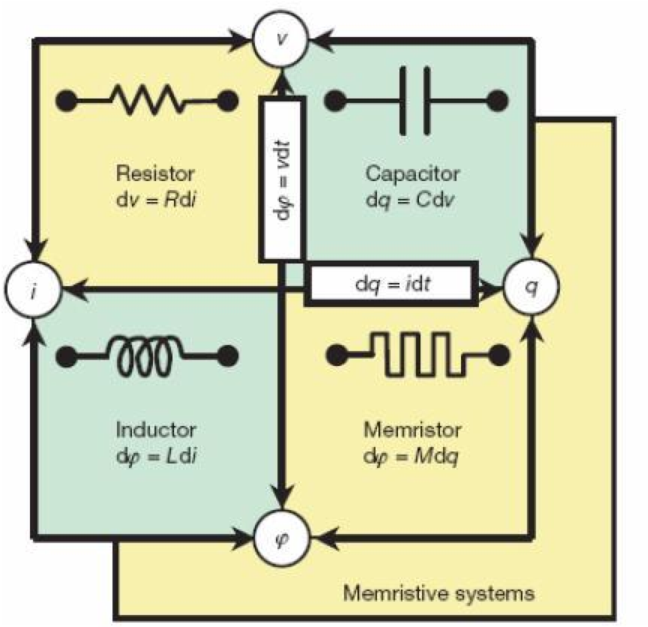


Figure 3.1: The four basic circuit elements

The memristor is a two terminal element, in which the magnetic flux (ϕ) between the terminals is a function of the electric charge that passes through the device [41] [32]. The memristor M used in this work is a flux controlled memristor that is characterized by its incremental menductance [41] function $W(\phi)$ describing the flux-dependent rate of change of charge:

$$W(\phi) = \frac{dq(\phi)}{d\phi} \tag{3.1}$$

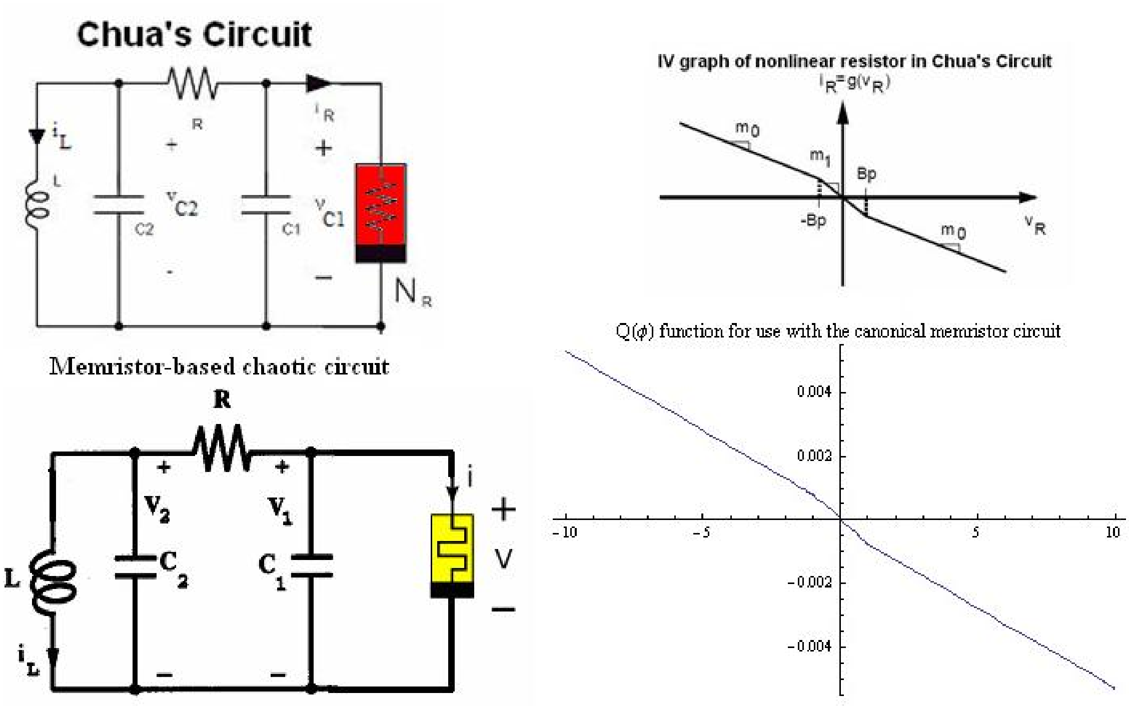
The relationship between the voltage across ($v(t)$) and the current through ($i(t)$) the memristor is thus given by:

$$i(t) = W(\phi(t))v(t) \tag{3.2}$$

Memristor is an acronym for *memory-resistor* because in Eq. 3.2, since $W(\phi(t)) = W(\int v(t))$, the integral operator on the menductance function means the function *remembers* the past history of voltage values. Of course, if $W(\phi(t)) = W(\int v(t)) =$ *constant*, a memristor is simply a resistor. For over thirty years, the memristor was not significant in circuit theory. In 2008, Williams et. al. [41] fabricated a solid state implementation of the memristor and thereby cemented its place as the 4th circuit element. They used two titanium dioxide films, with varying resistance which is dependent on how much charge has been passed through it in a particular direction. As a result of this realization, it is possible to have nonvolatile memory on a nano scale. Since memristive devices function at the nano-scale, high frequency oscillators can possibly be constructed using the memristor [42] and these have potential for applications in secure communication [34].

One of the first memristor based chaotic circuits have been proposed by Itoh and Chua [21]. However, their paper uses a passive nonlinearity based on the memristor characteristics obtained by Williams et. al.. Hence, their circuits are not suitable for secure communications because active nonlinearities are essential for high signal-to-noise ratio (SNR) [14]. The canonical Chua's circuit uses an active nonlinearity for obtaining chaos, and hence it has found a variety of applications ever since inception in 1984 in secure communications [19]. Therefore, a natural question to ask is: can we obtain a memristor based chaotic circuit from the classical Chua's circuit? This chapter answers yes to the question posed above. We obtain a memristor based chaotic circuit by simply increasing the dimensionality of the canonical Chua's circuit. We define simply increasing as adding a memristor with the same nonlinearity as the canonical Chua's circuit. This circuit is the subject of the next section. First, the

Table 3.1: In order to obtain the memristor based chaotic circuit, we replaced the nonlinear resistor in Chua's circuit with a memristor.


Chua's Circuit
R
i_L
L
+
v_C2
C2
-
i_R
+
v_C1
C1
−
N_R
IV graph of nonlinear resistor in Chua's Circuit
i_R=g(v_R)
m0
m1
Bp
-Bp
m0
v_R
Memristor-based chaotic circuit
R
+
+
V_2
V_1
L
C_2
C_1
i_L
i
+
V
−
−
−
Q(φ) function for use with the canonical memristor circuit
0.004
0.002
-10
-5
5
10
-0.002
-0.004

main result from this chapter:

Result 2. ***Table 3.1 is the canonical memristor based chaotic circuit that has been derived from Chua's circuit. This circuit is canonical since other chaotic circuits can be derived from this one. It also shares some similar properties to Chua's circuit, for example, we can replace the piecewise-linear menductance with a smooth nonlinearity and still obtain chaos (refer to the implementation section of this chapter for details). A rigorous verification of chaos in the canonical memristor based chaotic circuit is provided via topological horseshoe theory (refer to the topological horsetheory section of this chapter for details).***

3.3 Canonical Memristor Based Chaotic Circuit

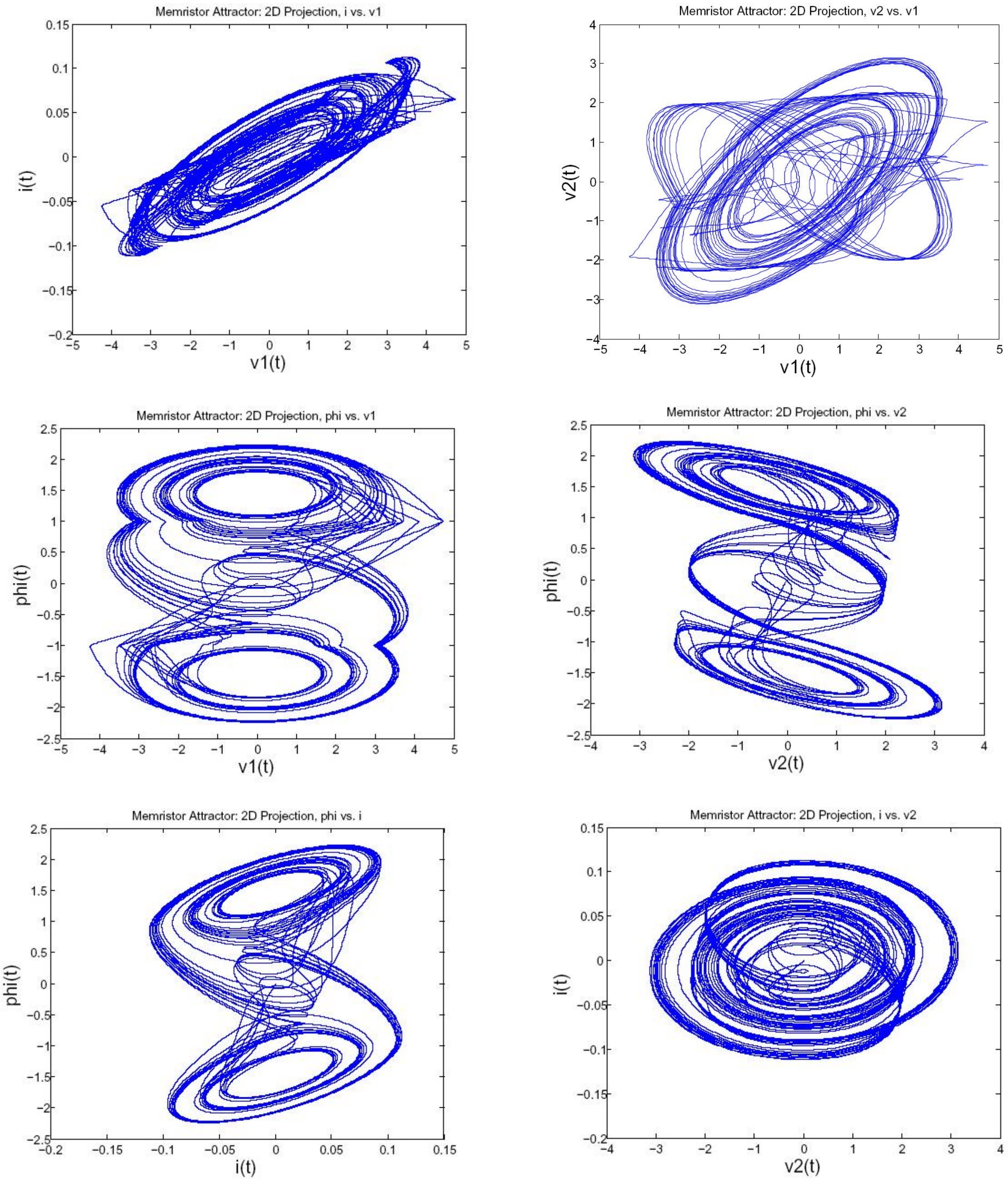
As stated in the introduction section, our first circuit dimensionally extends the canonical Chua's circuit. Refering to Table 3.1, we derived the memristor-based chaotic circuit by simply replacing the nonlinear resistor in Chua's circuit with a flux-controlled memristor. Note that we derived the menductance nonlinearity by using the constraint of hyperbolic equilibrium points (condition 1 of Shilnikov's Theorem [28]). Below are the equations governing our memristor-based chaotic circuit:

$$\begin{aligned}
\frac{d\phi}{dt} &= v_1(t) \\
\frac{dv_1(t)}{dt} &= \frac{1}{C_1}\left(\frac{v_2(t)-v_1(t)}{R} - W(\phi(t))\cdot v_1(t)\right) \\
\frac{dv_2(t)}{dt} &= \frac{1}{C_2}\left(\frac{v_1(t)-v_2(t)}{R} - i_L(t)\right) \\
\frac{di_L(t)}{dt} &= \frac{v_2(t)}{L}
\end{aligned} \tag{3.3}$$

The intuitive justification for dimensional extension is that an active nonlinearity is very important for obtaining a chaotic circuit. The dimensional extension not only preserves the active nonlinearity, it also introduces another nonlinearity in terms of the product $(W(\phi(t))v_1(t))$ in the equation above. These two nonlinearities should combine to give rise to chaos, as we observed. The $Q(\phi)$ function is obtained from the canonical Chua's circuit:

$$Q(\phi) = -0.5\cdot 10^{-3}\cdot\phi + \frac{-0.8\cdot 10^{-3}+0.5\cdot 10^{-3}}{2}\cdot(|(\phi+1)| - |(\phi-1)|) \tag{3.4}$$

Table 3.2: Attractors from the state-scaled canonical memristor-based circuit



The menductance function that is obtained from the $Q(\phi)$ function is:

$$W(\phi) = \frac{dq_m(\phi)}{d\phi} = \begin{cases} -0.5 \cdot 10^{-3} & \phi \leq -1 \\ -0.8 \cdot 10^{-3} & -1 < \phi < 1 \\ -0.5 \cdot 10^{-3} & \phi \geq 1 \end{cases} \tag{3.5}$$

Choosing the parameters as $C_1 = 5.5nF,\ C_2 = 49.5nF,\ L = 7.07mH,\ R = 1428\Omega$ and setting the initial conditions to $\phi(0) = 0,\ v_1(0) = 0,\ v_2(0) = 0,\ i(0) = 0.1$ we can see that this dimensionally extended circuit indeed generates chaotic behaviour. The simulation results in values for the states that are far beyond what is physically realizable. The system states can be rescaled to the appropriate (constrainted) values for $v_1(t)$ and $v_2(t)$. Table 3.2 shows the attractors obtained from the rescaled canonical memristor-based chaotic circuit. The appendix has the simulation code.

In the next section, we try to answer the question of whether we can reduce the number of elements in this circuit and still obtain chaos. Fewer elements in a circuit mean a more compact form for implementation on an integrated circuit. Since one of the greatest advantages of a memristor is a reduction in space (since the device functions at the nanoscale level), it is desirable to exploit this property.

3.4 Four Element Memristor Based Chaotic Circuit

We simplified the circuit from the previous section *a'la* Barboza and Chua's four-element chaotic circuit. This circuit is significant since it is the *simplest* possible circuit in terms of the number of elements and it also displays bifurcation phenomenon not seen in the canonical Chua's circuit. Fig. 3.2 and Fig. 3.3 shows the four Element Chua's circuit and its realization respectively. Fig. 3.4 shows our version of the four-element Chua's circuit in which the nonlinear resistor is replaced by the memristor. Fig. 3.5 shows this circuit reduced down to the basic circuit elements. In order to choose the menductance for the memristor M in Fig. 3.5 we used the hyperbolic equilibrium point constraint from Shilnikov's theorem [28]. The resulting function is

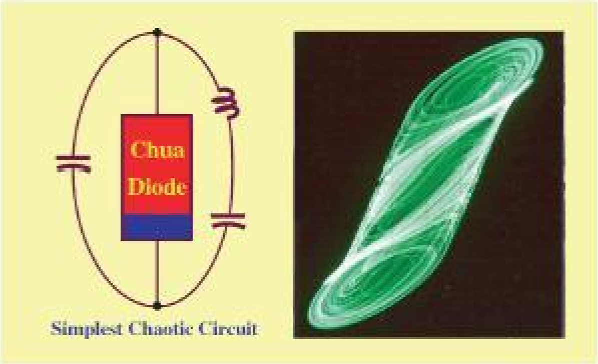


Figure 3.2: The simplest Chua's circuit and its typical attractor [2]

shown in Fig. 3.6. The system equations for Fig. 3.5 are:

$$\begin{aligned} \frac{d\phi}{dt} &= v_1 \\ i - W(\phi)v_1 &= C_1\frac{dv_1}{dt} \\ \kappa L_1\frac{di}{dt} &= v_1 - v_2 \\ i &= \frac{C_2}{\kappa}\frac{dv_2}{dt} \end{aligned} \tag{3.6}$$

The parameters for (3.6) are: $C_1 = 33\,nF$, $C_2 = 100\,nF$, $L_1 = 10\,mH$ and $\kappa = 8.33$. In (3.6), we are free to pick $W(\phi)$. Using MATLAB, we get the results shown in Fig. 3.7 and Fig. 3.8. The appendix has the MATLAB code used to obtain these results. One surprising result we obtained from this circuit is the fact that it can be simplified even further. The next section describes how this may be possible and suggests how the simplification may be advantageous.

3.5 Four Element Memristor Based chaotic circuit with one negative element

Unlike Barboza and Chua's four element chaotic circuit, we discovered that our circuit requires only one negative element. Hence, we can simplify the circuit from the previous section even further. Specifically, in the circuit shown in Fig. 3.5, if we

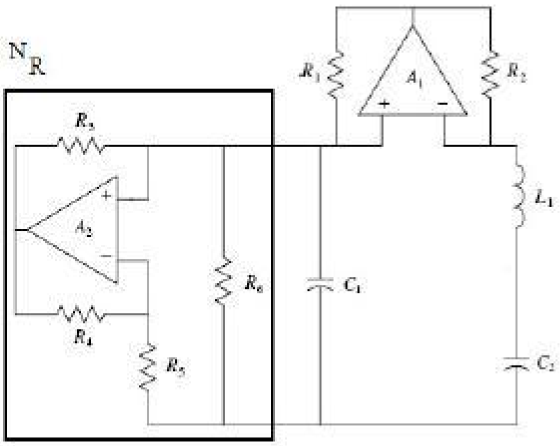


Figure 3.3: A realization of the four-element Chua's circuit [2]

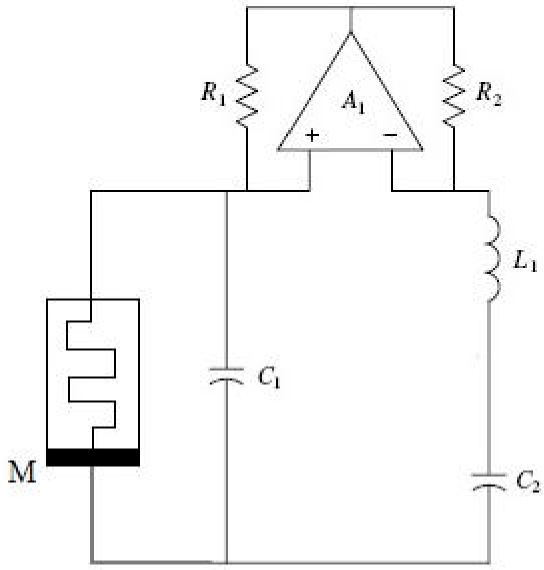


Figure 3.4: Four-element memristor-based chaotic circuit

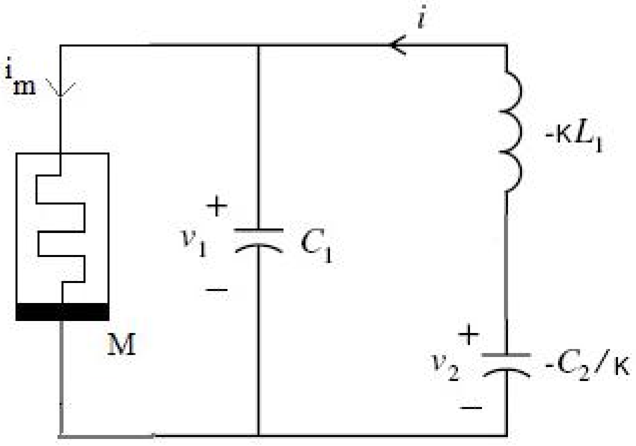
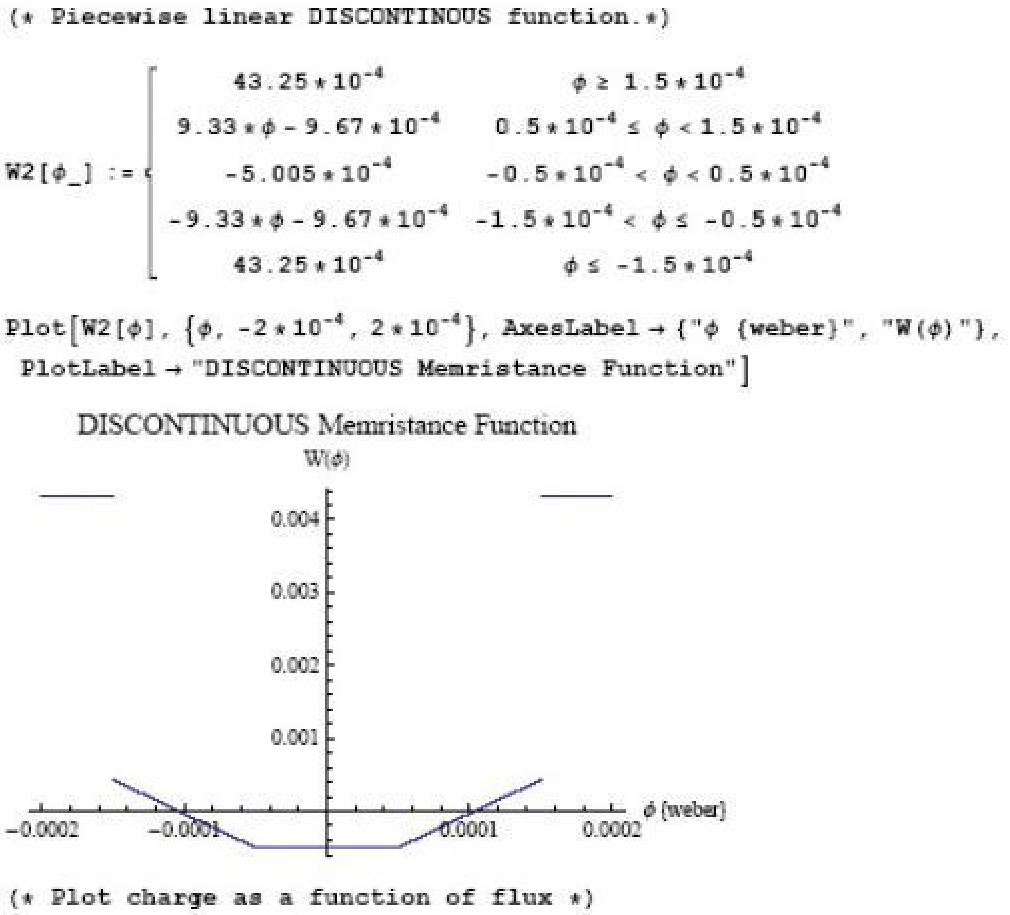


Figure 3.5: Four-element memristor-based chaotic circuit showing only the basic circuit elements. The effect of op-amp A_1 from Fig. 3.4 is the set $-\kappa$.

let C_2 be the only negative element and set κ to 1, we still get chaotic behaviour.



```
(* Piecewise linear DISCONTINOUS function.*)

           ⎧     43.25*10^-4               φ ≥ 1.5*10^-4
           ⎪ 9.33*φ - 9.67*10^-4     0.5*10^-4 ≤ φ < 1.5*10^-4
W2[φ_] :=  ⎨    -5.005*10^-4         -0.5*10^-4 < φ < 0.5*10^-4
           ⎪-9.33*φ - 9.67*10^-4   -1.5*10^-4 < φ ≤ -0.5*10^-4
           ⎩     43.25*10^-4               φ ≤ -1.5*10^-4

Plot[W2[φ], {φ, -2*10^-4, 2*10^-4}, AxesLabel → {"φ {weber}", "W(φ)"},
 PlotLabel → "DISCONTINUOUS Memristance Function"]
```

```
(* Plot charge as a function of flux *)
```

Figure 3.6: Memristance function $W(\phi)$ as defined in Mathematica.

System equations are:

$$\begin{aligned}\frac{d\phi}{dt} &= v_1 \\ i - W(\phi)v_1 &= C_1\frac{dv_1}{dt} \\ \kappa L_1\frac{di}{dt} &= v_1 - v_2 \\ i &= \frac{-C_2}{\kappa}\frac{dv_2}{dt}\end{aligned} \tag{3.7}$$

The other parameters, initial conditions and the menductance function $(W(\phi))$ are the same as the previous section. The simulation results are shown in Fig. 3.9 and Fig. 3.10, MATLAB simulation code is given in the appendix. Having a single negative element is advantageous because we have reduced the number of active elements in

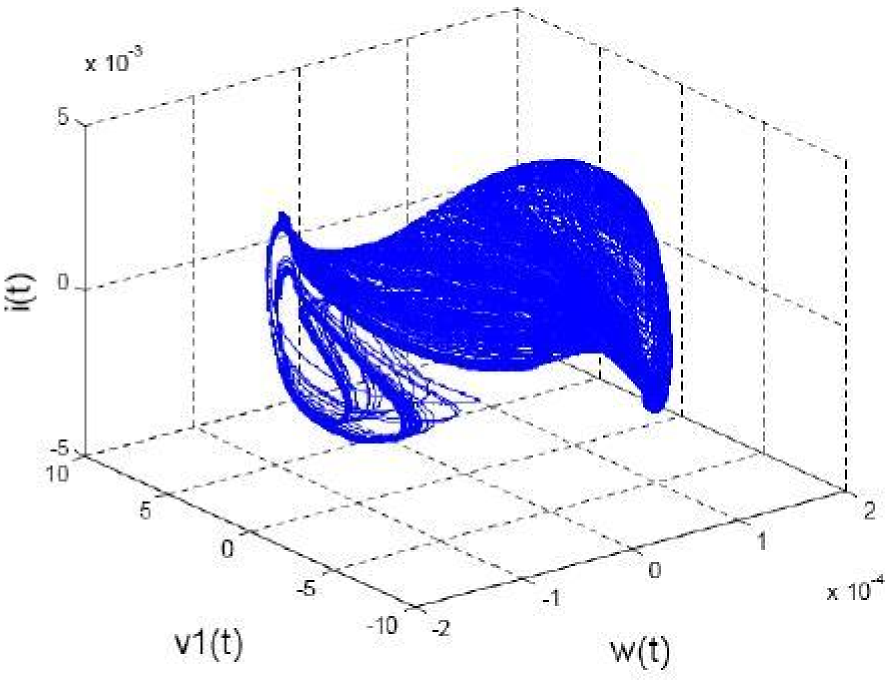


Figure 3.7: 3D attractor from the four-element memristor-based chaotic circuit.

the circuit. This leads to a reduction in power consumption.

Although the above circuit and the two circuit(s) from the previous section(s) seem to exhibit chaotic attractors via simulation, a strong empirical indicator of chaos are Lyapunov exponents [28]. Lyapunov exponents characterize the rate of separation of infinitesimally close trajectories in state-space [28]. The rate of separation can be different for different orientations of the initial separation vector, hence the number of Lyapunov exponents is equal to the number of dimensions in phase space. If we have a positive Lyapunov exponent, that means we have an expanding direction. However, if the *sum* of the Lyapunov exponents is negative, that means we have contracting volumes in phase space. These two seemingly contradicting properties of the Lyapunov exponents are indications of chaotic behaviour in the dynamical system. Lyapunov exponents for our three systems are computed in the next section.

3.6 Five Dimensional Memristor Based Chaotic Circuit

Fig. 3.11 shows that if we add inductor L_1 to the canonical memristor based chaotic circuit and set its value to 180 mH we still obtain chaos (all the other parameters,

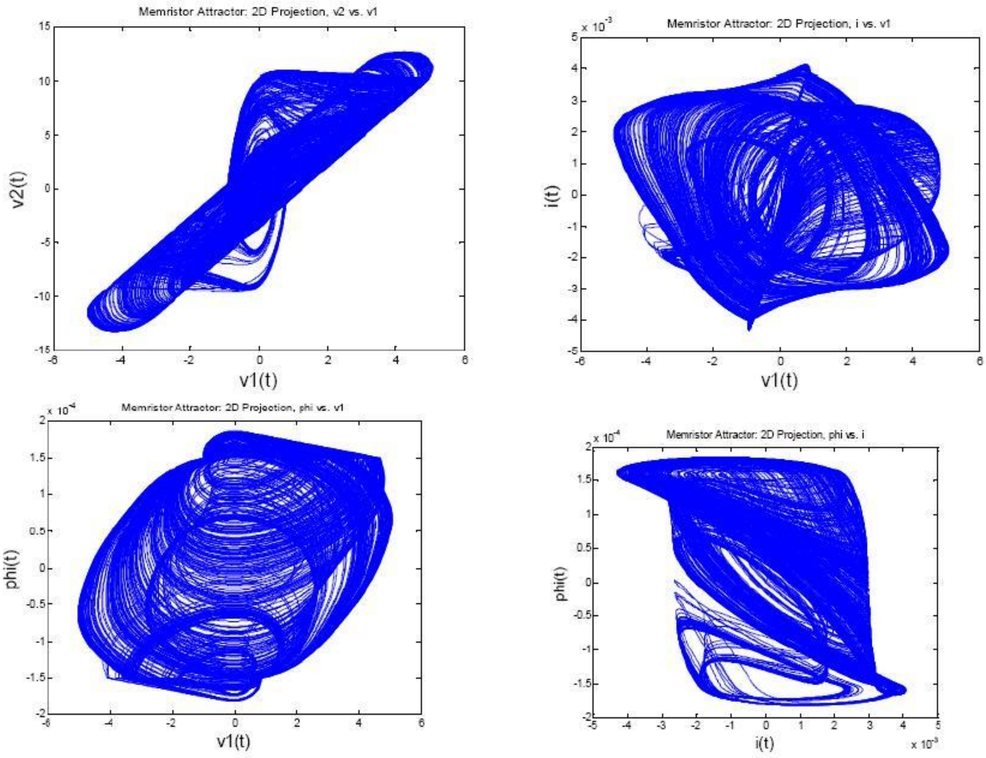


Figure 3.8: 2D Projections of the attractor from the four-element memristor-based chaotic circuit.

nonlinearity and initial conditions remain the same). The equations describing the five dimensional circuit are:

$$
\begin{aligned}
\frac{d\phi}{dt} &= v_1(t) \\
\frac{dv_1(t)}{dt} &= \frac{1}{C_1}\left(i_{L1} - W(\phi)\cdot v_1\right) \\
\frac{dv_2(t)}{dt} &= \frac{1}{C_2}\left(-i_{L1} - i_{L2}\right) \\
\frac{di_{L1}(t)}{dt} &= \frac{1}{L_1}\left(v_2 - v_1 - i_{L1}\cdot R\right) \\
\frac{di_{L2}(t)}{dt} &= \frac{v_2}{L_2}
\end{aligned}
\tag{3.8}
$$

Fig. 3.12 shows the attractors obtained from the rescaled canonical memristor-based chaotic circuit. Note that MATLAB simulation code for the circuit above has not been given since the MATLAB code can be easily extrapolated from the other appendices.

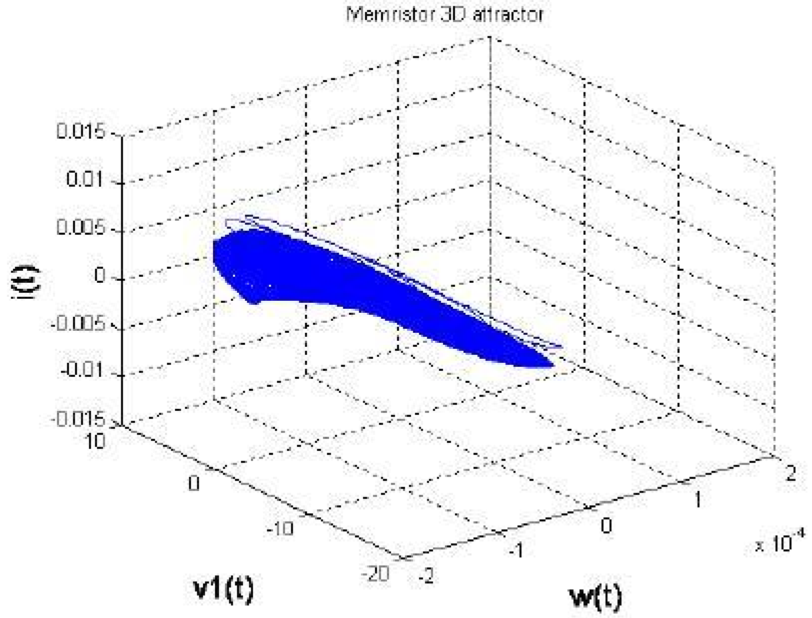


Figure 3.9: 3D attractor from the four-element memristor-based chaotic circuit with only one negative element.

The point to note here is the ease of extending the four-dimensional system to a higher dimension. We simply added another element to the circuit and just had to tune its value. Moreover, as pointed out in literature [25], higher dimensional systems are suitable for secure communication because their attractors do not have an easily identifiable structure unlike lower dimensional chaotic systems. This directs the attention of secure communication to higher-dimensional systems [24], [4]. Hence, an interesting question that warrants further study is whether even higher dimensional (6th order, 7th order,...) circuits can be obtained from this five dimensional circuit.

3.7 Lyapunov Exponent Calculations

Before computing the Lyapunov exponents, the time scales for the circuits are scaled to the order of seconds by: $\tau = \frac{t}{\sqrt{|L_1 C_2|}}$. This is necessary because the Lyapunov exponent algorithms numerically converge for these time scales. At lower time scales, we need really small step sizes for the ode solvers. This causes huge round-off errors. As mentioned earlier, we use two independent methods to estimate the Lyapunov exponents: the QR method from [13] and the time-series method from [44].

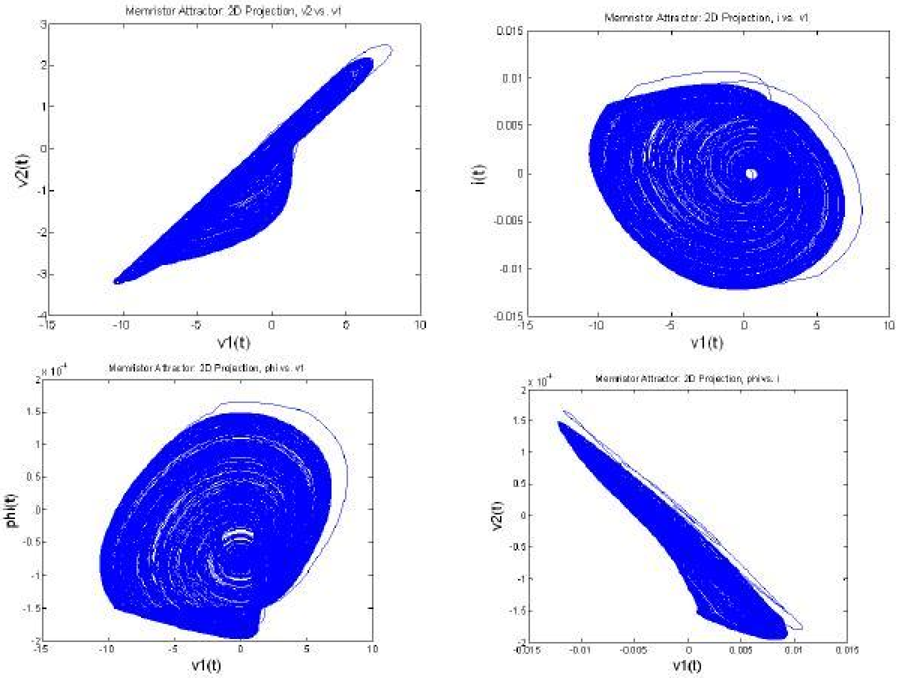


Figure 3.10: 2D Projections of the attractor from the four-element memristor-based chaotic circuit with only one negative element.

The LET toolbox from [38] and the Lyapunov time series toolbox from [18] have been used to estimate the exponents. Table 3.3 summarizes our results. The appendix has the MATLAB code used for obtaining the values in Table 3.3. Let us analyze each row in the Table 3.3.

Table 3.3: Summary of Lyapunov exponent computations via QR method and Time Series method

Type of Memristor Based Chaotic Circuit	QR	Time Series
canonical	0.085,0,-3e-3,-0.668	0.086,0,5e-4,-0.672
four-element	0.11,0,0,-1.16	0.1,0,0,-1.2
four-element with only negative C_2	0.1,0,0,-0.7	0.1,0,0,-0.7
five dimensional	0,0.22,0,-2.99,-3.05	0,0.22,0,-3.38,-1.85

1. For the canonical memristor based chaotic circuit , we notice that we have four Lyapunov exponents. This alludes to the possibility of hyperchaos. However, in the circuit presented in this paper, hyperchaos seems to be absent. Although the time series method indicates a second positive Lyapunov exponent of 0.0005,

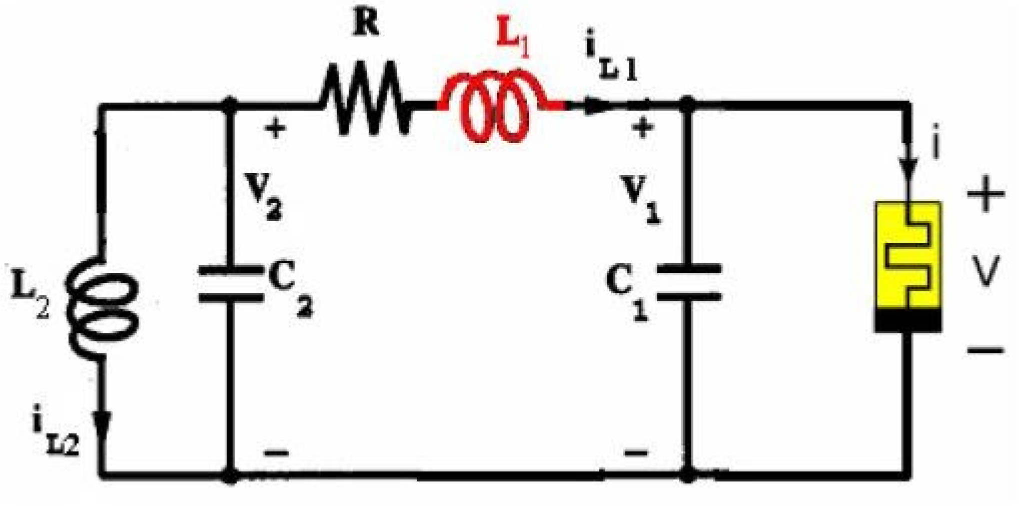


Figure 3.11: Note that the addition of the inductor L_1 results in a five dimensional circuit. We can obtain chaos for an inductor value of 180 mH.

this is probably numerical error and this Lyapunov exponent may tend towards zero as $t \to \infty$.

2. The four-element Chua's circuit with a memristor has one positive Lyapunov exponent, indicating the presence of chaos [18].

3. The four-element circuit with only a negative capacitance also gives to rise to chaotic behaviour because of the positive Lyapunov exponent [18]. However, Barboza and Chua's four-element circuit without the memristor does not seem to have chaotic behavior for the same set of C_1, L_1, C_2 and κ values. No matter what N_R function we choose for the Barboza circuit, we do not seem to get hyperbolic saddle equilibria if we have only one negative element. This difference between the Barboza-Chua circuit and the memristor circuit warrants further study.

4. The five dimensional memristor based chaotic circuit also has one positive Lyapunov exponent.

Also note that the sum of the Lyapunov exponents is negative for all the circuits. This implies that volumes contract in phase space, however the positive Lyapunov

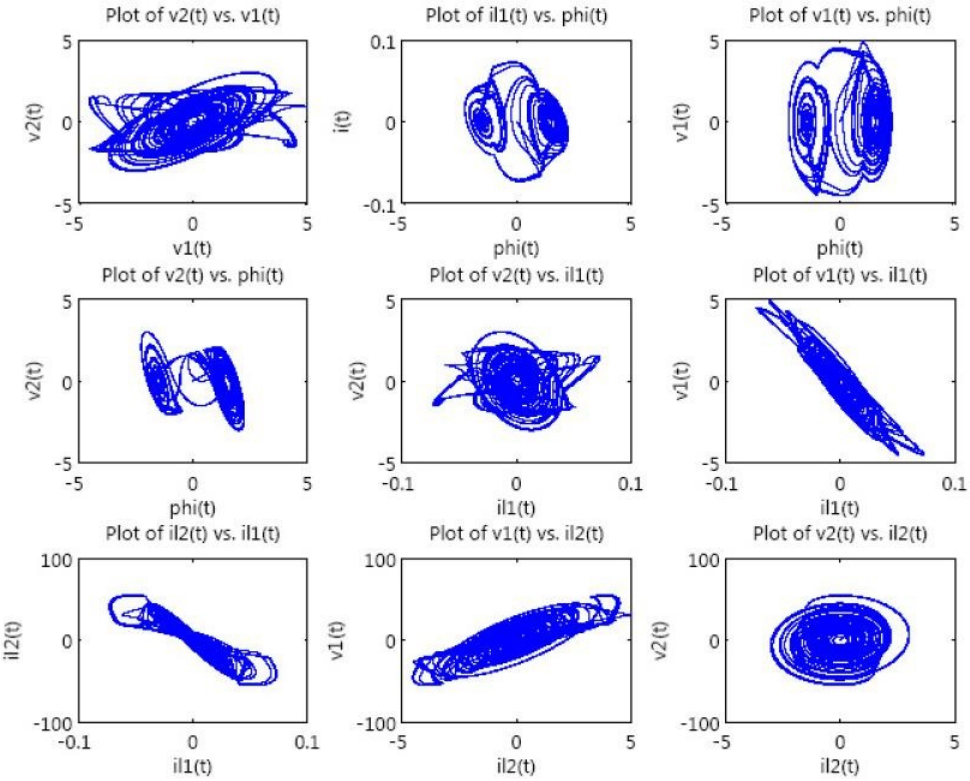


Figure 3.12: Attractors obtained from the five dimensional circuit. Note that state scaling has already been incorporated.

exponent indicates an expanding trajectory. Therefore the trajectories eventually converge to a fractal structure, namely, the chaotic attractor.

So far we have presented four memristor based chaotic circuits and provided numerical evidence of chaos in these circuits via Lyapunov exponents. The next section of this chapter provides a rigorous verification of chaos for these systems. We provide the general theory and prove chaos rigorously in the canonical memristor baed chaotic circuit. Proofs for the other circuits follow in a similar fashion and will not be pursued in this chapter.

3.8 Topological Horseshoe Theory and Rigorous Verification of Chaos

3.8.1 Intuitive Ideas Behind the Proof

Recall the Canonical Memristor Based chaotic circuit from Result . 2 and Eq. 3.3. The system equations are repeated below for convenience:

$$\begin{aligned}
\frac{d\phi}{dt} &= v_1(t) \\
\frac{dv_1(t)}{dt} &= \frac{1}{C_1}\left(\frac{v_2(t)-v_1(t)}{R} - W(\phi(t))\cdot v_1(t)\right) \\
\frac{dv_2(t)}{dt} &= \frac{1}{C_2}\left(\frac{v_1(t)-v_2(t)}{R} - i_L(t)\right) \\
\frac{di_L(t)}{dt} &= \frac{v_2(t)}{L}
\end{aligned} \tag{3.9}$$

The $Q(\phi)$ function is:

$$Q(\phi) = -0.5\cdot 10^{-3}\cdot\phi + \frac{-0.8\cdot 10^{-3}+0.5\cdot 10^{-3}}{2}\cdot(|(\phi+1)|-|(\phi-1)|) \tag{3.10}$$

The parameters of our system have already been defined as $C_1 = 5.5\ nF, C_2 = 49.5\ nF, L = 7.07\ mH, R = 1428\ \Omega$. Since the system above is four-dimensional ($\vec{x} \in \mathbb{R}^4$), Shilnikov's theorem is not applicable. However we can use higher-dimensional (topological horseshoes) [48] to rigorously prove the existence of chaos. The idea is to check if a Poincare map of the attractor leads to *crossing conditions* [49] similar in principle to the crossing conditions of the Smale horseshoe in three-dimensions [48].

But an important feature of our system is that it is not hyperchaotic because we have two Lyapunov exponents which are zero. Therefore we need only a two-dimensional Poincare map because we have only one expanding direction corresponding to the single positive Lyapunov exponent.

Hence the intuitive idea is to take a two-dimesional Poincare crossection of a three-dimensional attractor and check to see if a horseshoe is embedded in this Poincare map. Then using elementary symbolic dynamics [49] we can check if crossing conditions are satisfied. Our approach will basically mirror the approach in [49].

3.8.2 Rigorous Verification of Chaos in Canonical Memristor Based Chaotic Circuit

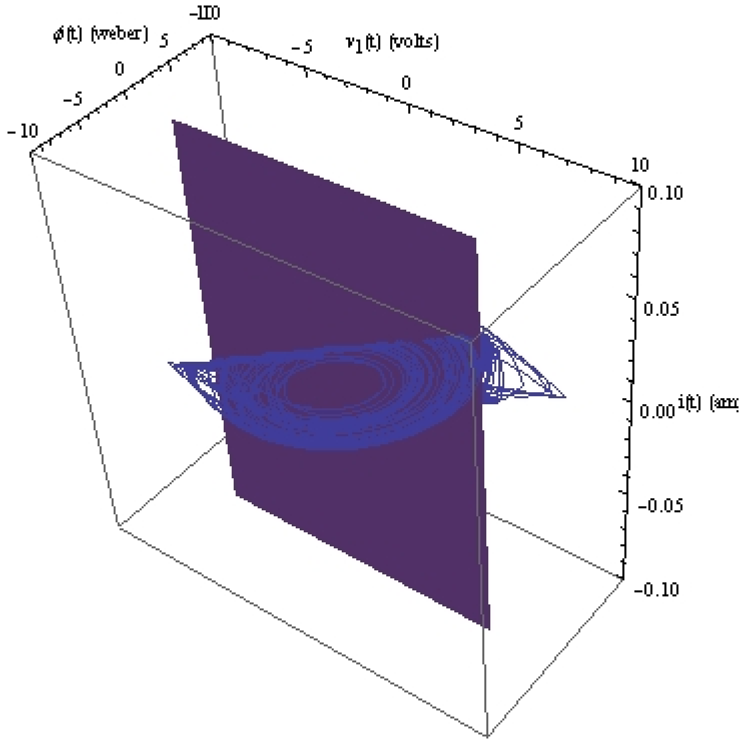


Figure 3.13: A cross section of the attractor obtained by the plane $\phi = -1$. Note that the variables have been linearly scaled to realistic values.

Consider the plane $\phi = -1$ as shown in Fig. 3.13. Note that picking a suitable cross-section takes time and knowledge of the phase space. It took the author approximately 5 hours to pick a suitable Poincare section! The code for obtaining Fig. 3.13 is in the Appendix. Next, we will pick a rectangle $|ABCD|$ on this plane, refer to Fig. 3.14. Now, consider the Poincare map:

$$P : |ABCD| \rightarrow \Pi \tag{3.11}$$

For every $x \in |ABCD|$, $P(x)$ is defined to be the second return point, i.e., the point where the orbit $\phi(x,t)$ of Eq. 3.3 intersects the plane Π a second time. Before we study the dynamics of P, we need the following result from topological horseshoe theory [48].

Theorem 2 (Topological Horseshoe Theorem)**.** *Let X be a separable metric space. Consider a continuous map $f : Q \rightarrow X$ where $Q \subset X$ is locally connected and compact. Suppose Q satisfies the following conditions:*

1. *There exist two subsets of Q, denoted by Q^1 and Q^2, which are disjoint and compact.*

2. *Each connected component of Q intersects both Q^1 and Q^2.*

3. *The cross number m of Q with respect to f is ≥ 2.*

Then, there exists a closed invariant set $Q_l \subset Q$ for which $f|Q_l$ is semiconjugate to a $m - shift$ map.

We need the following two definitions [48] for using Theorem 2. The first is the conventional definition of a semiconjugate shift map. The second is the definition of the cross number.

Definition 16. *If there exists a continuous and onto map: $h : Q_l \longrightarrow \Sigma_m$ such that $h \circ f = \sigma \circ h$, then f is said to be semiconjugate to σ.*

Definition 17. *We will define the cross number in (3) below.*

1. *A connection Γ for Q^1 and Q^2 is a compact subset of Q that intersects both Q^1 and Q^2.*

2. *A preconnection γ is a compact connected subset of Q for which $f(\gamma)$ is a connection.*

3. *The cross number m is now defined to be the largest number such that every connection contains at least m mutually disjoint preconnections.*

We now state the main theorem that is to be proved in this section.

Theorem 3. *For the Poincare map P corresponding to the cross section $|ABCD|$, there exists a closed invariant set $\Lambda \subset |ABCD|$ for which $P|\Lambda$ is semiconjugate to a 2-shift map.*

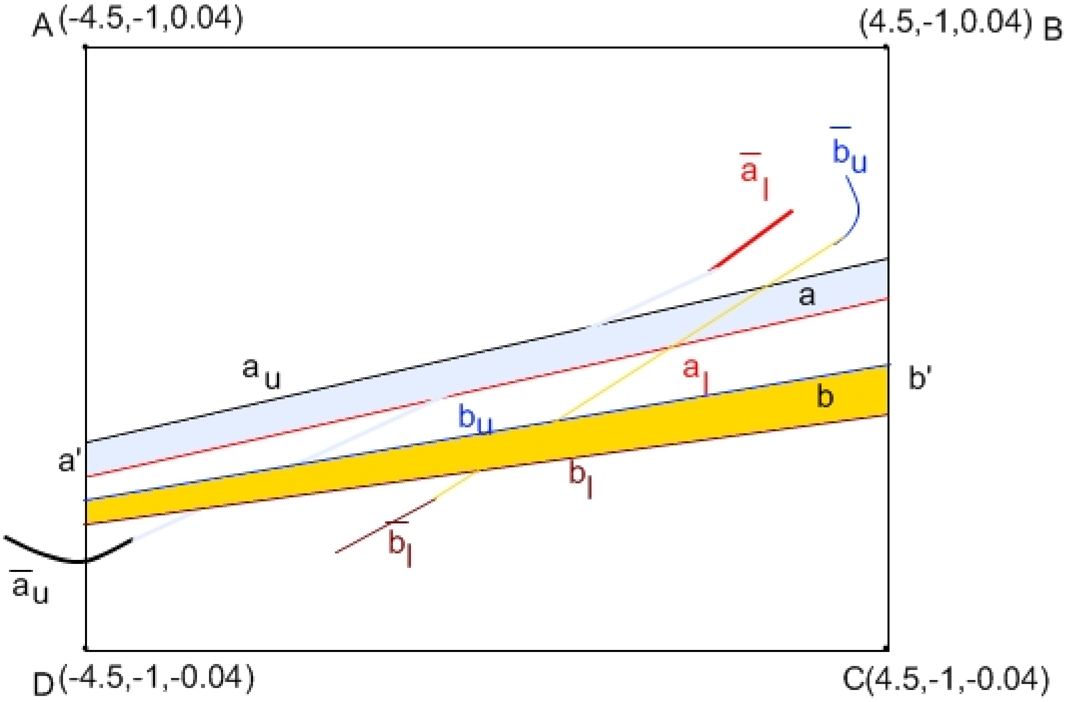


Figure 3.14: Subset Q of X for calculating the cross number. The two compact subsets Q^1 and Q^2 of Q are shown as parallelograms a' and b'

Proof. Now, in our case, the separable metric space X is obviously $\mathbf{R}^2$. Thus Q is the rectangle $|ABCD|$. Hence we need to prove that the the cross number of Q is 2. In order to do this, we fill find two subsets of $|ABCD|$ (i.e..Q^1 and Q^2 in Theorem 2) such that the cross number can be easily found. Consider the two parallelograms a' and b' in Fig. 3.14. It can be seen that the Poincare map sends the subset a (interior of a'), a_u (upper edge of a') and a_l (lower edge of a') to its image as follows.

$$a \rightarrow \bar{a}, a_u \rightarrow \bar{a_u}, a_l \rightarrow \bar{a}_l \tag{3.12}$$

Also, the Poincare map affects parallelogram b as follows:

$$b \rightarrow \bar{b}, b_u \rightarrow \bar{b_u}, b_l \rightarrow \bar{b_l} \tag{3.13}$$

Hence, $f(Q^1)$ and $f(Q^2)$ are connections. Therefore from Definition 17, we have $m = 2$. □

On a concluding remark to this section, note that although topological horseshoe theory can be used to *rigorously prove* the existence of chaos, we provided a *rigorous verification* of chaos. This is because we *numerically* determined the Poincare map. A *rigorous proof* would involve estimating the errors from the numerical approximation, this could be the subject of future work.

3.9 Implementing Memristor Based Chaotic Circuits

In this section, we will propose a circuit design for the canonical memristor based chaotic circuit. Since memristors (as of the writing of this thesis) are commercially unavailable, we will implement a memristor using analog circuitry. In the interests of simplicity, we replaced the piecwise linear characteristic with a smooth nonlinearity (similar to Zhong's [50] implementation of Chua's circuit with a cubic nonlinearity).

3.9.1 Practical Implementation of a Memristor

Consider the $Q(\phi)$ given below.

$$Q(\phi) = \alpha\phi + \beta\phi^3 \tag{3.14}$$

The expression for $i_m(t)$ can be derived from Eq. 3.14 and basic definitions Eqs. 3.1 and 3.2:

$$\begin{aligned} i_m(t) &= \frac{dq}{d\phi}\frac{d\phi}{dt} \\ &= \frac{d}{d\phi}\left(\alpha\phi + \beta\phi^3\right) \cdot v(t) \\ &= \left(\alpha + \beta \cdot 3\phi^2\right) \cdot v(t) \\ &= W(\phi(t)) \cdot v(t) \end{aligned}$$

Since $\phi(t) \triangleq \int v(t)dt$, we have the following expression for the current through our memristor with a cubic nonlinearity:

$$i_m(t) = \left(\alpha + \beta \cdot 3\left(\int v(t)dt\right)^2\right) \cdot v(t) \tag{3.15}$$

The basic circuit to realize the current expression above is the multiplier circuit in a feedback loop from [50]. A block diagram view of the circuit along with an integrator is shown in Fig. 3.15. The circuit in Fig. 3.15 is the analog implementation of our memristor. In Fig. 3.15, $U1$ is the integrator for implementing $\phi(t) = \int v(t)dt$. $U2$ is the multiplier that implements $\frac{\phi(t)^2}{10}$. Multiplier $U3$ implements $\frac{\phi(t)^2}{10} \cdot \frac{v(t)(R_4+R_5)}{10 \cdot R_4}$. Refer to the AD633 four-quadrant analog multiplier datasheet for further information. $U4$, op-amp AD711kN, is the current inverter and implements the function (if $R_1 = R_2$):

$$i(t) = \frac{-v}{R_3} + \left(\frac{\left(\int v(t)dt\right)^2}{10}\frac{v \cdot (R_4 + R_5)}{10 \cdot R_4}\right)\frac{1}{R_3} \tag{3.16}$$

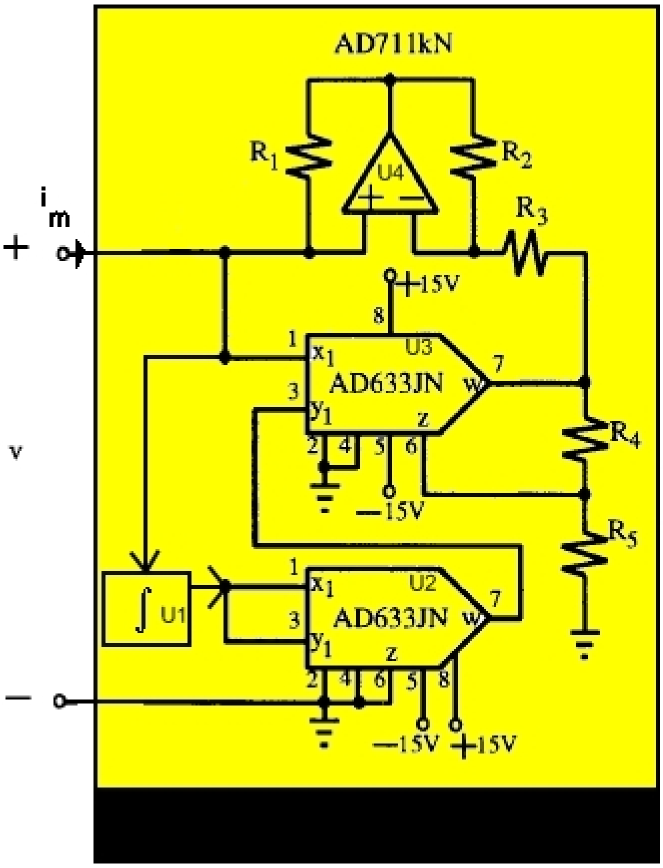


Figure 3.15: Practical Circuit for Realizing a memristor.

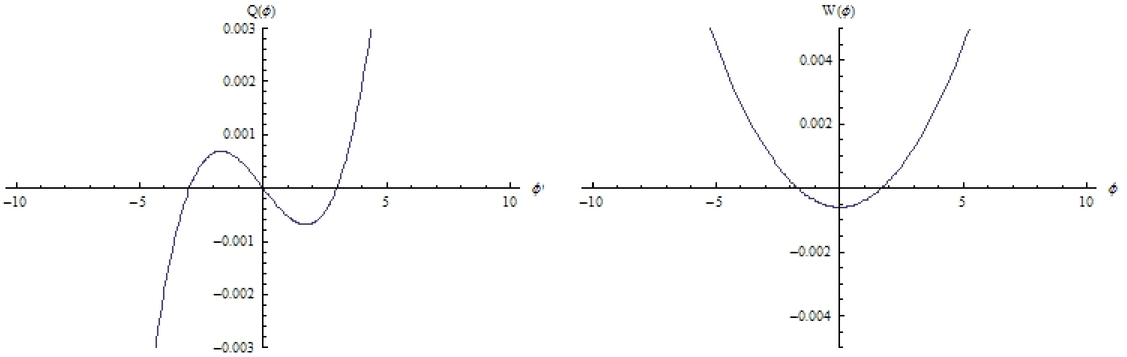


Figure 3.16: Plot of Charge vs. Flux along with a plot of the menductace function for our memristor

In Eq. 3.15, we will choose $\alpha = -0.599 \cdot 10^{-3}$ and $\beta = 0.0677 \cdot 10^{-3}$. These parameters were obtained from [50]. For these parameters, a Mathematica plot of the charge versus flux and the menductance function is shown in Fig. 3.16. Using the values of α and β in Eq. 3.16, we get the following values for the components in Fig. 3.15: $R_1 = R_2 = 2k,\ R_3 = 1.69k,\ R_4 = 3k,\ R_5 = 100k$. Now, the "memory" for the system is in the integrator in Eq. 3.16. Therefore, in order to realize a practical integrator circuit we rescaled the state variables in Eq. 3.3 to get the equations below.

$$\begin{aligned}
\frac{d\phi}{dt} &= -1000 \cdot v_1(t) \\
\frac{dv_1(t)}{dt} &= \frac{1}{C_1}\left(\frac{v_2(t) - v_1(t)}{R} - (\alpha + \beta 3 \cdot \phi^2) \cdot v_1(t)\right) \\
\frac{dv_2(t)}{dt} &= \frac{1}{C_2}\left(\frac{v_1(t) - v_2(t)}{R} - i_L(t)\right) \\
\frac{di_L(t)}{dt} &= \frac{v_2(t)}{L}
\end{aligned} \tag{3.17}$$

Now suppose $\phi_m \overset{\triangle}{=} \frac{-\phi}{1000}$. Then we can rewrite Eq. 3.17 above as:

$$\begin{aligned}
\frac{d\phi_m}{dt} &= v_1(t) \\
\frac{dv_1(t)}{dt} &= \frac{1}{C_1}\left(\frac{v_2(t) - v_1(t)}{R} - (\alpha + \beta 3 \cdot (1000)^2 \cdot \phi_m^2) \cdot v_1(t)\right) \\
\frac{dv_2(t)}{dt} &= \frac{1}{C_2}\left(\frac{v_1(t) - v_2(t)}{R} - i_L(t)\right) \\
\frac{di_L(t)}{dt} &= \frac{v_2(t)}{L}
\end{aligned} \tag{3.18}$$

Thus the menductance function that we implement in reality is:

$$W(\phi_m) = -\alpha + 3\beta \cdot (1000)^2 \cdot (\phi_m)^2 \tag{3.19}$$

The menductance parameters have already been defined. The circuit parameters in Eq. 3.18 are $C_2 = 68\ nF,\ C_1 = 6.8\ nF,\ L = 18\ mH$ (parameters from [50]). We now have all the parameters necessary to synthesize the circuit. The complete circuit schematic is shown in Fig. 3.17. We use a voltage buffer $U1\_1$ to avoid loading effects.

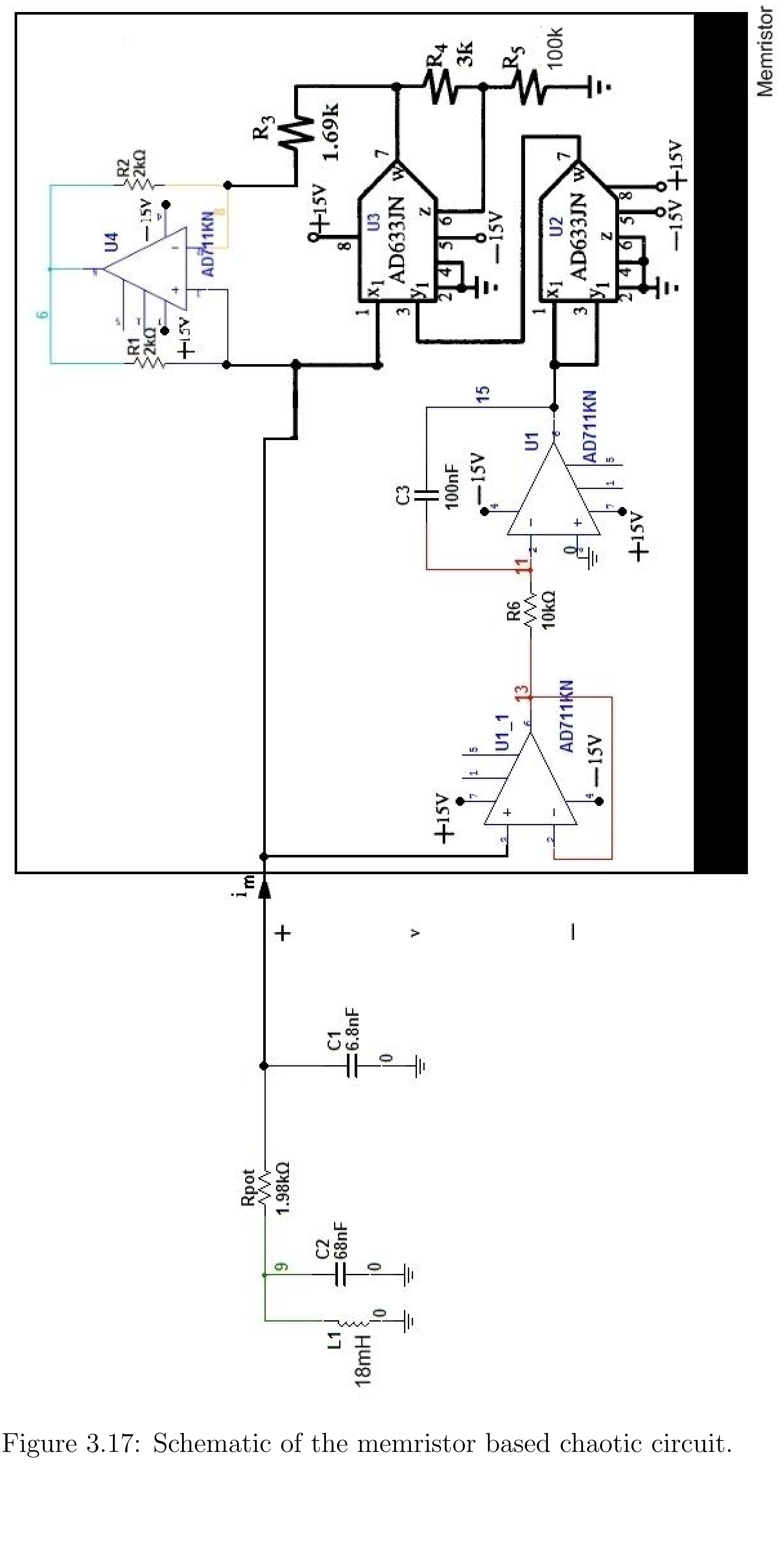


Figure 3.17: Schematic of the memristor based chaotic circuit.

3.9.2 Limit Cycles, Strange Attractors and Power Spectra from the Memristor Chaotic Circuit

Fig. 3.18 shows a limit cycle and a chaotic attractor from the memristor based chaotic circuit. The images were taken using a Agilent Intuilink and National Instruments software. The oscilloscope used is an Agilent 54600D.

Notice the difference in the frequency spectra between the limit cycle and the attractor. The fourier transform suggests that the limit cycle is periodic. The frequency spectrum of the attractor is wideband, suggesting chaotic behavior.

In order to illustrate bifurcation phenomenon and test the robustness of the circuit, we used a different set of capacitor values: $C_2 = 47\ nF,\ C_1 = 4.7\ nF$. The schematic for this circuit is shown in Fig. 3.19. The bifurcation of a limit cycle into a strange attractor is shown in Fig. 3.20 and Fig. 3.21 for different values of Rpot. Notice that the crossing trajectories in two dimensional phase space indicate the presence of a limit cycle in three dimensions.

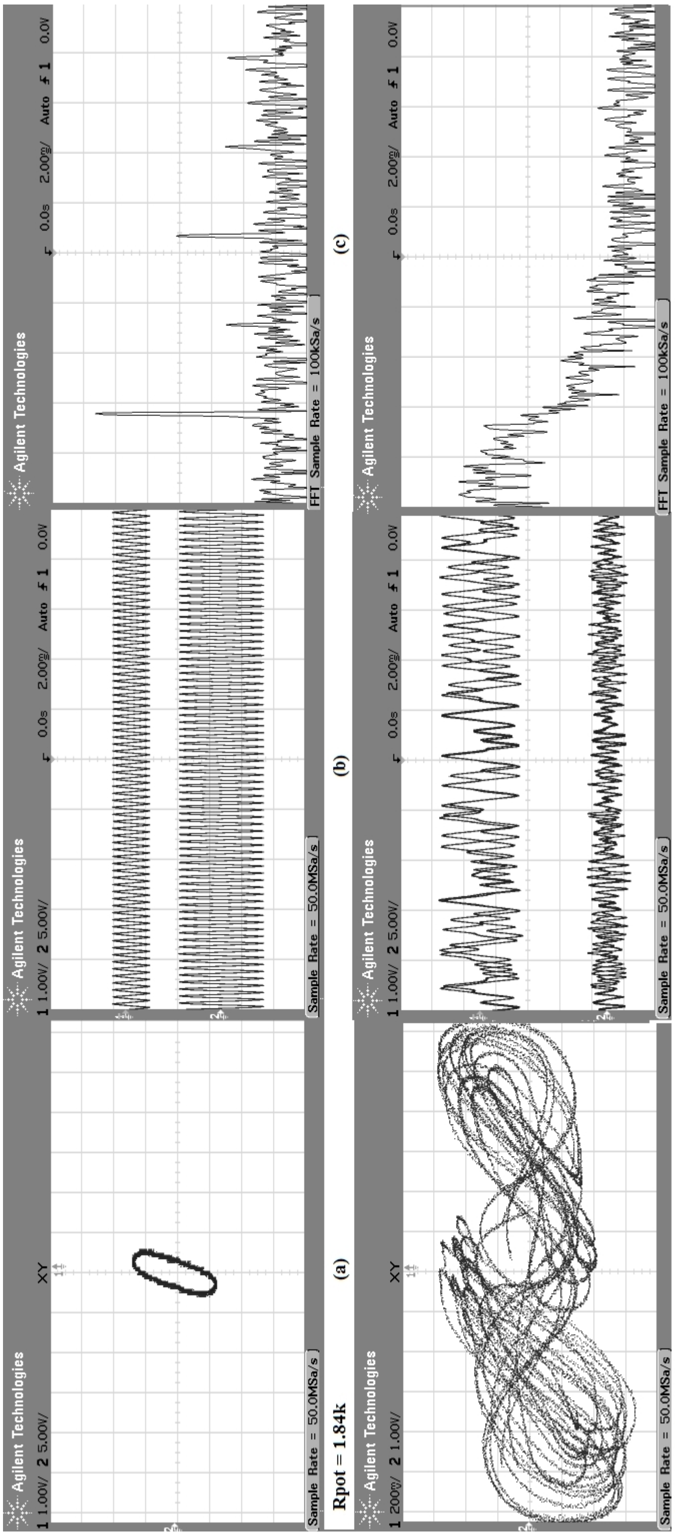


Figure 3.18: Phase plot, time domain waveforms and power spectra for a limit cycle and a strange attractor from the memristor based chaotic circuit. Channel 1 (X) is across $\phi(t)$ (Node 15 in Fig. 3.17) and Channel 2 (Y) is across $v_2(t)$ (Node 9 in Fig. 3.17).

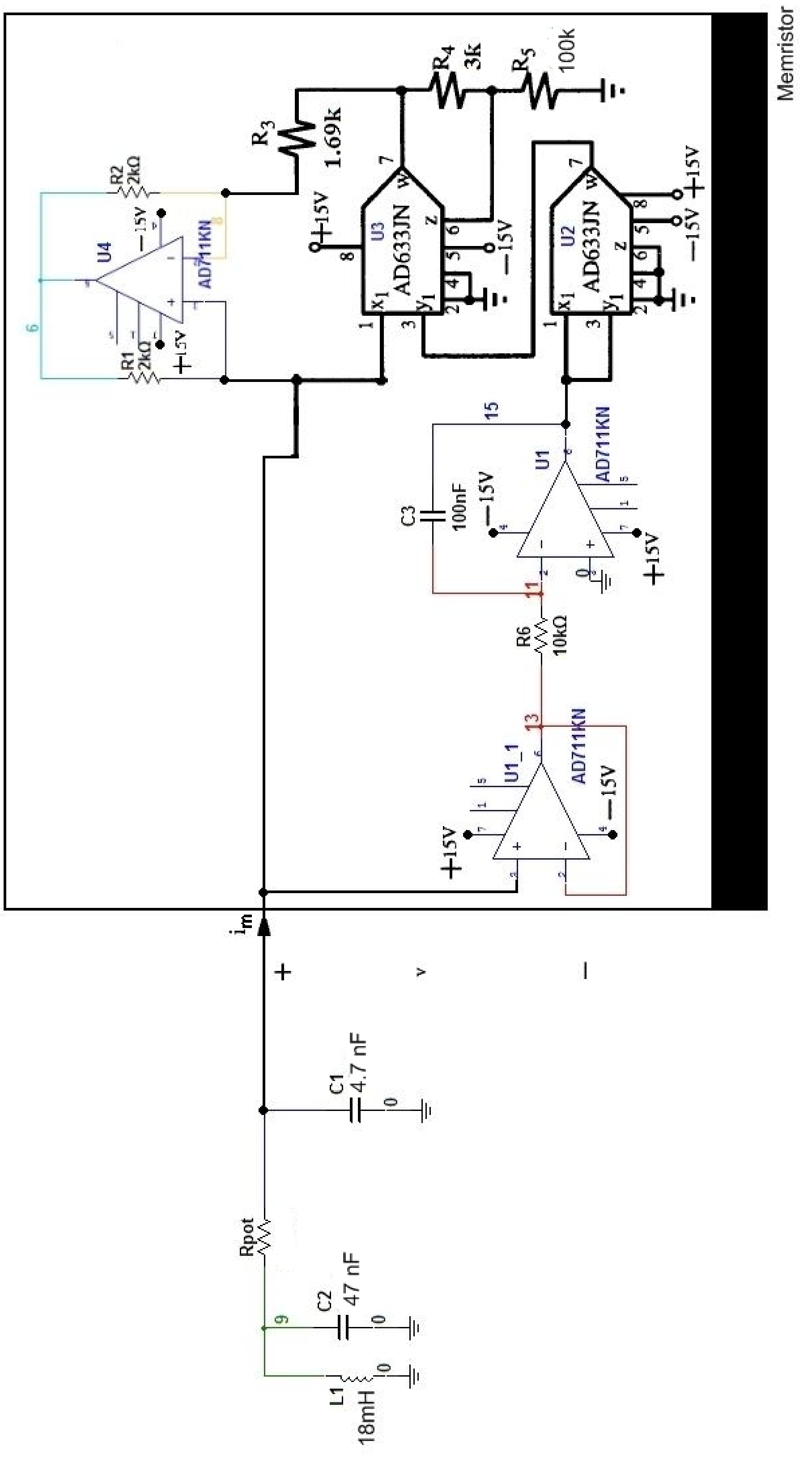


Figure 3.19: Schematic of the memristor based chaotic circuit for illustrating bifurcation phenomenon

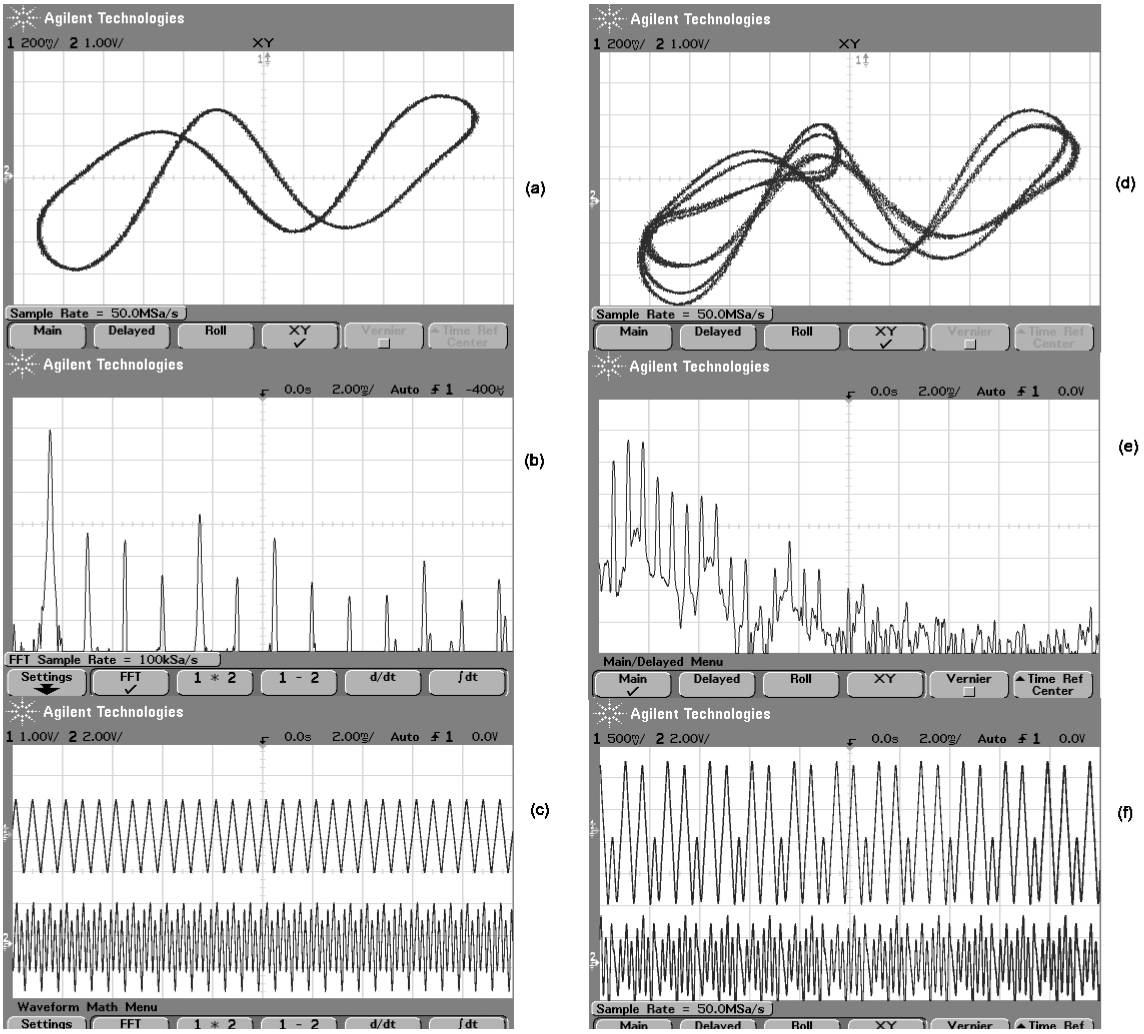


Figure 3.20: Phase portrait, power spectrum and time domain waveform for (a) through (c), period-1 limit cycle $Rpot = 2.24k$;(d) through (f) period-2 limit cycle, $Rpot = 2.20k$. Here, $C_2 = 47\ nF$ and $C_1 = 4.7\ nF$. The menductance parameters remain the same. Channel 1 (X) is across $\phi(t)$ (Node 15 in Fig. 3.19) and Channel 2 (Y) is across $v_2(t)$ (Node 9 in Fig. 3.19).

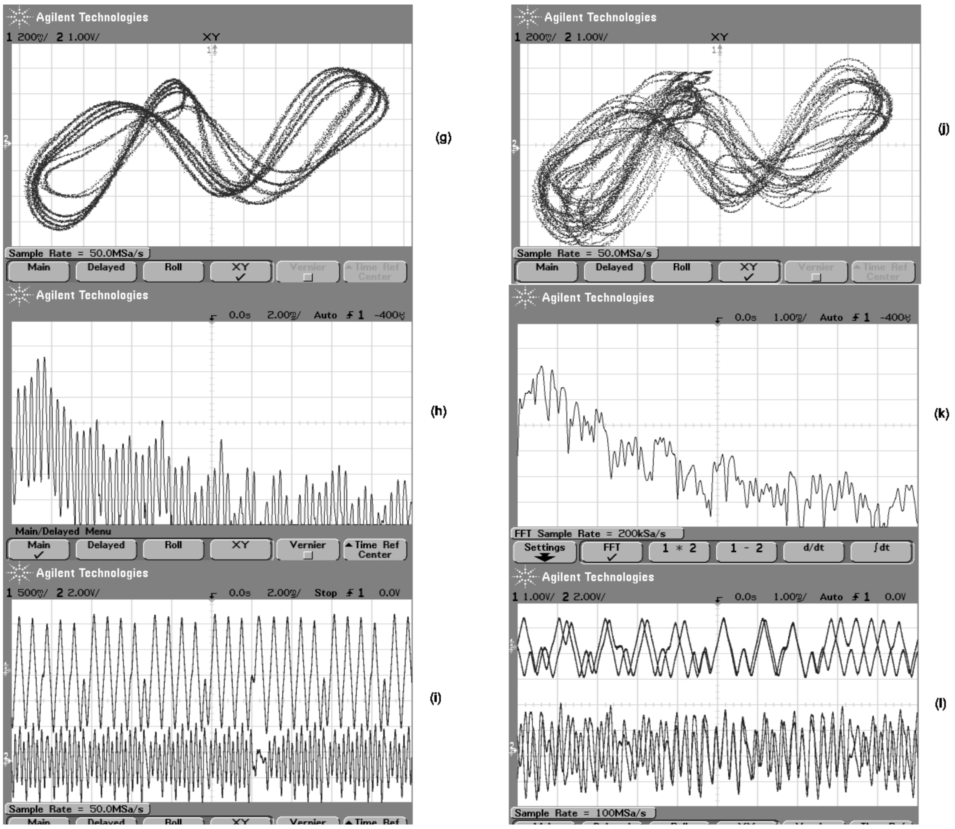


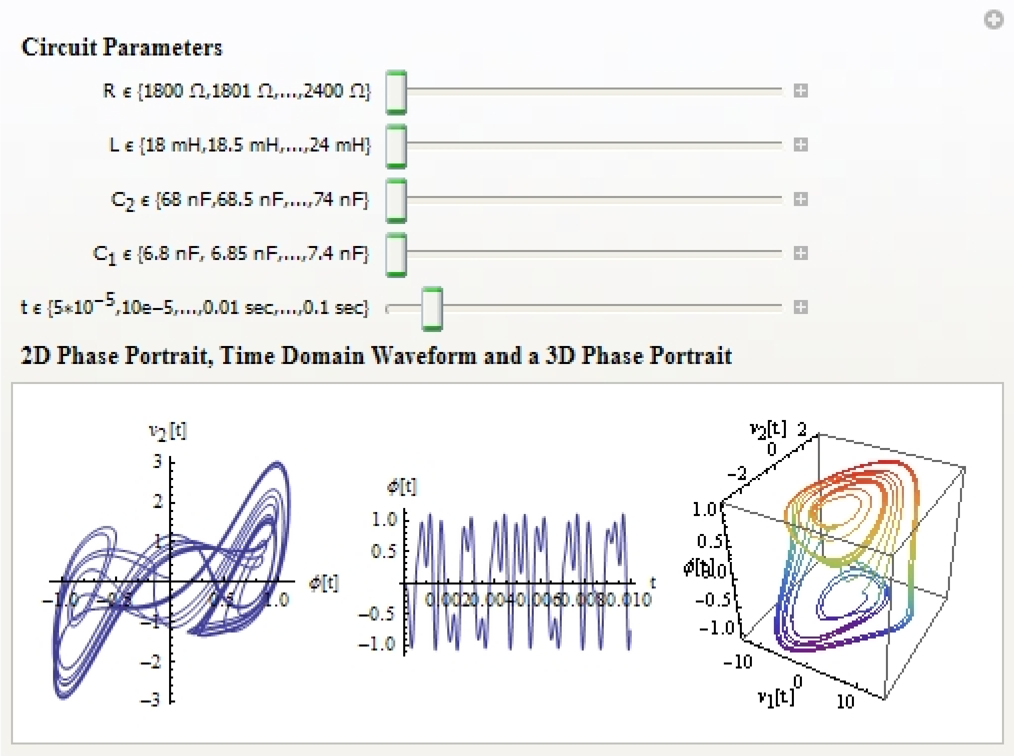
Figure 3.21: Phase portrait, power spectrum and time domain waveform for (g) through (i), period-4 limit cycle $Rpot = 2.18k$; (j) through (l) attractor, $Rpot = 2.16k$. Here, $C_2 = 47\ nF$ and $C_1 = 4.7\ nF$. The menductance parameters remain the same. Channel 1 (X) is across $\phi(t)$ (Node 15 in Fig. 3.19) and Channel 2 (Y) is across $v_2(t)$ (Node 9 in Fig. 3.19).

3.10 Lyapunov Exponents for The Practical Memristor Based Chaotic Circuit

Although we did compute Lyapunov exponents for the Canonical Memristor Based Chaotic circuit, we recompute them for the circuit in Eq. 3.18. We need to time scale the equations for numerical stability: $\tau \overset{\triangle}{=} \frac{t}{\sqrt{L \cdot C_2}}$. The code for computing the Lyapunov exponents is very similar to the Canonical Memristor based chaotic circuit case and has been omitted from the Appendix. The resulting exponents from LET [38] are 0.0803, 0, 0, -1.08 and Time Series analysis [18] are 0.077, 0, 0, -1.078 respectively. Notice the presence of a positive Lyapunov exponent indicates the presence of chaos. Also the sum of the Lyapunov exponents is negative indicating that volumes contract in phase space.

3.11 A Note On Implementation Issues

If we simulate the system in Eq. 3.18 using a Mathematica demonstration [30], we get the results shown in Fig. 3.22. The demonstration code is given in the Appendix.



Circuit Parameters

R ∊ {1800 Ω,1801 Ω,...,2400 Ω}

L ∊ {18 mH,18.5 mH,...,24 mH}

C_2 ∊ {68 nF,68.5 nF,...,74 nF}

C_1 ∊ {6.8 nF, 6.85 nF,...,7.4 nF}

t ∊ {$5*10^{-5}$,10e−5,...,0.01 sec,...,0.1 sec}

2D Phase Portrait, Time Domain Waveform and a 3D Phase Portrait

Figure 3.22: A screenshot from a Mathematica demonstration showing the $v_2(t)$ vs $\phi(t)$ attractor, among other plots.

Notice the difference in attractor shapes between Fig. 3.18 and Fig. 3.22. This can be attributed to the non-idealities in the op-amp implementing the integrator in Fig. 3.19. Since the op-amp input terminal currents are never exactly zero they introduce offset voltages across the capacitor. This implies that we do not get the exact integral term from Eq. 3.18 in the physical circuit. This approximation translates to a distortion of the attractor. Techniques for removing this implementation issue are the subject of future work.

3.12 Conclusion

In this chapter, we discussed memristor based (higher dimensional) chaotic circuits. We studied a variety of circuits and concentrated on one circuit in particular.

The most important contribution of this chapter is a practical realization of a memristor based chaotic circuit.

In the next section of this thesis, we will study a different area of nonlinear dynamics, Cellular Automata.

Part II

Cellular Automata

Chapter 4

Integer Factorization and Cellular Automata

4.1 Introduction and Chapter Organization

In this chapter, the relationship between integer factorization and Cellular Automata are investigated. This provides an alternate classification scheme for Cellular Automata evolution as opposed to Wolfram's [47] and Chua's [12]. The advantage of this scheme is that it relates Cellular Automata to a fundamental property of numbers, factorization.

In the next section, important definitions for understanding Cellular Automata and the main result from this chapter are provided. This is followed by a rigorous proof of the integer factorization property in Rule 46. The classification scheme is then derived from the behavior of Rule 46, followed by conclusions.

4.2 Important Definitions and Main Result

Fig. 4.1 shows that a Cellular Automata consists of L cells ($L = n + 1$ in Fig. 4.1) with periodic boundary conditions [12]. The state of each cell is either a 0 or 1. Each cell i interacts with only its nearest neighbors $i - 1$ and $i + 1$. Table 4.1 shows how the state, x_i^n, of each cell at iteration n is updated based on the state of its nearest neighbors x_{i-1}^n and x_{i+1}^n. Here, $\beta_k \in \{0, 1\}\ \forall k \in \{0, 1, ..., 7\}$. The cellular automaton rule that we are interested in is Rule 46. Table 4.2 shows the update function for Rule 46. Now, consider the evolution of a Cellular Automaton with $L = 6$ and initial condition 001001, shown in Fig. 4.2. Notice how the last row in Fig. 4.2 is equal to the second row. This shows that the we have a repeating sequence, this sequence is called as an *attractor* [12].

Fig. 4.2 is more informative if we view the evolution in the decimal number system

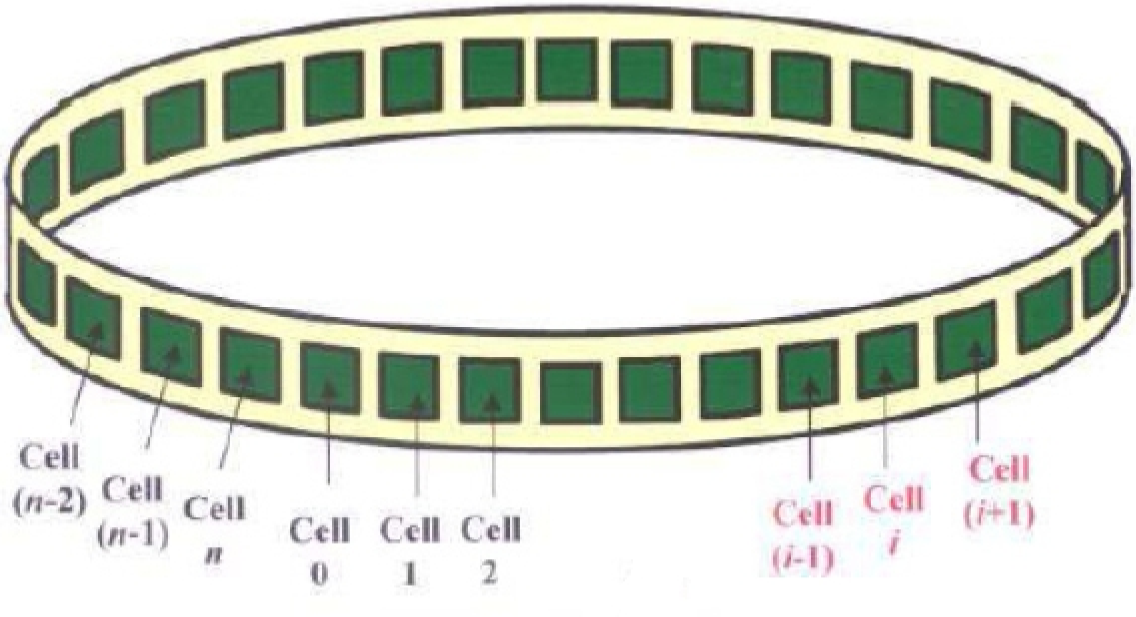


Figure 4.1: Structure of a Cellular Automata

(least significant bit is Cell 0, in accordance with convention):

$$\{\{9\}, \{27, 54, 45, 27\}\}$$

In our notation above, $\{27, 54, 45, 27\}$ is called as the *attractor* and $\{9\}$ is the *basin of attraction.* The **period** of our attractor above is 3. Consider Table 4.3. We list the bit length, the corresponding attractors and the periods for each attractor.

Table 4.1: Cellular Automaton Update Function

x_{i-1}^{n}	x_{i}^{n}	x_{i+1}^{n}	x_{i}^{n+1}
0	0	0	β_0
0	0	1	β_1
0	1	0	β_2
0	1	1	β_3
1	0	0	β_4
1	0	1	β_5
1	1	0	β_6
1	1	1	β_7

Mathematica 6 was used to obtain all the results, the code is given in the Appendix. Based on Table 4.3, we have the following result.

Result 3. ***Given a Cellular Automaton of length*** L ***that is evolving under Rule 46, if*** L ***is even, then the attractor with period*** $\frac{L}{2}$ ***(a factor of*** L***) starts at the decimal number*** $(2^{\frac{L}{2}} + 1) \cdot 3$***.***

4.3 Rigorous Proof of Factorization Property of Rule 46

Proof. In order to prove Result 3, consider a Cellular Automaton of length L that is evolving under rule 46. Suppose the smallest integer in the attractor is $(2^{\frac{L}{2}} + 1)3$. Since Rule 46 is a $\sigma = 1$, $\tau = 1$ Bernoulli Cellular Automaton [11], the bits shift left by 1 at every iteration in the evolution of the attractor. Therefore, after $\frac{L}{2}$ iterations we have:

$$\begin{aligned} 2^{\frac{L}{2}} \cdot ((2^{\frac{L}{2}} + 1)3) &= 2^{\frac{L}{2}} \cdot (2^{\frac{L}{2}}(2+1) + (2+1)) \\ &= 2^{\frac{L}{2}} \cdot (2^{\frac{L}{2}+1} + 2^{\frac{L}{2}} + 2 + 1) \\ &= 2^{L+1} + 2^{L} + 2^{\frac{L}{2}+1} + 2^{\frac{L}{2}} \end{aligned}$$

But since the attractor evolution is *modulo* L, we have:

$$\begin{aligned} 2^{L+1} + 2^{L} + 2^{\frac{L}{2}+1} + 2^{\frac{L}{2}} &= 2 + 1 + 2^{\frac{L}{2}+1} + 2^{\frac{L}{2}} \\ &= 2 + 1 + 2^{\frac{L}{2}}(2+1) \\ &= (2^{\frac{L}{2}} + 1)3 \end{aligned}$$

□

4.4 A New Classification Scheme for Cellular Automata Evolution

From [11], Rule 46 is not the only Bernoulli Cellular Automaton. Therefore, we have the following result.

Result 4. ***Result 3 also applies to the following Bernoulli*** $\sigma = 1$,$\tau = 1$ ***rules: 2, 10, 16, 24, 34, 42, 56, 130, 138, 162, 170 and 184.***

Results 3 and 4 imply that we can *rigorously predict* the elements of an attractor for *even* L in the case of Bernoulli $\sigma = 1$, $\tau = 1$ rules.

4.5 Conclusion

In this chapter, we rigorously predicted the evolution of attractors in Bernoulli totalistic one-dimensional Cellular Automata for *even* L. Nevertheless, referring to Table 4.3, we have the two following conjectures for Rule 46 (and possibly other Bernoulli $\sigma = 1$, $\tau = 1$) evolution:

Conjecture 1. Given a Cellular Automaton of length L that is evolving under Rule 46, the attractor periods are:

1. 1 and L if L is prime

2. 1 and the factors of L (excluding 2) if L is composite

Conjecture 2. Given a Cellular Automaton of length L that is evolving under Rule 46, then **any** attractor **always** starts at x_{0A} such that $x_{0A} \equiv 0 \ (mod\ 3)$. In other words, x_{0A} is divisible by 3.

A natural topic for future work would be to try and prove the conjectures above.

Table 4.2: Rule 46 Update Function

x_{i-1}^n	x_i^n	x_{i+1}^n	x_i^{n+1}
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

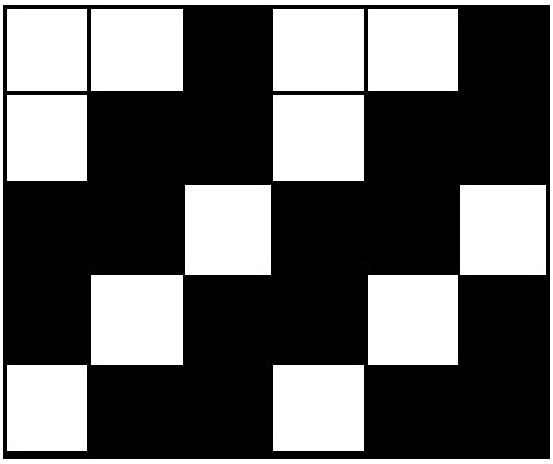


Figure 4.2: Evolution of Rule 46 ($L = 6$) for a specific initial condition

Table 4.3: Rule 46 Attractor List and Period(s) for a few bit lengths

L	Attractor(s)	Period(s)
7	$\{\{0\},\{3,6,12,24,48,96,65\}\},$ $\{27,54,108,89,51,102,77\}$	1,7
8	$\{\{0\},\{51,102,204,153\},$ $\{3,6,12,24,48,96,192,129\},$ $\{27,54,108,216,177,99,198,141\}\}$	1,4,8
12	$\{\{0\},\{1755,3510,2925\},$ $\{819,1638,3276,2457\},$ $\{195,390,780,1560,3120,2145\},$ $\{3,6,12,24,48,96,192,384,768,1536,3072,2049\},$ $\{27,54,108,216,432,864,1728,3456,2817,1539,3078,2061\},$ $\{51,102,204,408,816,1632,3264,2433,771,1542,3084,2073\},$ $\{99,198,396,792,1584,3168,2241,387,774,1548,3096,2097\},$ $\{219,438,876,1752,3504,2913,1731,3462,2829,1563,3126,2157\},$ $\{411,822,1644,3288,2481,867,1734,3468,2841,1587,3174,2253\},$ $\{435,870,1740,3480,2865,1635,3270,2445,795,1590,3180,2265\}$	1,3,4,6,12

Part III

End Matter

Chapter 5

Conclusions and Future Work

This thesis studied autonomous chaotic circuits and cellular automata. The main results along with suggestions for future work are summarized below:

1. Result 1: The alternative Chua's system in Eq. 2.4 is chaotic in the sense of Shilnikov if $\alpha = 25.24, \beta = 69.39, m_0 = -0.16, m_1 = 0.29, \gamma^* \in J \overset{\triangle}{=} [0, 0.75]$. Notice that $\gamma^* = 0$ is the Four-Element Chua's circuit. This circuit also undergoes a period-doubling route to chaos, as γ^* decreases from 1 *to* 0.

2. Result 2: Table 3.1 is the canonical memristor based chaotic circuit that has been derived from Chua's circuit. This circuit is canonical since other chaotic circuits can be derived from this one. It also shares some similar properties to Chua's circuit, for example, we can replace the piecewise-linear menductance with a smooth nonlinearity and still obtain chaos (refer to the implementation section of this chapter for details). A rigorous verification of chaos in the canonical memristor based chaotic circuit is provided via topological horseshoe theory.

   Future work could include investigating further implementations of memristor based chaotic circuits. For instance, once memristors are commercially available, once could design chaotic circuits which are free of the distortion introduced by nonideal op-amps emulating memristors. Future work could also include rigorously estimating the errors in the computations of the Poincare maps (thereby providing a rigorous proof of chaos) and also rigorously analyzing other (period-adding, intermittency) possible routes to chaos in these circuits.

3. Results 3 and 4: Given a Cellular Automaton of length L that is evolving under the rules 2, 10, 16, 24, 34, 42, 46, 56, 130, 138, 162, 170 and 184, if L is **even**, then the attractor with period $\frac{L}{2}$ (a factor of L) starts at the decimal number $(2^{\frac{L}{2}} + 1) \cdot 3$.

   Future work would be proving conjectures 1 and 2.

Bibliography

[1] Alligood, K. T., Sauer, T. D., and Yorke, J. A. ***Chaos: An Introduction to Dynamical Systems***. Springer-Verlag, New York, 1996.

[2] Barboza, R., and Chua, L. O. **The Four Element Chua's Circuit**. *International Journal of Bifurcation and Chaos 18*, 4 (2008), 1–13.

[3] Bonani, F., and Gilli, M. **Analysis of Stability and Bifurcations of Limit Cycles in Chua's Circuit Through the Harmonic-Balance Approach**. *IEEE Transactions on Circuits and Systems I - Fundamental Theory and Applications 46*, 8 (August 1999), 881–890.

[4] Carroll, T. L., Heagy, J. F., and Pecora, L. M. Transforming Signals With Chaotic Synchronization. *Physical Review E 54* (1996), 4676–4680.

[5] Chua, L. O. **Memristor - The Missing Circuit Element**. *IEEE Transactions on Circuit Theory CAT-18*, 5 (1971), 507–519.

[6] Chua, L. O. **The Genesis Of Chua's Circuit**. *Archiv fur Elektronik und Uebertragungstechnik 46*, 4 (July 1992), 250–257.

[7] Chua, L. O., Desoer, C. A., and Kuh, E. S. ***Linear and Nonlinear Circuit Theory***. McGraw-Hill, New York, 1987.

[8] Chua, L. O., Komuro, M., and Matsumoto, T. **The Double Scroll Family**. *IEEE Transactions on Circuits and Systems 32*, 11 (November 1986), 1072–1118.

[9] Chua, L. O., Sbitnev, V. I., and Yoon, S. **A Nonlinear Dynamics Perspective if Wolfram's New Kind of Science, Part II: Universal Neuron**. *International Journal of Bifurcation and Chaos 13*, 9 (2003), 2377–2491.

[10] Chua, L. O., Sbitnev, V. I., and Yoon, S. **A Nonlinear Dynamics Perspective if Wolfram's New Kind of Science, Part III: Predicting the Unperdictable**. *International Journal of Bifurcation and Chaos 14*, 11 (2004), 3689–3820.

[11] Chua, L. O., Sbitnev, V. I., and Yoon, S. **A Nonlinear Dynamics Perspective if Wolfram's New Kind of Science, Part IV: From Bernoulli Shift to $\frac{1}{f}$ Spectrum**. *International Journal of Bifurcation and Chaos 15*, 4 (2005), 1045–1183.

[12] Dogaru, R., Yoon, S., and Chua, L. O. **A Nonlinear Dynamics Perspective if Wolfram's New Kind of Science, Part I: Threshold of Complexity**. *International Journal of Bifurcation and Chaos 12*, 12 (2002), 2655–2766.

[13] Eckmann, J. P., and Ruelle, D. Ergodic Theory of Chaos and Strange Attractors. *Review of Modern Physics 57*, 3 (July 1985), 617–656.

[14] Elwakil, A. S., and Kennedy, M. P. **Construction of Classes of Circuit-Independent Chaotic Oscillators Using Passive-Only Nonlinear Devices**. *IEEE Transactions on Circuits and Systems - I 48*, 3 (2001), 289–308.

[15] Feigenbaum, M. J. **Quantitative Universality for a Class of Nonlinear Transformations**. *Journal of Statistical Physics 19* (1978), 25–52.

[16] Gelb, A., and Velde, W. E. V. ***Multiple-Input Describing Functions and Nonlinear System Design***. McGraw Hill, NEW YORK, 1968.

[17] Genesio, R., and Tesi, A. **A Harmonic Balance Approach for Chaos Prediction: Chua's Circuit**. *International Journal of Bifurcation and Chaos 2*, 1 (1992), 61–79.

[18] Govorukhin, V. Lyapunov Exponents for ODEs. *http://www.mathworks.com/matlabcentral/fileexchange/* (June 2008).

[19] Hasler, M., Kennedy, M., and Schweizer, J. **Secure Communications Via Chua's Circuit**. *International Symposium on Nonlinear Theory and its Applications*, 4 (1993), 87–92.

[20] Heyns, L. J., and Kruger, J. J. **A Describing-Function Based Criterion for a Route to Chaos in Autonomous Nonlinear Systems**. *International Journal of Control 61*, 1 (1995), 211–228.

[21] Itoh, M., and Chua, L. O. **Memristor Oscillators**. *International Journal of Bifurcation and Chaos 18*, 11 (November 2008), 3183 – 3206.

[22] Kennedy, M. P. **Robust Op-Amp Realization of Chua's Circuit**. *Frequenz 46*, 3-4 (March-April 1992), 66–80.

[23] Khalil, H. K. ***Nonlinear Systems***. Prentice Hall, New Jersey, 2001.

[24] Kocarev, L., Parlitz, U., and Stojanovski, T. An Application Of Synchronized Chaotic Dynamic Arrays. *Physical Letters A 217* (1996), 280–284.

[25] Li, Z., and Xu, D. **A Secure Communication Scheme Using Projective Chaos Synchronization**. *Chaos, Solitons and Fractals 22*, 2 (2004), 477–481.

[26] Lorenz, E. N. **Deterministic Nonperiodic Flow**. *Journal of Atmospheric Sciences 20* (January 1963), 130–141.

[27] Makin, J. G. **A Computational Model of Human Blood Clotting: Simulation, Analysis, Control, and Validation**. Tech. Rep. UCB/EECS-2008-165, Electrical Engineering and Computer Sciences Department, University of California, Berkeley, Berkeley, California, Dec. 2008.

[28] Matsumoto, T., Chua, L. O., and Komuro, M. **The Double Scroll**. *IEEE Transactions on Circuits and Systems 32* (August 1985), 797–818.

[29] Medrano, R. O., Baptista, M. S., and Caldes, I. L. **Homoclinic Orbits in a Piecewise System and Their Relation with Invariant Sets**. *Physica D 183* (2003), 133–147.

[30] Muthuswamy, B. Memristor Based Chaotic Circuit - Mathematica Demonstration. *http://demonstrations.wolfram.com/MemristorBasedChaoticSystem/* (November 2009).

[31] Muthuswamy, B., and Chang, J. **Optimal CNN Templates for Linearly Separable Cellular Automata**. *International Journal of Bifurcation and Chaos 17*, 3 (April 2007), 747 – 791.

[32] Muthuswamy, B., and Kokate, P. P. **Memristor-Based Chaotic Circuits**. *IETE Technical Review 26*, 6 (November 2009), 415–426.

[33] Rossler, O. E. **An Equation for Continuous Chaos**. *Physics Letters 57A*, 5 (1976), 397–398.

[34] Roy, P. K., and Basuray, A. **A High Frequency Chaotic Signal Generator: A Demonstration Experiment**. *American Journal of Physics 71* (2003), 34–37.

[35] Sarkar, P. **A Brief History of Cellular Automata**. *ACM Computing Surveys 32*, 1 (March 2000), 80–107.

[36] Shilnikov, L. P. **A Case of the Existence of a Countable Number of Periodic Motions**. *Sov. Math. Doklady 6* (January 1965), 163–166.

[37] Silva, C. P. **Shilnikov's Theorem - A Tutorial**. *IEEE Transactions on Circuits and Systems 40*, 10 (October 1993), 675–682.

[38] Siu, S. LET. *http://www.mathworks.com/matlabcentral/fileexchange/* (June 2008).

[39] Slotine, J. J. E., and Li, W. ***Applied Nonlinear Control***. Prentice-Hall, New Jersey, 1991.

[40] Strogatz, S. H. ***Nonlinear Dynamics and Chaos with Applications to Physics, Biology and Engineering***. Perseus Books Publishing, Massachusetts, 1994.

[41] Strukov, D. B., Snider, G. S., Stewart, G. R., and Williams, R. S. **The Missing Memristor Found**. *Nature 453* (2008), 80–83.

[42] Sudheer, K. S., and Sabira, M. **Adaptive Function Projective Synchronization of two-cell Quantum-CNN Chaotic Oscillators with Uncertain Parameters**. *Physics Letters A 373* (2009), 1847–1851.

[43] Wiggins, S. ***An Introduction to Dynamical Systems and Chaos***. Springer, New York, 2003.

[44] Wolf, A., Swift, J. B., Swinney, H. L., and Vastano, J. A. Determining Lyapunov Exponents from a Time Series. *Physica 16D* (1985), 285–317.

[45] Wolfram, S. ***Theory and Applications of Cellular Automata: Including Selected Papers 1983-1986***. WorldScientific Inc., New Jersey, 1986.

[46] Wolfram, S. ***A New Kind of Science***. Wolfram Media, Champaign, Ilinois, 2002.

[47] Wolfram, S., Martin, O., and Odlyzko, A. M. **Algebraic Properties Of Cellular Automata**. *Communications in Mathematical Physics 93* (March 1984), 219–258.

[48] Yang, X. S. **Topological Horseshoes And Computer Assisted Verification of Chaotic Dynamics**. *International Journal of Bifurcation and Chaos 19*, 4 (2009), 1127–1145.

[49] Yang, X. S., and Yangi, F. **A Rigorous Verification of Caos in an Inertial Two-Neuron System**. *Chaos, Solitons and Fractals 20* (2004), 587=591.

[50] Zhong, G. **Implementation of Chua's Circuit with a cubic nonlinearity**. *IEEE Transactions on Circuits and Systems 41*, 12 (1994), 934–941.

Appendix A

Mathematica 6.0 and MATLAB R2007b Code

Note: Each numbered equation must be entered on a separate line in Mathematica or MATLAB. If there is no numbering, then the entire script file should be used in Mathematica or MATLAB.

A.1 The Lorenz Butterfly

Mathematica 6.0 simulation code for Fig. 1.2

$$\begin{aligned} lorenz = NDSolve[\{x'[t] == -\sigma * x[t] + \sigma * y[t], y'[t] == \rho * x[t] - y[t] - x[t] * z[t],\\ z'[t] == -\beta * z[t] + x[t] * y[t], x[0] == 10, y[0] == 20, z[0] == 30\}/.\{\sigma \rightarrow 10, \rho \rightarrow 28,\\ \beta \rightarrow \frac{8}{3}\}, \{x, y, z\}, \{t, 0, 100\}, MaxSteps \rightarrow \infty] \end{aligned} \tag{A.1}$$

$$\begin{aligned} ParametricPlot3D[\{x[t], y[t], z[t]\}]/.lorenz, \{t, 0, 100\},\\ AxesLabel \rightarrow \{Style[''x'', Large, Bold],\\ Style[''y'', Large, Bold], Style[''z'', Large, Bold]\}] \end{aligned} \tag{A.2}$$

A.2 Four Element Chua's circuit

Mathematica 6.0 simulation code for Fig. 2.2 and Fig. 2.3

$$g_2[v\_] := 9.33 * 10^{-4} * v + (\frac{-5 * 10^{-4} - 9.33 * 10^{-4}}{2}) * (Abs[v + 1] - Abs[v - 1]) \tag{A.3}$$

$$fourElement = NDSolve[\{v_1'[t] == \frac{(i[t] - g_2[v_1[t]])}{C_1}, i'[t] == \frac{(v_1[t] - v_2[t])}{\kappa * L_1},$$
$$v_2'[t] == \frac{\kappa * i[t]}{C_2}, v_1[0] == 0.1, i[0] == 0, v_2[0] == 0\}/.\{C_1 \to 33 * 10^{-9},$$
$$C_2 \to 100 * 10^{-9}, L_1 \to 10 * 10^{-3}, \kappa \to 8.33\},$$
$$\{v_1, i, v_2\}, \{t, 0, 100 * 10^{-3}\}, MaxSteps \to \infty] \quad (A.4)$$

$$ParametricPlot[Evaluate[\{v_1[t], v_2[t]\}/.fourElement], \{t, 0, 100 * 10^{-3}\},$$
$$AxesLabel \to \{``v_1(t)\ (volts)'','' v_2(t)\ (volts)''\}, PlotPoints \to 10000] \quad (A.5)$$

$$Plot[Evaluate[i[t]/.fourElement], \{t, 0, 5 * 10^{-3}\},$$
$$AxesLabel \to \{``t\ (seconds)'','' i(t)\ amps''\}] \quad (A.6)$$

A.3 Four Element Chua's circuit - eigenvalues and eigenspaces

Mathematica 6.0 simulation code for Fig. 2.11 The command below plots U_{-1} and U_1

$$ParametricPlot3D[\{Tooltip[\{-1, u, v\}, ''U_{-1} \triangleq -1'', Tooltip[\{1, u, v\}, ''U_1 \triangleq 1'']\},$$
$$\{u, -3, 3\}, \{v, -3, 3\}, PlotRange \to \{\{-3, 3\}, \{-3, 3\}, \{-3, 3\}\},$$
$$AxesLabel \to \{''x'', ''y'', ''z''\}, Mesh \to None, PlotStyle \to Opacity[0.5]] \quad (A.7)$$

The command below plots the eigenspace corresponding to the complex conjugate eigenvalues at the origin.

$$Show[\%\%, Plot3D[Tooltip[\frac{-9.3}{2.4} * x + \frac{-12.9}{2.4} * y, ''E^s(0) \triangleq 9.3x + 12.9y + 2.4z = 0'',$$
$$\{x, -1, 1\}, \{y, -3, 3\}, PlotRange \to \{\{-1, 1\}, \{-3, 3\}, \{-3, 3\}\},$$
$$Mesh \to None, ClippingStyle \to None, PlotStyle \to Opacity[0.5]] \quad (A.8)$$

The command below adds the unstable eigenspace through the origin along with the points representing the origin and the other equilibrium points.

$$Show[\%\%, ParametricPlot3D[Tooltip[\{-0.817 * t, -0.045 * t, 0.574 * t\},$$
$$''E^u(0) \triangleq t(-0.817, -0.045, 0.574)''], \{t, -1, 1\},$$
$$PlotStyle \to Directive[Orange, Thick]], Graphics3D[\{PlotStyle \to Directive[Black],$$
$$PointSize[Large], Point[\{1.16, 0, -1.16\}]\}],$$
$$Graphics3D[\{PlotStyle \to Directive[Black], PointSive[Large],$$
$$Point[\{-1.16, 0, 1.16\}]\}]] \quad (A.9)$$

Next, we will plot the eigenspaces corresponding to P^+. First we will plot the plane associated with the complex conjugate eigenvale, next we will plot the eigenspace corresponding to the real eigenvalue.

$$\begin{gathered}Show[\%\%, Plot3D[Tooltip[\frac{0.129}{0.017}, \frac{-0.1}{0.017} * x + \frac{0.153}{0.017} * y, \\ "E^u(P^+) \triangleq 0.1x - 0.153y + 0.017z = 0.129"], \{x, 1, 3\}, \{y, -3, -3\}, \\ PlotRange \rightarrow \{\{1, 3\}, \{-3, 3\}, \{-3, 3\}\}, \\ Mesh \rightarrow None, ClippingStyle \rightarrow None, PlotStyle \rightarrow Opacity[0.5]], \\ ParametricPlot3D[Tooltip[\{-0.913t + 1.16, 0.054t, 0.425t - 1.16], "E^s(P^+) \triangleq \\ t(-0.903, 0.054, 0.425) + (1.16, 0, -1.16)"], t, -1, 0.5, \\ PlotStyle \rightarrow Directive[Blue, Thick]] \end{gathered} \tag{A.10}$$

Finally, plot the eigenspaces corresponding to P^-. First we will plot the plane associated with the complex conjugate eigenvale, next will plot the eigenspace corresponding to the real eigenvalue.

$$\begin{gathered}Show[\%\%, Plot3D[Tooltip[\frac{-0.129}{0.017}, \frac{-0.1}{0.017} * x + \frac{0.153}{0.017} * y, \\ "E^u(P^+) \triangleq 0.1x - 0.153y + 0.017z = -0.129"], \{x, -3, -1\}, \{y, -3, -3\}, \\ PlotRange \rightarrow \{\{-3, -1\}, \{-3, 3\}, \{-3, 3\}\}, \\ Mesh \rightarrow None, ClippingStyle \rightarrow None, PlotStyle \rightarrow Opacity[0.5]], \\ ParametricPlot3D[Tooltip[\{-0.913t - 1.16, 0.054t, 0.425t + 1.16], "E^s(P^+) \triangleq \\ t(-0.903, 0.054, 0.425) + (-1.16, 0, 1.16)"], t, -0.5, 1, \\ PlotStyle \rightarrow Directive[Blue, Thick]] \end{gathered} \tag{A.11}$$

A.4 Mathematica code for the Four-Element Bifurcation Diagram

```
Manipulate[Module[{sol, iterate, pts, fsol},
  sol[c] :=
   NDSolve[{x'[t] ==
      a*(y[t] - (0.29*x[t] +
           0.5*(-0.16 - 0.29)*(Abs[x[t] + 1] - Abs[x[t] - 1]))),
     y'[t] == x[t] - c*y[t] + z[t], z'[t] == -69.39*y[t],
     x[0] == 0.01, y[0] == 0, z[0] == 0}, {x, y, z},{t, 0, 300},
    MaxSteps -> Infinity];
  iterate =
   Compile[c, {fsol = sol[c];
     Map[{c, #} &,
      FindMaxValue[z[t] /. fsol, {t, #, # + 1}] & /@
       Table[i, {i, 100, 200.0, 1}]]}];
  pts = Quiet[
    Flatten[Table[iterate[c, {c, 1, 0.8, -0.0005}],
     1]]; ListPlot[pts,
   PlotStyle ->
    Table[{PointSize[0.01], RGBColor[.49, 0, 0]}, {i, 1,
      Length[pts]}], Frame -> True, ImageSize -> {400, 350},
   FrameLabel -> {Style["c",
      Italic], Style["z", Italic]}, ImageSize -> {500, 500},
   AspectRatio -> 1, ImagePadding -> {{35, 10}, {35, 10}}]],
 {{a, 25.24, "a"}, 25, 26, 0.24, Appearance -> "Labeled"},
 ControlPlacement -> Top, SynchronousUpdating -> False]
```

A.5 MATLAB simulation code for Canonical Memristor Based Chaotic Circuit

The file below is called Canonical.m. The corresponding W.m is shown after Canonical.m

```
%% Memristor based chaotic Chua’s circuit simulation
%% Bharathwaj Muthuswamy,
%% Pracheta Kokate
%% June 13th 2008 - July 13th 2008,
%% June 2009
%% mbharat@cory.eecs.berkeley.edu
%% Ref: Stephen Lynch,
%% Dynamical Systems with Applications
%% using MATLAB
clear;
%% MAKE SURE YOU PUT CODE BELOW ON A
%% SINGLE LINE!
dmemristor=inline('[y(2);1/5.5e-9*
((y(3)-y(2))/1428
- W(y)*y(2));1/49.5e-9*((y(2)-y(3))/1428
- y(4)); y(3)/7.07e-3]','t','y');

options = odeset('RelTol',1e-7,'AbsTol',
1e-7);
```

```
[t,ya]=ode45(dmemristor,[0 10e-3],
[0,0,0,0.1],options);


plot(ya(:,2),ya(:,4));
title('Memristor Attractor: 2D Projection,
 i vs. v1')
fsize=15;
xlabel('v1(t)','Fontsize',fsize);
ylabel('i(t)','Fontsize',fsize);
figure
plot(ya(:,2),ya(:,3))
title('Memristor Attractor: 2D Projection,
 v2 vs. v1')
xlabel('v1(t)','Fontsize',fsize);
ylabel('v2(t)','Fontsize',fsize);
figure
plot(ya(:,2),ya(:,1))
title('Memristor Attractor: 2D Projection,
phi vs. v1')
xlabel('v1(t)','Fontsize',fsize);
ylabel('phi(t)','Fontsize',fsize);
figure
plot(ya(:,3),ya(:,1))
title('Memristor Attractor: 2D Projection,
phi vs. v2')
xlabel('v2(t)','Fontsize',fsize);
ylabel('phi(t)','Fontsize',fsize);

figure
plot(ya(:,4),ya(:,1))
title('Memristor Attractor: 2D Projection,
phi vs. i')
xlabel('i(t)','Fontsize',fsize);
ylabel('phi(t)','Fontsize',fsize);

figure
plot(ya(:,3),ya(:,4))
title('Memristor Attractor: 2D Projection,
i vs. v2')
xlabel('v2(t)','Fontsize',fsize);
ylabel('i(t)','Fontsize',fsize);
```

```
figure
plot(t,ya(:,2))
hold
plot(t,ya(:,4),'r')
title('Memristor chaotic time series:
v1 (blue) and v2 (red)')
figure
plot(t,ya(:,1))
hold
plot(t,ya(:,3),'r')
title('Memristor chaotic time series:
w (blue) and i (red)')
%% 3d plot: flux, current and voltage
figure
plot3(ya(:,1),ya(:,2),ya(:,3));
grid on
xlabel('w(t)','Fontsize',fsize);
ylabel('v1(t)','Fontsize',fsize);
zlabel('i(t)','Fontsize',fsize);
title('Memristor 3D attractor');

%% Menductance function W.m
function r = W(y)
    if(y(1) <= -1)
        r = -0.5e-3;
    elseif((y(1) > -1) && (y(1) < 1))
        r = -0.8e-3;
    else
        r = -0.5e-3;
end
```

A.6 MATLAB simulation code for Rescaled Canonical Memristor Based Chaotic Circuit

Use the same W.m as in the previous canonical memristor based chaotic circuit simulation code.

```
%% Memristor based chaotic Chua's circuit
%% simulation
%% Bharathwaj Muthuswamy, Pracheta Kokate
%% June 13th 2008 - July 13th 2008,
%% June 2009
%% mbharat@cory.eecs.berkeley.edu
%% Ref: Stephen Lynch,
%% Dynamical Systems with Applications
%% using MATLAB
clear;
%% MAKE SURE YOU PUT CODE BELOW ON A SINGLE LINE!
dmemristor=inline('[y(2); 1/5.5e-9*((y(3)-y(2))/1428-
W(y)*y(2)); 1/49.5e-9*((y(2)-y(3))/1428 -
y(4)); y(3)/7.07e-3]','t','y');

 options = odeset('RelTol',1e-7,'AbsTol',1e-7);
[t,ya]=ode45(dmemristor,[0 10e-3],[0,0,0,0.1],options);
ya(:,1) = ya(:,1);
ya(:,2) = ya(:,2)./10000;
ya(:,3) = ya(:,3)./5000;
ya(:,4) = ya(:,4)./500;

plot(ya(:,2),ya(:,4));
title('Memristor Attractor: 2D Projection, i vs. v1')
fsize=15;
xlabel('v1(t)','Fontsize',fsize);
ylabel('i(t)','Fontsize',fsize);
figure
plot(ya(:,2),ya(:,3))
title('Memristor Attractor: 2D Projection, v2 vs. v1')
xlabel('v1(t)','Fontsize',fsize);
ylabel('v2(t)','Fontsize',fsize);
figure
plot(ya(:,2),ya(:,1))
title('Memristor Attractor: 2D Projection,
 phi vs. v1')
xlabel('v1(t)','Fontsize',fsize);
ylabel('phi(t)','Fontsize',fsize);
figure
plot(ya(:,3),ya(:,1))
title('Memristor Attractor: 2D Projection,
phi vs. v2')
xlabel('v2(t)','Fontsize',fsize);
ylabel('phi(t)','Fontsize',fsize);
```

```
figure
plot(ya(:,4),ya(:,1))
title('Memristor Attractor: 2D Projection,
 phi vs. i')
xlabel('i(t)','Fontsize',fsize);
ylabel('phi(t)','Fontsize',fsize);

figure
plot(ya(:,3),ya(:,4))
title('Memristor Attractor: 2D Projection,
 i vs. v2')
xlabel('v2(t)','Fontsize',fsize);
ylabel('i(t)','Fontsize',fsize);

figure
plot(t,ya(:,2))
hold
plot(t,ya(:,4),'r')
title('Memristor chaotic time series:
v1 (blue) and v2 (red)')
figure
plot(t,ya(:,1))
hold
plot(t,ya(:,3),'r')
title('Memristor chaotic time series:
 w (blue) and i (red)')
%% 3d plot: flux, current and voltage
figure
plot3(ya(:,1),ya(:,2),ya(:,3));
grid on
xlabel('w(t)','Fontsize',fsize);
ylabel('v1(t)','Fontsize',fsize);
zlabel('i(t)','Fontsize',fsize);
title('Memristor 3D attractor');
```

A.7 MATLAB simulation code for four element memristor based chaotic circuit

There are two files: the ode solver (memristorAudio.m) and the memristance function (W.m). Shown below is memristorAudio.m:

```
%% Memristor based chaotic Chua's circuit
%% simulation
%% Bharathwaj Muthusway,
%% Pracheta Kokate
%% June 13th 2008 - July 13th 2008,
%% June 2009
%% mbharat@cory.eecs.berkeley.edu
%% Ref:  Stephen Lynch,
%% Dynamical Systems with Applications
%% using MATLAB
clear;
%% MAKE SURE YOU PUT CODE BELOW ON A
 %% SINGLE LINE!
dmemristor=inline('[y(2);(y(3)-
W(y)*y(2))/33e-9;
(y(2)-y(4))/(8.33*10e-3);
y(3)*(8.33/100e-9)]','t','y');
options = odeset('RelTol',1e-7,
'AbsTol',1e-7);
[t,ya]=ode45(dmemristor,[0 100e-3],
[0,0.1,0,0],options);
plot(ya(:,2),ya(:,4))
title('Memristor Attractor: 2D Projection,
v2 vs. v1')
fsize=15;
xlabel('v1(t)','Fontsize',fsize);
ylabel('v2(t)','Fontsize',fsize);
figure
plot(ya(:,2),ya(:,3))
title('Memristor Attractor: 2D Projection,
 i vs. v1')
xlabel('v1(t)','Fontsize',fsize);
ylabel('i(t)','Fontsize',fsize);
figure
plot(ya(:,2),ya(:,1))
title('Memristor Attractor: 2D Projection,
 phi vs. v1')
xlabel('v1(t)','Fontsize',fsize);
ylabel('phi(t)','Fontsize',fsize);
figure
plot(ya(:,3),ya(:,1))
title('Memristor Attractor: 2D Projection,
 phi vs. i')
xlabel('i(t)','Fontsize',fsize);
ylabel('phi(t)','Fontsize',fsize);
figure
plot(t,ya(:,2))
hold
plot(t,ya(:,4),'r')
title('Memristor chaotic time series:
 v1 (blue) and v2 (red)')
figure
plot(t,ya(:,1))
hold
plot(t,ya(:,3),'r')
title('Memristor chaotic time series:
 w (blue) and i (red)')

%% 3d plot: flux, current and voltage
figure
plot3(ya(:,1),ya(:,2),ya(:,3));
grid on
xlabel('w(t)','Fontsize',fsize);
ylabel('v1(t)','Fontsize',fsize);
zlabel('i(t)','Fontsize',fsize);
title('Memristor 3D attractor');
```

Shown below is W.m:

```
%% Menductance functions for use with
%% memristorAudio.m and
%% memristorAudioSIMPLEST.m
%% Bharathwaj Muthusway,
%% Pracheta Kokate
%% July 17th 2008,
%% June 2009
%% mbharat@cory.eecs.berkeley.edu

function r=W(y)
    if y(1) <= -1.5e-4
        r = 43.25e-4;
    elseif y(1) > -1.5e-4 && y(1) <= -0.5e-4
        r = -9.33*y(1)-9.67e-4;
    elseif y(1) > -0.5e-4 && y(1) < 0.5e-4
        r = -5.005e-4;
    elseif y(1) >= 0.5e-4 && y(1) < 1.5e-4
        r = 9.33*y(1)-9.67e-4;
    else
        r = 43.25e-4;
    end
end
```

The circuit parameters above were chosen such that the circuit frequencies are in the audio range. To listen to sounds of chaos, use the MATLAB command:

```
soundsc(ya(:,1),44000)
```

A.8 MATLAB simulation code for four element memristor based chaotic circuit with single negative element

The file below is called memristorAudioSIMPLEST.m, use the same W.m in the previous section.

```
%% Memristor based chaotic Chua’s circuit
%% simulation
%% Bharathwaj Muthusway, Pracheta Kokate
%% June 13th 2008 - July 3rd 2008
%% June 2009
%% mbharat@cory.eecs.berkeley.edu
%% Ref:  Stephen Lynch,
%% Dynamical Systems with Applications
%% using MATLAB

clear;
%% MAKE SURE CODE BELOW IS ON A SINGLE LINE.
%% W(y) is the same function from the
%% previous appendix
dmemristor=inline(’[y(2);(y(3)-W(y)*y(2))/33e-9;
(y(2)-y(4))/(1*-10e-3);
y(3)*(1/100e-9)]’,’t’,’y’);

options = odeset(’RelTol’,1e-7,’AbsTol’,
1e-7);
[t,ya]=ode45(dmemristor,[0 100e-3],
[0,0.1,0,0],options);
```

```
plot(ya(:,2),ya(:,4))
title('Memristor Attractor: 2D Projection,
v2 vs. v1')
fsize=15;
xlabel('v1(t)','Fontsize',fsize);
ylabel('v2(t)','Fontsize',fsize);

figure
plot(ya(:,2),ya(:,3))
title('Memristor Attractor: 2D Projection,
 i vs. v1')
xlabel('v1(t)','Fontsize',fsize);
ylabel('i(t)','Fontsize',fsize);

figure
plot(ya(:,2),ya(:,1))
title('Memristor Attractor: 2D Projection,
phi vs. v1')
xlabel('v1(t)','Fontsize',fsize);
ylabel('phi(t)','Fontsize',fsize);

figure
plot(ya(:,3),ya(:,1))
title('Memristor Attractor: 2D Projection,
 phi vs. i')
xlabel('i(t)','Fontsize',fsize);
ylabel('phi(t)','Fontsize',fsize);

fsize=15;
xlabel('v1(t)','Fontsize',fsize);
ylabel('v2(t)','Fontsize',fsize);

figure
plot(t,ya(:,2))
hold
plot(t,ya(:,4),'r')
title('Memristor chaotic time series:
v1 (blue) and v2 (red)')

figure
plot(t,ya(:,1))
```

```
hold
plot(t,ya(:,3),'r')
title('Memristor chaotic time series:
 w (blue) and i (red)')

%% 3d plot: flux, current and voltage
figure
plot3(ya(:,1),ya(:,2),ya(:,3));
grid on
xlabel('w(t)','Fontsize',fsize);
ylabel('v1(t)','Fontsize',fsize);
zlabel('i(t)','Fontsize',fsize);
title('Memristor 3D attractor');
```

A.9 Lyapunov Exponent programs

The Lyapunov exponent program for the four-element memristor-based chaotic circuit is shown below.

```
function OUT = fourElementMemristor(t,X)
%MEMRISTOR Model of memristor based four
%Element chaotic circuit

% Settings:
% ODEFUNCTION: fourElementMemristor
% Final Time:  1000, Step: 0.01,
% Relative & Absolute Tol: 1e-007
% No. of discarded transients: 100,
% update Lyapunov: 10
% Initial Conditions: 0 0.1 0 0,
% no. of linearized ODEs: 16

% The first 4 elements of the input data X
% correspond to the
% 4 state variables.  Restore them.
% The input data X is a 12-element vector
% in this case.
% Note: x is different from X
w = X(1); x = X(2);y = X(3);z = X(4);

%% MAKE SURE CODE IS ON A SINGLE LINE!
```

```
% Parameters.
L1 = 10e-3;C1=33e-9;C2=100e-9;k=8.33;

% ODE
dw = (sqrt(L1*C2))*x;
dx = (sqrt(L1*C2)/C1)*(y-fourElementW(w)*x);
% dy = (sqrt(L1*C2)/(k*L1))*(x-z);
% comment dy above and uncomment dy
% below for ONE negative element
% this single negative element chaotic
% circuit may be the simplest
% possible four dimensional and
% four-element chaotic circuit
% ALSO, YOU NEED TO CHANGE JACOBIAN!
k = 1;
dy = (sqrt(L1*C2)/(k*-L1))*(x-z);
% end uncomment code
dz = ((sqrt(L1*C2)*k)/C2)*y;

% Q is a 4 by 4 matrix, so it has 12
% elements.
% Since the input data is a column
% vector, rearrange the last 12
% elements of the input data in a
% square matrix.

Q = [X(5), X(9),  X(13),X(17);
     X(6), X(10), X(14),X(18);
     X(7), X(11), X(15),X(19);
     X(8), X(12), X(16),X(20)];

% Linearized system (Jacobian)
 J = [ 0 sqrt(L1*C2) 0 0;
        0 -(sqrt(L1*C2)/C1)*fourElementW(w)
        sqrt(L1*C2)/C1 0
        %0 (sqrt(L1*C2)/(k*L1))
        % 0 -(sqrt(L1*C2)/(k*L1))
        % replace row above with
        % row below for one negative element
        % memsristor circuit
        0 -(sqrt(L1*C2)/(k*L1)) 0
(sqrt(L1*C2)/(k*L1))
```

```
        0 0 ((sqrt(L1*C2)*k)/C2) 0];

% Multiply J by Q to form a variational
% equation
F = J*Q;

OUT = [dw;dx; dy; dz; F(:)];
end
```

The Lyapunov exponent program for the canonical memristor-based chaotic circuit is shown below.

```
function OUT = fourDMemristorCanonical(t,X)
% Lyapunov exponent computation for
% Four-D Canonical Memristor

% Settings:  ODEFUNCTION: fourdm
% Final Time:  10000, Step: 1,
% Relative & Absolute Tol: 1e-007
% No. of discarded transients: 100,
% update Lyapunov: 10
% Initial Conditions: 0 0 0 2e-5,
% no. of linearized ODEs: 16

% The first 4 elements of the input
% data X correspond to the
% 4 state variables.  Restore them.
% The input data X is a 12-element
% vector in this case.
p = X(1); q = X(2);r = X(3);s = X(4);
% time scaling
tau = 1/sqrt(7.07e-3*49.5e-9);
% ODE
dp = (q*10e3)/tau;
dq = (1/(tau*5.5e-9))*(r/(2*1428)-q/1428
-canonicalW(p)*q);
dr = (1/(tau*49.5e-9))*((2*q)/1428-r/1428
-s/10);
ds = (10*r)/(tau*7.07e-3);

% Q is a 4 by 4 matrix, so it has
% 12 elements.
% Since the input data is a column
% vector, rearrange
% the last 12 elements of the input
% data in a square matrix.

Q = [X(5), X(9),  X(13),X(17);
     X(6), X(10), X(14),X(18);
     X(7), X(11), X(15),X(19);
     X(8), X(12), X(16),X(20)];

% Linearized system (Jacobian)
  J = [ 0,10e3/tau,0,0;
        0,-(1/(tau*5.5e-9))*(1/1428+
        canonicalW(p)),
        (1/(tau*5.5e-9))*(1/(2*1428)),0;
        0,(1/(tau*49.5e-9))*(2/1428),
        -(1/(tau*49.5e-9))*(1/1428),
        -(1/(tau*49.5e-9))*(1/10);
         0,0,10/(tau*7.07e-3),0];

% Multiply J by Q to form a variational
%equation
F = J*Q;

OUT = [dp; dq; dr; ds;F(:)];
end
```

For the Time-Series method, we use the same programs above, but the call functions are different and are given below. First is the call function for the four-element memristor-based chaotic circuits followed by the call function for the canonical memristor-based chaotic circuit.

```
options = odeset('RelTol',1e-7,'AbsTol',1e-7);
[T,Res]=lyapunov(4,@fourElementMemristor,@ode45,
0,0.01,1000,[0 0.1 0 0],10);
plot(T,Res);
title('Dynamics of Lyapunov exponents');
xlabel('Time'); ylabel('Lyapunov exponents');

options = odeset('RelTol',1e-7,'AbsTol',1e-7);
[T,Res]=lyapunov(4,@fourDMemristorCanonical,
@ode45,options,0,1,10000,[0 0 0 2e-5],1);
```

```
plot(T,Res);
title('Dynamics of Lyapunov exponents');
xlabel('Time'); ylabel('Lyapunov exponents');
```

A.10 Mathematica Code for Rigorous Verification of Chaos in Memristor Chaotic Circuit

$$Q[\phi\_] := -0.5*10^{-3}*\phi+0.5*(-0.8*10^{-3}+0.5*10^{-3})*(Abs[\phi+1]-Abs[\phi-1]) \quad (A.12)$$

$$W[\phi\_] := \begin{cases} -0.5*10^{-3} & \phi \leq -1 \\ -0.8*10^{-3} & -1 < \phi < 1 \\ -0.5*10^{-3} & \phi \geq 1 \end{cases} \quad (A.13)$$

$$\begin{gathered}
Manipulate[Module[\{memristorCircuit\}, \\
With[\{tx = a\}, \\
memristorCircuit = \\
Quiet@NDSolve[\{\phi'[t] == v_1[t], \\
v_1'[t] == 1/(5.5 * 10^{-9})((v_2[t] - v_1[t]) \\
1428 - W\phi[t]] * v_1[t]), \\
v_2'[t] == 1/(49.5 * 10^{-9}) * ((v_1[t] - v_2[t]) \\
1428 - i[t]), \\
i'[t] == v_2[t]/(7.07 * 10^{-3}), \phi[0] == 0, \\
v_1[0] == 0.0, v_2[0] == 0, \\
i[0] == -0.2\}, \{\phi, v_1, v_2, \\
i\}, \{t, 0, 50 * 10^{-3}\}, MaxSteps \rightarrow Infinity]; \\
Show[ParametricPlot3D[\\
Evaluate[\{v_1[t]/5000, \phi[t], i[t]/1000\}/. \\
memristorCircuit], \{t, 0, tx\}, \\
AxesLabel- > \{''v_1(t)(volts)'', \\
''\phi(t)(weber)'', ''i(t)(amps)''\}, \\
PlotRange \rightarrow \{\{-10, 10\}, \{-10, 10\}, \{-0.1, 0.1\}\}, \\
BoxRatios \rightarrow \{1, 0.5, 1\}], \\
ParametricPlot3D[\{x, -1, z\}, \{x, -7, 7\}, \{z, -0.1, 0.1\}, \\
Mesh \rightarrow None, PlotStyle \rightarrow Opacity[1], \\
PlotRange \rightarrow \{\{-10, 10\}, \{-10, 10\}, \{-0.1, 0.1\}\}], \\
Switch[First[Evaluate[\phi[tx]/.memristorCircuit]] > -1, True, \\
Graphics3D[\{PointSize[Large], Blue, \\
Point[First[\\
Evaluate[\{v_1[tx]/5000, \phi[tx], i[tx]/ \\
1000\}/.memristorCircuit]]]\}], \_, \\
Graphics3D[\{PointSize[Large], Yellow, \\
Point[First[\\
Evaluate[\{v_1[tx]/5000, \phi[tx], i[tx]/ \\
1000\}/.memristorCircuit]]]\}]]]]], \{a, 0.1 * 10^{-3}, 50 * 10^{-3}, 0.05 * 10^{-3}\}]
\end{gathered} \tag{A.14}$$

A.11 Mathematica Demonstration for Cubic Memristor Based Chaotic Circuit

Mathematica 6.0 simulation code for Fig. 3.22

$$
\begin{gathered}
Manipulate[[Module[\{sol = \{\phi, v_1, v_2, i\}/. \\
Quiet[NDsolve[\\
\{\phi'[t] == -v_1[t] * 1000, \\
v_1'[t] == \frac{1}{C_1}\left(\frac{v_2[t]-v_1[t]}{R} - \left(\left(-0.599 * 10^{-3} + 0.0677 * 10^{-3} * \phi[t]^2 * 3\right) * v_1[t]\right)\right), \\
v_2'[t] == \frac{1}{C_2}\left(\frac{v_1[t]-v_2[t]}{R} - i[t]\right), i'[t] == \frac{v_2[t]}{L}, \phi[0] == 0, \\
v_[1] = 0.11, v_2[0] = 0.11, i[0] == 0.0\}/.\{C_1 \rightarrow C1val, C_2 \rightarrow C2val, L \rightarrow Lval, \\
R \rightarrow Rval\}, \{\phi, v_2, v_2, i\}, \{t, 0, tmax\}, MaxSteps \rightarrow Infinity]][[1]]\}, \\
Grid[\{\{ParametricPlot[\{First[sol][t], First[Rest[Rest[sol]]][t]\}, \{t, 0, tmax\}, \\
AxesLabel \rightarrow \{"\phi[t]", "v_2[t]"\}, ImageSize \rightarrow \{150, 150\}, AspectRatio \rightarrow 1], \\
PlotRange \rightarrow All, MaxRecursion \rightarrow 8, PerformanceGoal \rightarrow 8], \\
Plot[\{First[sol][t], \{t, 0, tmax\}, AxesLabel \rightarrow \{"t", "\phi[t]"\}, \\
ImageSize \rightarrow \{150, 150\}, PlotRange \rightarrow All, MaxRecursion \rightarrow 8, \\
PerformanceGoal \rightarrow 8], ParametricPlot3D[\{First[Rest[sol]][t], \\
First[Rest[Rest[sol]]][t], First[sol][t]\}, \{t, 0, tmax\}, AxesLabel \rightarrow \{"v_1[t]", "v_2[t]", \\
"\phi[t]"\}, ImageSize \rightarrow \{150, 150\}, BoxRatios \rightarrow \{1, 1, 1\}, PlotRange \rightarrow Full, \\
MaxRecursion \rightarrow 8, PerformanceGoal \rightarrow 8, ColorFunction \rightarrow "Rainbow"]\}\}]], \\
Text[Style["CircuitParameters", FontSize \rightarrow Medium, FontWeight \rightarrow Bold]], \\
\{\{Rval, 1.8 * 10^3, "R \in \{1800\Omega, 1801\Omega, ..., 2400\Omega\}"\}, 1.8 * 10^3, 2.4 * 10^3, 1\}, \\
\{\{Lval, 18{*}10^{-3}, "L \in \{18\ mH, 18.5\ mH, ..., 24\ mH\}"\}, 18{*}10^{-3}, 24{*}10^{-3}, 0.5{*}10^{-3}\}, \\
\{\{C2val, 68{*}10^{-9}, "C_2 \in \{68\ nF, 68.5\ nF, ..., 74\ nF\}"\}, 68{*}10^{-9}, 74{*}10^{-9}, 0.5{*}10^{-9}\}, \\
\{\{C1val, 6.8 * 10^{-9}, "C_1 \in \{6.8\ nF, 6.85\ nF, ..., 7.4\ nF\}"\}, 6.8 * 10^{-9}, 7.4 * 10^{-9}, \\
0.05 * 10^{-9}\}, \{\{tmax, 0.01, "t \in \{5 * 10^{-5}, 10e-5, ..., 0.01\ sec, ..., 0.1\ sec\}"\}, \\
5 * 10^{-5}, 0.1, 5 * 10^{-5}\}, Text[Style["2D\ Phase\ Portrait,\ Time\ Domain \\
Waveform\ and\ a\ 3D\ Phase\ Portrait", \\
FontSize \rightarrow Medium, FontWeight \rightarrow Bold]], SynchronousUpdating \rightarrow False]
\end{gathered}
\tag{A.15}
$$

A.12 Mathematica code for Cellular Automata evolution and Integer Factorization

Mathematica 6.0 simulation code for chapter 3.

```
(* Function to find attractors and ALL basins. *)
(* Usage: BartCA[105,3] *)

BartCA[rule_,bitLength_] :=
Module[{n,l,DynamicsList,p,initPattern,x},
    (* Declare some utility functions first *)
    (* The function below converts the input in decimal to a *)
       (* list of its binary equivalent suitable for use in
          CellularAutomaton *)
    DecimalToBinaryList[n_] := IntegerDigits[n, 2, bitLength];

    (* Convert a binary number (represented as a list) to a decimal
       integer *)
    BinaryToDecimal[l_] := FromDigits[l, 2];

    (* The function below can be used to convert the output from
       CellularAutomaton to integers *)
    BinaryToDecimalEvolution[DynamicsList_] :=
     Map[BinaryToDecimal, DynamicsList];

    (* Functions below puts the output in the format we want:
    {basin,{attractor}} OR {attractor} *)
    (* getEvolutionListAndAttractorPositions works because the
      last element in each evolution list is *)
    (* repeated again somewhere in the evolution list.  Hence, if
      we will get an array of two elements if we *)
    (* find the positions of the last element in each evolution list.*)
    getEvolutionListAndAttractorPositions[l1_] :=
     Join[{l1}, {Flatten[Position[l1, Last[l1]]]}];

    (* Now, lets say we have the following evolution list and
       attractor positions in the list: *)
    (* LP = {{{0,0},{1,2}},{{1,3,7,13,7},{3,5}}} *)
    (* We want: {{{},{0}},{{1,3},{7,13,7}}} *)
          evolutionFormat[LP_] :=
          {Flatten[Take[LP[[1]],LP[[2]][[1]]-1]],
          Take[LP[[1]],LP[[2]]]};
          (* Function below is obvious: it computes one iteration
             of an evolution rule *)
      (* by running CellularAutomaton once *)
    OneIteration[p_] := Last[CellularAutomaton[rule,p,1]];
          (*  Run until you hit an attractor using the NestWhileList
               function *)
   CARunUntilAttractor[initPattern_] :=
    NestWhileList[OneIteration,initPattern, UnsameQ, All];
    x = Table[i,{i,0,(2^bitLength)-1}];

    Map[evolutionFormat,Map[getEvolutionListAndAttractorPositions,
    Map[BinaryToDecimalEvolution,Map[CARunUntilAttractor,
    (Map[DecimalToBinaryList,x])]]]]];


(* Function to find unique attractors *)
(* Format: {{attractor 1},{attractor 2},...} *)
(* Example: {{0},{3,12},{6,9}} *)
(* Usage:  GetUniqueAttractors[BartCA[105,3]]*)

GetUniqueAttractors[evolutionList_] :=
Module[{tempList},
(* Utility Function to check for empty list *)
CheckIfEmptyQ[l_] := Length[l[[1]]] == 0;

(* Mathematica example function to take union of elements WITHOUT
sorting *)
UnsortedUnion[x_]:=Module[{f},f[y_]:=(f[y]=Sequence[];y);f/@x];

(* This utility function finds the position of the smallest number
 in each attractor *)
(* We use this as a predicate in NestWhile to rotate the attractor
 until the smallest *)
(* number is the head of the attractor *)
FirstPosSmallestQ[l_] := Position[l, Min[l]] != {{1}};

RotateUntil[l_] := NestWhile[RotateLeft, l, FirstPosSmallestQ];

(* If the first element is empty, then we know that we have an
   attractor *)
(* Then, we just extract the second element from each list,
   that is the attractor. *)
(* Next, we flatten each element in the list for nicer formatting *)
(* By taking an UnsortedUnion, we remove the last duplicate entry
   in each attractor *)
(* We then sort the list so that attractors of the same period are
   grouped *)
(* Then, we just use RotateUntil so that we can finally take the
   Union to get the unique attractors *)
   Union[Map[RotateUntil,Sort[Map[UnsortedUnion,
   Map[Flatten,Map[Rest,Select[evolutionList,
   CheckIfEmptyQ]]]]]]]];

(* Function to find attractor periods *)
(* Output format:  {{no-of-attractors of period p1,
{Period, p1}},{no-of-attractors of period p2,{Period, p2}},...*)
(* Example:  {{1, {Period, 1}},{2,{Period, 2}}} *)
(* Usage: FindAttractorPeriods[GetUniqueAttractors[BartCA[105,3]]] *)

FindAttractorPeriods[UniqueAttractorList_] :=
Module[{},
OutputFormat[l_] := {Length[l],{Period,First[l]}};
Map[OutputFormat,Split[Map[Length,UniqueAttractorList]]]];
```