

POLITECNICO DI TORINO

TESI DI LAUREA MAGISTRALE

Memristor Oscillators for Bio-Inspired Analogue Computing

Autore:
Francesco MARRONE

Relatore:
Prof. Fernando CORINTO

*Tesi in adempimento a quanto richiesto per il conseguimento
della Laurea Magistrale in Ingegneria Elettronica*

svolta presso

Linear and Nonlinear Circuits & Systems
Dipartimento di Elettronica e Telecomunicazioni

April 3, 2018

POLITECNICO DI TORINO

Abstract

Dipartimento di Elettronica e Telecomunicazioni

Laurea Magistrale in Ingegneria Elettronica

Memristor Oscillators for Bio-Inspired Analogue Computing

by Francesco MARRONE

The proposed thesis investigates a novel area of research that makes use of a disruptive technology as the fundamental computation element, the memristor introduced by Prof. Leon Chua more than forty years ago. The memristor is a non-linear two-terminal circuit element capable of tuning and retaining its state of resistance (memristance). Its characteristic equation is defined in terms of charge q (integral of the current) and flux φ (integral of the voltage) as $q = f(\varphi)$ (or $\varphi = f(q)$). It turns out that the analogy between memristor and biological synapses is related to the ions' dynamics in the membrane, and the ionic-electronic flow due to the drift-diffusion in organic/inorganic semiconductors. The first characterization of memristor device at the HP Lab in 2008, has spurred research in this area forward and has led to the creation of other memristor devices. More importantly it seems that the adaptive properties of memristors are ideal for use in neural network and neuromorphic engineering applications. The ultimate goal of the thesis is to investigate ultra-dense arrays of memristor oscillators for low-power, highly integrated and portable neuromorphic real-time systems. In particular, nonlinear dynamic behavior of memristors is exploited in oscillatory and chaotic circuits. A thorough study is necessary to understand the rich complex nonlinear phenomena emerging in memristor circuits. The systematic description (mainly based on the network theory technique referred to as the tableau method) leads to large systems of nonlinear DAEs, whose solution requires efficient numerical simulation tools. In the thesis a novel systematic methodology for the analysis of nonlinear circuits containing memristors is exploited. The main advantage of the proposed method is that it enables to describe memristor-based circuits by means of Initial Values Problems for a reduced number of ODEs compared to current approaches available. This permits to simplify the investigation of nonlinear dynamic behavior and bifurcations without parameters in memristor circuits and to make clear the influence of initial conditions. The presence of suitable pulses in the circuits permits to tune the periodic and chaotic nonlinear dynamic behavior. This may permit to apply memristor technological innovation towards analogue computing, establishing unconventional associative memristive memories, but also novel computational formalisms. An architecture embedding a Hopfield neural network is proposed to function as content addressable memory. The obtained chaotic nonlinear dynamic behavior is exploited in order to let the NN escape local minima and converge towards global minimum of its associated energy function. The results suggest that chaos can enable an enhancement in successful reconstruction rate if conveniently introduced in the network. This confirms the latest research works available in literature extending them to a classical problem as corrupted data reconstruction.

Acknowledgements

I would like to thank Prof. Fernando Corinto for having introduced me to these fascinating and frontier research topics and for all the useful directions given whenever I was stuck.

I would like to thank all my friends, those met here at the university and those who have always been by me, for their precious support in hard times and for all the unforgettable laughs and moments lived together. A special thank to Fabio whose presence was fundamental during these months.

Last but not least I thank my family for their material and emotional support during the last 25 years, for bringing me here never influencing my way but always lighting up the path, for all the trust and for making me who I am.

Contents

Abstract	iii
Acknowledgements	v
1 Introduction	1
1.1 Introduction to dynamical systems	1
1.1.1 Dynamical systems definitions	1
State equation	1
Phase space	2
1.1.2 Conservative dynamical systems	3
1.1.3 Dissipative dynamical systems	3
1.1.4 Attractors	3
1.1.5 Linear stability analysis	3
1.1.6 Limit cycles	4
Poincaré map	4
1.2 Bifurcation phenomena	5
1.2.1 Saddle-node bifurcation	5
1.2.2 Transcritical bifurcation	6
1.2.3 Pitchfork bifurcation	7
Supercritical bifurcation	7
Subcritical bifurcation	7
1.2.4 Hopf bifurcation	8
1.3 Introduction to chaos theory	9
1.3.1 Landau's theory of turbulence	9
1.3.2 Chaos theory	9
Strange attractors	10
1.3.3 Lyapounov exponents	10
1.3.4 Transition towards chaos	11
Period-doubling cascade	11
Intermittency	12
2 Memristive systems	15
2.1 Introduction to memristive systems	15
2.1.1 The original theorization	15
2.1.2 The hallmark of all memristors	17
2.2 Taxonomy of memristive systems	18
2.2.1 Current controlled extended memristor	19
2.2.2 Current controlled generic memristor	19
2.2.3 Current controlled ideal memristor	20
2.3 Memristor as nonvolatile memory element	20
2.4 Flux-charge analysis method	21
2.4.1 Constitutive relations for two-terminal elements in $\varphi - q$ domain	21
Ideal independent voltage source	21

	Ideal independent current source	22
	Ideal resistor	22
	Ideal capacitor	23
	Ideal inductor	23
	Ideal flux controlled memristor	24
	Ideal charge controlled memristor	24
2.4.2	Incremental Kirchoff Laws	25
2.4.3	DAEs in the flux-charge domain	25
2.4.4	State equations in flux-charge domain	26
2.5	Traditional voltage-current analysis method	27
2.6	Bifurcation phenomena in memristive circuits	27
2.6.1	Invariant manifolds	27
2.6.2	Fixed-invariant manifold	28
2.6.3	Bifurcations without parameters	28
3	Memristive Chua circuit	29
3.1	Memristive Chua circuit	29
3.1.1	Analysis in (v-i) domain	29
3.1.2	Analysis in Flux-Charge domain	30
3.2	Circuit implementation	35
3.2.1	Inductance emulation	35
3.2.2	Ideal memristor emulation	35
3.2.3	Negative resistance	36
3.2.4	Current-Voltage state equations	36
3.2.5	Flux-Charge state equations	36
	Numerical simulations	38
	Circuit simulation	42
	Pulse programming	45
3.2.6	Physical implementation	46
	Dealing with nonidealities	46
	Circuit simulation	47
	Laboratory acquired measurements	51
4	Application of chaos	53
4.1	Artificial Neural Networks	53
4.2	Recurrent Neural Networks	53
4.3	Hopfield Neural Networks	53
4.3.1	Hopfield NN topology	54
4.3.2	State update	54
4.3.3	Energy function	55
4.3.4	Training	55
	Hebbian learning rule	56
	Storkey learning rule	56
	Spurious patterns	56
	Network capacity	56
4.4	Introduction of chaos in Hopfield NN	57
4.4.1	The dataset, the network and the training	57
4.4.2	Simulation methodology	57
	The amplitude of fluctuations	59
	Example of pattern reconstruction enhanced by chaotic thresh- olds	60

5 Conclusions and future work	63
5.1 The obtained results	63
5.2 Future work	63
Bibliography	65

List of Figures

1.1	Example of phase space	2
1.2	Direct saddle-node bifurcation diagram	6
1.3	Inverse saddle-node bifurcation diagram	6
1.4	Transcritical bifurcation diagram	7
1.5	Supercritical bifurcation diagram	7
1.6	Subcritical bifurcation diagram	8
1.7	Hopf bifurcation diagram	9
1.8	Rössler attractor example	10
1.9	Subharmonic cascade example 1	12
1.10	Subharmonic cascade example 2	12
1.11	Subharmonic cascade example 3	12
2.1	2-terminal devices incomplete table	15
2.2	2-terminal devices complete table	16
2.3	Memristor pinched-loop testing setup	17
2.4	Memristor pinched v - i curves examples	18
2.5	Ideal voltage source in $\varphi - q$ domain	22
2.6	Ideal current source in $\varphi - q$ domain	22
2.7	Ideal resistor in $\varphi - q$ domain	23
2.8	Ideal capacitor in $\varphi - q$ domain	23
2.9	Ideal inductor in $\varphi - q$ domain	24
2.10	Ideal flux controlled memristor in $\varphi - q$ domain	24
2.11	Ideal charge controlled memristor in $\varphi - q$ domain	25
2.12	Algebraical and dynamical subnetworks separation	26
3.1	Memristive Chua circuit	29
3.2	Memristive Chua circuit in $\varphi - q$ domain	30
3.3	Bifurcation on a fixed manifold in $\varphi - q$ domain	33
3.4	Bifurcation on a fixed manifold in $i - v$ domain	33
3.5	Bifurcation without parameters in $\varphi - q$ domain	34
3.6	Bifurcation without parameters in $i - v$ domain	34
3.7	Antoniou circuit	35
3.8	Ideal memristor emulator	35
3.9	Negative resistance	36
3.10	Equilibrium point stability 1	39
3.11	Bifurcation without parameters of the implemented oscillator in $\varphi - q$ domain 1	39
3.12	Bifurcations without parameters of the implemented oscillator in $i - v$ domain 1	40
3.13	Equilibrium points stability 2	40
3.14	Bifurcations without parameters of the implemented oscillator in $\varphi - q$ domain 1	41

3.15	Bifurcations without parameters of the implemented oscillator in $i - v$ domain 2	41
3.16	Implemented circuit full schematic LTspice	42
3.17	Ideal operational amplifier implementation	42
3.18	LTspice simulation 1	43
3.19	LTspice simulation 2	43
3.20	LTspice simulation 3	43
3.21	LTspice simulation 4	44
3.22	LTspice simulation 5	44
3.23	LTspice simulation 6	44
3.24	Pulse programming of the implemented circuit	45
3.25	2D phase portrait from LTspice simulation of the pulse programming of the implemented circuit	45
3.26	Waveforms from LTspice simulation of the pulse programming of the implemented circuit	46
3.27	Finite and infinite gain integrators comparison	47
3.28	Effect of feedback resistance on integrator transfer function	47
3.29	Implemented circuit full schematic SIMetrix	48
3.30	SIMetrix simulation 1	48
3.31	SIMetrix simulation 2	49
3.32	SIMetrix simulation 3	49
3.33	SIMetrix simulation 4.0	49
3.34	SIMetrix simulation 4.1	50
3.35	SIMetrix simulation 4.2	50
3.36	SIMetrix simulation 4.3	50
3.37	Picture of the circuit mounted in laboratory	51
3.38	Waveforms from the circuit mounted in laboratory	51
3.39	Phase space projection from the circuit mounted in laboratory	52
4.1	Hopfield network example	54
4.2	Dataset	57
4.3	Energy levels without chaotic thresholds	58
4.4	Energy levels with chaotic thresholds	58
4.5	Chaotic waveform injected into the thresholding units	59
4.6	Hopfield neural network simulation setup schematic	59
4.7	Success rates for different KC and corruption values	60
4.8	Enhanced pattern recovery example	61

A Leda e Tino da sempre lanterne del mio errare...

Chapter 1

Introduction

1.1 Introduction to dynamical systems

Dynamical systems analysis is about the study of the evolution in time of a set of measurable quantities that describe a system. The first step is to find that set for the system under study and find a mathematical description of their temporal behaviour. At the end of the first step a set of differential equations (ODEs) describing the system is obtained. In the following only systems described by ordinary differential equations (derivatives only in the time variable) are treated. The main goal is not to find exact analytical solutions of the ODEs (which in most cases do not exist), but to have insights on the possible long term behaviours.

1.1.1 Dynamical systems definitions

In the following the fundamental definitions regarding dynamical systems used in the whole manuscript are reported. A vector $\mathbf{X} = (x_1, x_2, \dots, x_n)^T \in \mathbb{R}^n$, called the *state vector* is considered to be the measurable set of quantities that characterize the system.

State equation

The set of first order ODEs that describe the evolution in time of the systems is called *state equation*. If higher order ODEs appear during the modeling of the system, the model can always be reformulated as first order introducing additional dimensions (state variables in \mathbf{X}).

The state equation can be written in explicit form as [1.1](#).

$$\begin{cases} \frac{dx_1}{dt} = f_1(x_1, x_2, \dots, x_n, t) \\ \frac{dx_2}{dt} = f_2(x_1, x_2, \dots, x_n, t) \\ \dots \\ \frac{dx_n}{dt} = f_n(x_1, x_2, \dots, x_n, t) \end{cases} \quad (1.1)$$

The SE can also be written in vectorial form as [1.2](#).

$$\frac{d\mathbf{X}}{dt} = \mathbf{F}(\mathbf{X}, t) \quad (1.2)$$

Where $\mathbf{F} : \mathbb{R}^{n+1} \rightarrow \mathbb{R}^n$ is a vector field defined as.

$$\mathbf{F}(\mathbf{X}, t) = (f_1(\mathbf{X}, t), f_2(\mathbf{X}, t), \dots, f_n(\mathbf{X}, t))^T \quad (1.3)$$

Considering the initial value problem 1.4 under the assumption that each f_i in F is differentiable in an open set $\Omega \subset \mathbb{R}^n$ and $x_0 \in \Omega$.

$$\begin{cases} \frac{dX}{dt} = F(X, t) \\ X(0) = x_0 \end{cases} \quad (1.4)$$

The Cauchy-Lipschitz theorem asserts that the problem has an unique solution $x(t)$ defined on a time interval which may be either finite or infinite. This theorem is the mathematical formalization of the practical concept of determinism which states that given an initial condition only one future is possible.

The system in 1.2 is called **non-autonomous**, if $\frac{dX}{dt} = F(X, t)$ the system called **autonomous**. The dependency on time of the field F can be removed by increasing by one the dimension of the state vector X obtaining a system described by the state equation $\frac{dX'}{dt} = F'(X')$.

Phase space

The space \mathbb{R}^n , which X belongs to, is said the **phase space**. n is obviously the system's number of **degrees of freedom**. The solution of IVP 1.4 $x(t)$ can be numerically found by integration, and the continuous of values it assumes in the phase space for each instant t of definition is called the system's **trajectory**.

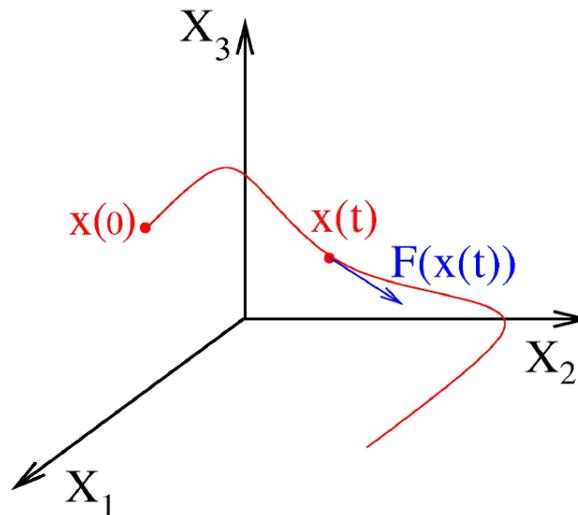


FIGURE 1.1: A trajectory in the phase space for a system with $n = 3$ degrees of freedom [1].

As shown in figure 1.1 the vector field F is always tangent to the trajectory. The visualizaion of the phase space is possible, obviously, for $1 \leq n \leq 3$; when $n \geq 4$ only projection of trajectories on lower dimension space is possible. Trajectories starting from different x_0 do never intersect. This constraint is a logical consequence of determinism (Cauchy-Lipschitz theorem), if it was not true then assuming the intersection point as a new initial condition two possible trajectories would be possible hence two possible futures from the same starting condition. The points in the phase space where many trajectories seem to intersect actually are convergence point and only at $t = +\infty$ the intersection takes place. When a trajectory is projected on a lower dimension space apparent intersections may appear.

1.1.2 Conservative dynamical systems

A conservative dynamical system is an ideal system that preserves indefinitely its energy, from a physical point of view it does neither radiate energy nor increase its internal entropy. From a dynamical analysis prospective this means that F conserves volumes, formally written as 1.5.

$$\nabla F = 0 \quad (1.5)$$

1.1.3 Dissipative dynamical systems

The dissipative class of dynamical systems represent real physical phenomena. The dynamical condition for a system to be dissipative is that on average F contracts volumes. It can be formally written as 1.6.

$$\nabla F < 0 \quad (1.6)$$

1.1.4 Attractors

Attractors are subsets of the phase space where nearby trajectories tend to converge to during the transient. Asymptotically all the trajectories close enough to the attractor settle on the attractor. Formally an attractor is a subset of the phase space invariant for F . Attractors are possible only for dissipative systems, under this condition it follows that the volume of an attractor is 0 (volume contraction for dissipative F). Conservative systems hold the initial energy associated with their initial conditions thus they cannot tend to other energy states. The *basin of attraction* is the set of all the initial conditions which generate trajectories asymptotically converging to the attractor.

The following attractors exist:

- *Stable fixed point*: it is a stationary state x^* where $F(x^*) = 0$. It can exist for any system of dimension $n \geq 1$.
- *Limit cycle*: it is a time dependent solution represented by a closed trajectory in the phase space. It can exist for any system of dimension $n \geq 2$.
- *Chaotic attractor*: it is a time dependent solution of great interest which will be discussed later in this chapter. For its existence the system must either be of dimension $n \geq 3$ and F must be a non linear field vector or be a discrete map ($n \geq 1$).

1.1.5 Linear stability analysis

In order to get an idea of what is the behaviour of trajectories nearby a fixed point x^* the linearization of the SE near x^* is the approach to use. Considering a point in proximity of x^* : $x = x^* + \delta x$ being $\delta x(0)$ "small", the qualitative behaviour near x^* is given by the time evolution of $\delta x(t)$. All the intermediate steps of Taylor first order expansion of the vector field F are omitted and only the final formulation is reported.

Considering J_F the *jacobian matrix* of F then for $\delta x(0)$ sufficiently small (higher order contributions negligible) the linear SE 1.7 describes the evolution of $\delta x(t)$.

$$\frac{d\delta x}{dt} = J_F(x^*)\delta x \quad (1.7)$$

The n eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_n \in \mathbb{C}^n$ of the jacobian matrix evaluated in x^* dictate the behaviour of the trajectories in a neighbourhood of x^* .

The general rule for the stability of a fixed point is:

- If at least one eigenvalue has real part **strictly positive**, then the fixed point is **unstable**.
- If all the eigenvalues have real part **strictly negative**, then the fixed point is **stable**.
- If all the eigenvalues have real part **strictly negative** but at least one has null real part, then the fixed point stability cannot be derived from linear analysis. The nonlinearity determines the stability of the system.

A detailed taxonomy of fixed points based on the stability properties exists which is omitted here for conciseness' sake. The specific names of fixed point are introduced when needed.

1.1.6 Limit cycles

A closed isolated trajectory in the phase space is called *limit cycle*. A limit cycle can be either stable or unstable. In the former case trajectories nearby the cycle will spiral towards it in the latter case those trajectories will spiral away from it. The form, the amplitude and the frequency of the limit cycle is independent from the initial conditions.

Closed orbits also exist in conservative systems (correspond to center varieties, all the eigenvalues lay on the imaginary axis) but they are in infinite number nested into each other and their amplitudes are determined by initial conditions only. Center varieties, by definition, are not limit cycles.

Poincaré map

In order to study the dynamic of systems with dimension $n > 2$ the *Poincaré map* is the tool to use. By intersecting the higher dimensional trajectory with a surface (generally a plane of dimension $(n - 1)$) a set of intersection points called the *Poincaré section* is obtained. For different orientations of the plane in the phase space different patterns (sections) are obtained.

The Poincaré map T 1.8 is a continuous map that links a point of the section to the next one.

$$P_{k+1} = T(P_k) = T(T(P_{k-1})) = T^2(P_{k-1}) = T^{k+1}(P_0) \quad (1.8)$$

The higher dimensional continuous F can be transformed in a discrete function T that has the same topological properties of the original vector field.

For a limit cycle an appropriate choice of the $(n - 1)$ dimensional plane Λ leads to a Poincaré section showing an isolate point $P^* = T(P^*)$. The study of the stability of the limit cycle reduces to the study of stability of a fixed point for a discrete function. By considering a point $P_0 = P^* + \delta P_0$ with δP_0 small enough the function $T : \Lambda \rightarrow \Lambda$ can be linearized. Considering the jacobian matrix $J_T \in \mathbb{R}^{(n-1) \times (n-1)}$ associated with T , omitting all the steps of the linearization of T , the series of distances δP_k (analogous of $\delta x(t)$) can be written as 1.9.

$$\delta P_k \simeq J_T(P^*) \delta P_0 \quad (1.9)$$

The key result for the study of stability of a limit cycle are:

- If all the eigenvalues of $J_T(P^*)$ have their module strictly less than 1 the limit cycle is stable
- If at least one of the eigen values of $J_T(P^*)$ is in module greater than 1 the limit cycle is unstable
- If all the eigenvalues of $J_T(P^*)$ are in module less or equal to 1 and at least 1 of them has module equal to 1 then the linearization of the T map cannot be used for determining the stability of P^*

1.2 Bifurcation phenomena

The vector field F may depend on parameters (multiplicative factors). The time evolution of the system may vary by changing those parameters. For a dissipative system the nature and number of attractor may vary as well by a variation of the parameters. This phenomenon of changing behaviour is called *bifurcation*. Only the simple case of a vector field F depending on 1 parameter only is reported in the following. The bifurcations reported are *local* and of *codimension 1* meaning that only a limited parte of the phase space is involved in the change.

Near a bifurcation the system SEs can be reduced to a *normal form* which is the a simple generic mathematical expression resulting from a change of variables and a reduction of dimensionality.

The value of the parameter that causes the bifurcation is called *critical value*.

1.2.1 Saddle-node bifurcation

A *saddle point* is a subclass of fixed points always unstable. For example it is obtained in a $n = 2$ dimensional system when J_F has 2 real non-null distinct roots of opposite sign.

A *node* is a subclass of fixed points that may be stable or unstable. For example it is obtained in a $n = 2$ dimensional system when J_F has 2 real non-null distinct roots of same sign, if the sign is positive the node is unstable otherwise it is stable.

The *saddle-node* bifurcation is practically the apparition or anihilation of a pair of fixed points. Its direct normal form can be written as 1.10.

$$\frac{dx}{dt} = \mu - x^2 \quad (1.10)$$

In the direct saddle-node bifurcation when the parameter $\mu < 0$ no fixed points exist while when $\mu > 0$ a stable fixed point and an unstable fixed point do exist. $\mu = 0$ is the *critical value*. The bifurcation diagram for a direct saddle-node bifurcation is repoted in figure 1.2.

Its inverse normal form can be written as 1.11.

$$\frac{dx}{dt} = \mu + x^2 \quad (1.11)$$

In the inverse saddle-node bifurcation when the parameter $\mu > 0$ no fixed points exist while when $\mu < 0$ a stable fixed point and an unstable fixed point do exist. $\mu = 0$ is the *critical value*. The bifurcation diagram for an inverse saddle-node bifurcation is repoted in figure 1.3.

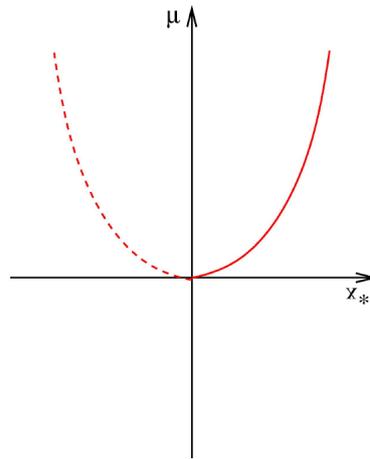


FIGURE 1.2: Bifurcation diagram for a direct saddle-node bifurcation.

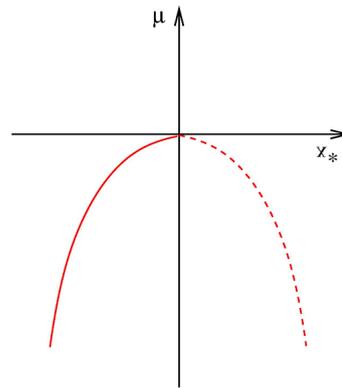


FIGURE 1.3: Bifurcation diagram for an inverse saddle-node bifurcation

1.2.2 Transcritical bifurcation

The *transcritical* bifurcation is practically the exchange of stability between two fixed points. Its normal form can be written as 1.12.

$$\frac{dx}{dt} = \mu x - x^2 \quad (1.12)$$

The critical value of the parameter is $\mu = 0$. The bifurcation diagram for a transcritical bifurcation is reported in figure 1.4.

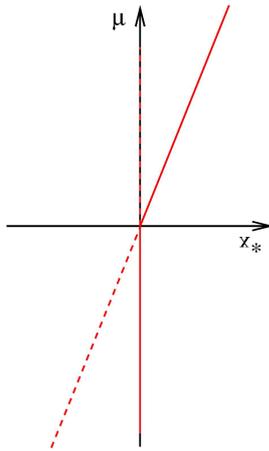


 FIGURE 1.4: Bifurcation diagram for a transcritical bifurcation.

1.2.3 Pitchfork bifurcation

The *pitchfork* bifurcation can be either *supercritical* or *subcritical*.

Supercritical bifurcation

The *supercritical* bifurcation is practically the loss of stability of one fixed point and the birth of two new stable fixed points for the critical value of the parameter μ . Its normal form can be written as 1.13.

$$\frac{dx}{dt} = \mu x - x^3 \quad (1.13)$$

The critical value of the parameter is $\mu = 0$. The bifurcation diagram for a supercritical bifurcation is reported in figure 1.5.

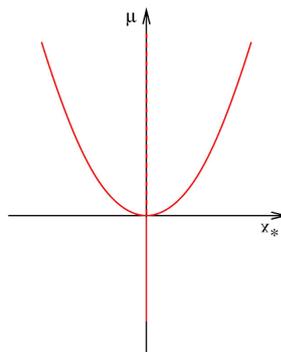


 FIGURE 1.5: Bifurcation diagram for a supercritical bifurcation.

Subcritical bifurcation

The *subcritical* bifurcation is trickier than the *supercritical* one. Its normal form involves an higher order term term as in 1.14.

$$\frac{dx}{dt} = \mu x + x^3 - x^5 \quad (1.14)$$

For $\mu < -\frac{1}{4}$ only one stable fixed point $x_0^* = 0$ exists. For $\mu \in [-\frac{1}{4}, 0]$ x_0^* maintains its stability and four additional fixed points come into existence two of which are stable while the other two are unstable. For $\mu \geq 0$ only 3 fixed points remain, x_0^* loses its stability while the two other stable fixed points for $\mu \in [-\frac{1}{4}, 0]$ maintain their stability. The bifurcation diagram for a supercritical bifurcation is reported in figure 1.6.

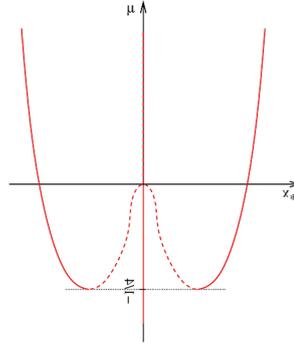


FIGURE 1.6: Bifurcation diagram for a subcritical bifurcation.

As shown in figure 1.6 in the interval $\mu \in [-\frac{1}{4}, 0]$ 3 stable solution coexist (bistability). By tuning μ one way and the other around the values $-\frac{1}{4}$ and 0 an hysteresis cycle will be observed.

1.2.4 Hopf bifurcation

The *Hopf* bifurcation is practically the emergence of a limit cycle from a stationary one. The limit cycle is a bidimensional object, thus a bidimensional normal form is needed in order to describe the bifurcation. This can be done by writing the normal form using a complex variable $z \in \mathbb{C}$ as in 1.15.

$$\frac{dz}{dt} = (\mu + i\gamma)z - z|z|^2 \quad (1.15)$$

By writing in cartesian coordinates $z = x + iy$ 1.15 can be rewritten as 1.16.

$$\begin{cases} \frac{dx}{dt} = \mu x - \gamma y - x(x^2 + y^2) \\ \frac{dy}{dt} = \mu y + \gamma x - y(x^2 + y^2) \end{cases} \quad (1.16)$$

By studying the sign of the eigenvalues of 1.16 it can be found that the point $(0, 0)$ is stable for $\mu < 0$ and unstable for $\mu > 0$. The behaviour in the neighbourhood of $(0, 0)$ changes from a stable spiral attractor (both complex eigenvalues with non-null negative real part) to an unstable spiral (both complex eigenvalues with non-null positive real part).

In order to understand what happens for $\mu > 0$ it is more convenient to write 1.15 using polar coordinates $z = re^{i\theta}$ obtaining 1.17.

$$\begin{cases} \frac{dr}{dt} = \mu r - r^3 \\ \frac{d\theta}{dt} = \gamma \end{cases} \quad (1.17)$$

The modulus equation in 1.17 is a normal form for the pitchfork supercritical bifurcation. The solution for $\mu > 0$ is a stable periodic solution with constant modulus

$\sqrt{\mu}$ and linear phase $\theta = \gamma t + \theta_0$.

The bifurcation diagram for Hopf bifurcation is reported in 1.7.

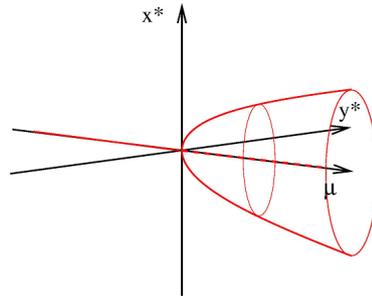


FIGURE 1.7: Bifurcation diagram for a Hopf bifurcation [1].

Also a *subcritical Hopf* bifurcation exists. The results are the same, just for opposite values of μ .

1.3 Introduction to chaos theory

Before expanding the topic of *chaotic attractors* introduced in 1.1.4 a first distinction has to be made. Both noisy signals and chaotic signals are irregular by nature but while a noisy signal is stochastic, a chaotic signal is deterministic. In reality all dynamical systems experience influence from the stochastic environment they are in.

1.3.1 Landau's theory of turbulence

The accepted theory for the description of the emergence of chaos between the 40's and the 70's was a vision based on quasi-periodicity. This phenomenon is caused by a series of successive Hopf bifurcations which take place by varying a parameter of the system. Each bifurcation adds more and more new frequencies to the spectrum. After several subsequent Hopf bifurcations, with a finite resolution spectrum analyzer, the observed spectrum seems to be continuous. The trajectories of this kind of systems are on a high-dimensional torus in a high-dimensional phase space.

This theory is based on the idea that chaos can arise only in high dimensional systems.

1.3.2 Chaos theory

At the end of the 70's the chaos theory was formulated. In contrast with Landau's theory, the chaos theory states that chaos is not due to the high dimensionality of a system but it is an intrinsic characteristic that can appear even in 3-dimensional nonlinear deterministic continuous dynamical systems.

A common definition [6] of chaos states that, to classify a dynamical system as chaotic, it must have the following properties:

- It must be **sensitive to initial conditions**: an arbitrarily small perturbation of the current trajectory may lead to very different future behaviour.
- It must be **topologically mixing**: the system evolves over time so that any given region of its phase space eventually overlaps with any other given region.

- It must have trajectories which are **periodic** and **dense**: every point in the phase space is approached arbitrarily closely by periodic orbits.

Strange attractors

The chaotic trajectories often are confined only in a subset of phase space. Unlike stable fixed point attractors and limit cycles, the attractors emerging from chaotic systems, called strange attractors have great complexity. Strange attractor can exist both in some discrete systems and in continuous dynamical systems.

The former ones have no dimensional constraint for showing chaotic dynamic, the latter ones on the other hand must be either infinite dimensional or at least three dimensional nonlinear systems.

The key feature of those attractors is the sensitivity to initial conditions: two initially close trajectories will diverge one from the other after some time. The distance will grow exponentially with time.

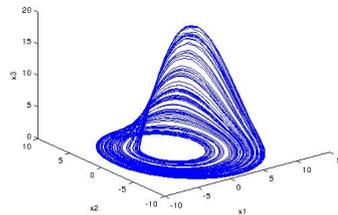


FIGURE 1.8: An example of strange attractor: the Rössler attractor.[1]

In figure 1.8 an example of Rössler attractor is reported. Strange attractors need in their topology something that allows divergence of the trajectories, a feature called *stretching*. Because of their attractive nature, all the trajectories need to stay in a bounded part of the phase space, the trajectories must be re-injected in the same subspace. This feature is called *folding*.

Due to successive folding and stretching a strange attractor has a foliated structure whose dimension cannot be an integer. The attractor is a fractal object.

1.3.3 Lyapounov exponents

The divergence between trajectories is quantified using the *Lyapounov exponents*. The only way to obtain those is by numeric integration. A possible algorithm for computing the exponents can be summarized as following:

1. Integrate the system with a random initial condition for a time long enough to extinguish the transient.
2. The end point of the previous integration $\mathbf{X}(0)$ becomes the new starting condition for a new trajectory $\mathbf{X}(t)$. Considering a point $\mathbf{X}'(0)$ very close to $\mathbf{X}(0)$ and the trajectory $\mathbf{X}'(t)$ stemming from it, define $\delta(t) = \|\mathbf{X}'(t) - \mathbf{X}(t)\|$ the distance at time t between the two trajectories.
3. Consider the evolution in time $\delta(t)$. If it evolves exponentially $\delta(t) \sim \delta(0)e^{t\lambda}$ then λ is the Lyapounov exponent.

Actually there are as many Lyapounov exponents as many dimensions of the system. If at least one Lyapounov exponent is positive then the system is unambiguously

chaotic. The behaviour of $\delta(t)$ is determined by the largest exponent, called **the** Lyapunov exponent. In order to obtain a good estimation of the exponents the process must be repeated on several trajectories $X'(t)$.

1.3.4 Transition towards chaos

A lot of different ways to transit towards chaos exist. Only two of them are reported in the following as examples. From a mathematical point of view, the study of a one-dimensional map can lead to a very good understanding of the mechanism that underly chaos. Successive bifurcations can lead to chaos when a parameter is tuned.

Period-doubling cascade

Starting from a periodic behaviour of period T , the increase of a parameter leads to a first bifurcation to a $2T$ periodic regime. Each successive bifurcation doubles the period. $2^n T$ periodic regimes are observed with increasing n until a value of the parameter is reached for which the system becomes chaotic.

The logistic map is the function in 1.18, it can be a Poincaré map obtained from a dynamics of higher dimension.

$$x_{n+1} = f(x_n) = rx_n(1 - x_n) \quad (1.18)$$

Omitting the details, considering $r \in [0, 4]$ and $x_i \in [0, 1]$, the *period doubling* transition goes in the following way:

- $r \in [0, 1]$: a unique stable fixed point $x_0 = 0$ exists.
- $r \in [1, 3]$: the fixed point x_0 loses its stability but there is a new stable fixed point $x_1 = \frac{r-1}{r}$.
- $r \in [3, 3.449]$: x_1 loses its stability through a pitchfork bifurcation and the system oscillates between two values.
- $r \in [3.449, 3.569946\dots]$: the cascade of period doubling pitchfork bifurcations takes place by increasing r in this range.
- $r \in [3.569946\dots, 4]$: either chaotic or periodic behaviour can be observed depending on the value r . Subranges of r exist for which the regime is periodic, the largest of those is the $3T$ window.

In figure 1.9 the whole bifurcation diagram is reported. In figure 1.10 the period-doubling cascade leading to chaos is more evident than figure 1.9. In figure 1.11 the periodic windows are more evident than in figure 1.10.

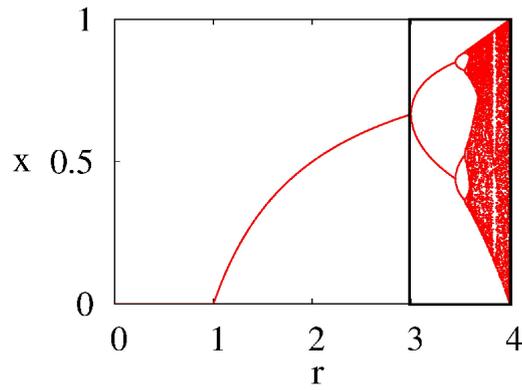


FIGURE 1.9: Entire bifurcation diagram in for $r \in [0, 4]$ [1]

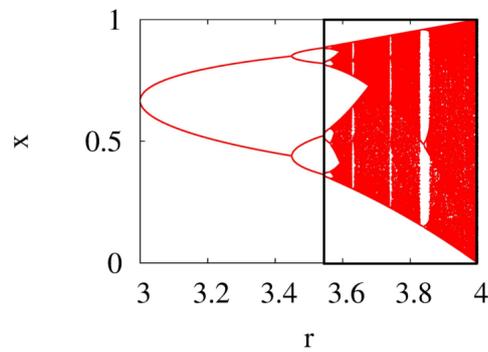


FIGURE 1.10: Entire bifurcation diagram in for $r \in [3, 4]$ [1]

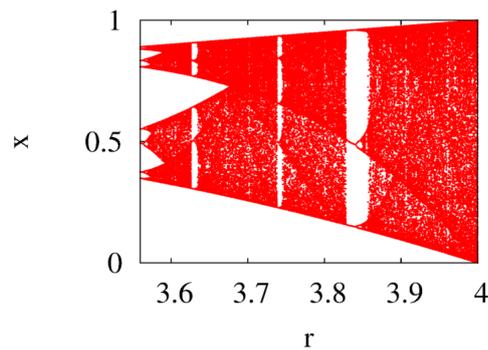


FIGURE 1.11: Entire bifurcation diagram in for $r \in [3.569946\dots, 4]$ [1]

Intermittency

By tuning the bifurcation parameter bursts of irregular behaviour appear interrupting the regular oscillation. As the parameter is tuned towards chaos more and more burst appear until a chaotic regime is reached.

A *type I intermittency* can be described by the following function:

$$x_{n+1} = f(x_n) = x + \epsilon + x^2 \quad (1.19)$$

When the parameter ϵ changes from a small negative value to a small positive one a saddle-node bifurcation takes place and one stable fixed point annihilates with an unstable fixed point. For $\epsilon \geq 0$ as the parameter ϵ tends to 0 more and more bursts are present.

Chapter 2

Memristive systems

2.1 Introduction to memristive systems

A memristor is a two terminal non-linear circuit element whose existence was theorized back in 1971 by professor Leon O. Chua. It took its name from the contraction of words "memory" and "resistor". This is explicative of its main characteristic: the electrical resistance in a memristor is not constant but depends on the amount of charge that has flowed through it. The non-volatility property of memristive devices implies that after the power supply is turned off the memory state is preserved making it an interesting device for storing data.

Only in recent years, after the identification in 2008 of memristive behaviour in a nanoscale 2-terminal resistance switching system made at HP Labs by Strukov *et al.*[15], the research interest of the scientific community has renewed. Even if its experimental discovery resulted from research in the field of ReRAMs for overcoming the imminent end of Moore's Law nonetheless the memristor finds numerous other applications (e.g. memristor synapses for neuromorphic systems, memristor-based chaotic circuits).

2.1.1 The original theorization

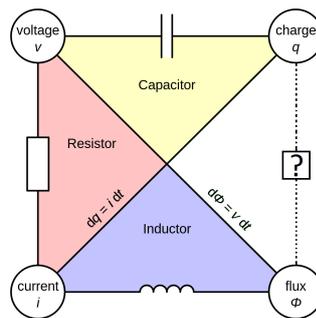


FIGURE 2.1: The four fundamental electrical variables and their connecting 2-terminal devices till 1971.

Given a two-terminal device and given the four fundamental circuit variables (flux-linkage ϕ , voltage v , charge q and current i) six 2-combinations of those are possible. Until 1971 only five out of six possible relationships had been explored (figure 2.1). Two of these were the trivial links between voltage v and flux-linkage ϕ on the one hand and charge q and current i on the other hand.

$$\begin{aligned} dq &= i dt \\ d\phi &= v dt \end{aligned}$$

The other three were the CRs for the inductor, the capacitor and the resistor.

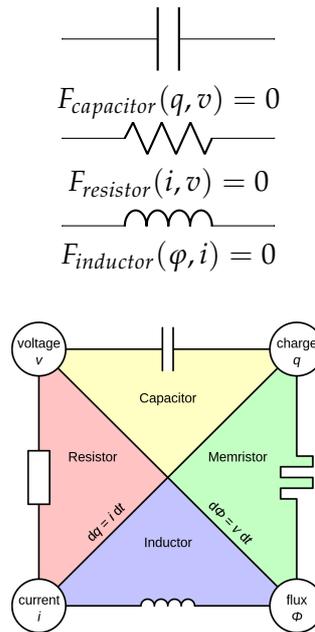
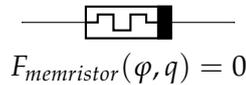


FIGURE 2.2: The four fundamental electrical variables and their connecting 2-terminal devices after 1971.

Leon O. Chua in [4] "from the logical as well as axiomatic points of view for the sake of completeness" postulated the fourth basic two-terminal missing circuit element: the memristor, completing figure 2.1 as shown in figure 2.2.



If the relation $F_{\text{memristor}}(\varphi, q) = 0$ can be expressed as a function $q \rightarrow \varphi$ ($\varphi \rightarrow q$) then the memristor is said *charge (flux) controlled*. Assuming the differentiability of either function, by deriving with respect to time we can express the voltage (current) in terms of current (voltage).

For a charge controlled memristor:

$$\frac{\partial \varphi}{\partial t} = \frac{\partial \varphi}{\partial q} \frac{\partial q}{\partial t} \Rightarrow v = R(q)i$$

For a flux controlled memristor:

$$\frac{\partial q}{\partial t} = \frac{\partial q}{\partial \varphi} \frac{\partial \varphi}{\partial t} \Rightarrow i = G(\varphi)v$$

$R(q(t))$ is called *memristance* because its unit of measurement is Ω , $G(\varphi(t))$ is called *memductance* because its unit of measurement is $S = \frac{1}{\Omega}$. Being $q(t) = \int_{-\infty}^t i(\tau) d\tau$ ($\varphi(t) = \int_{-\infty}^t v(\tau) d\tau$) then at each instant $t = t_0$ the memristor behaves like an ordinary resistor, while its resistance (conductance) depends on the whole past history of $i(t)$ ($v(t)$).

If the function $q \rightarrow \varphi$ (or $\varphi \rightarrow q$) is monotonically increasing then the memristor is a passive 2-terminal device ($P(t) = v(t)i(t) \geq 0 \forall t$).

An important remark: the memristor is intrinsically a non-linear device, if the function $q \rightarrow \varphi$ (or $\varphi \rightarrow q$) was linear then the memristor would become a linear resistor.

2.1.2 The hallmark of all memristors

As pointed out by its theorizer in recent years [3] *pinched hysteresis loops are the hallmarks of all memristors*. This *operational* definition proves useful when experimentally checking via bipolar zero-mean input signal (either voltage or current) the memristive behaviour of a device that has no known associated mathematical model.

Being a memristor is not linked to any specific physical implementation. As matter of fact experimentally pinched-loop i-v curves are found in a large variety of both biological, electrical and electrochemical systems such as arc lamps, discharge tubes, sweat ducts and K-Na ion channels. The identification of the memristive behaviour in a nanoscale 2-terminal device originated the great amount of research work in the recent years.

In the following, starting from hypothetical ideal CRs for two different memristors, the typical pinched-loop i-v curves are shown.

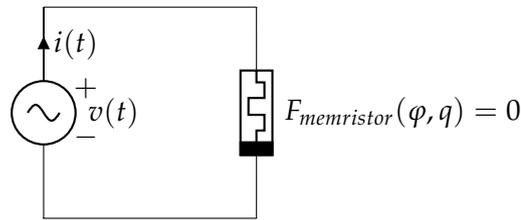


FIGURE 2.3: Pinched-loop i-v curves testing setup.

Considering a charge-controlled memristor with a CR expressible as $\varphi = e^q$, by deriving both sides of the CR with respect to time we can find the that in i-v domain the memristor is described by the following DAE.

$$\begin{cases} v = e^q i \\ \frac{dq}{dt} = i \end{cases} \quad (2.1)$$

Considering a flux-controlled memristor with a CR expressible as $q = \text{atan}(\varphi)$, by deriving both sides of the CR with respect to time we can find the that in i-v domain the memristor is described by the following DAE.

$$\begin{cases} i = \frac{1}{1+\varphi^2} v \\ \frac{d\varphi}{dt} = v \end{cases} \quad (2.2)$$

Solving the IVPs associated with 2.1 and 2.2 for $q(0) = 0$ and $\varphi(0) = 0$ respectively using, for simplicity's sake, sinusoidal waves inputs (although any bipolar zero-mean periodic wave would be a valid alternative) the pinched-loop i-v curves reported in figure 2.4 are obtained.

As numerically shown the fingerprint that distinguishes a memristor from other non-linear dynamical systems is the pinched i-v loop. Nonetheless the latter gives no relevant information about the memristor under test because each pinched-loop i-v curve is just the response to a specific input: different pinched-loops are obtained changing either the input signal frequency or the amplitude.

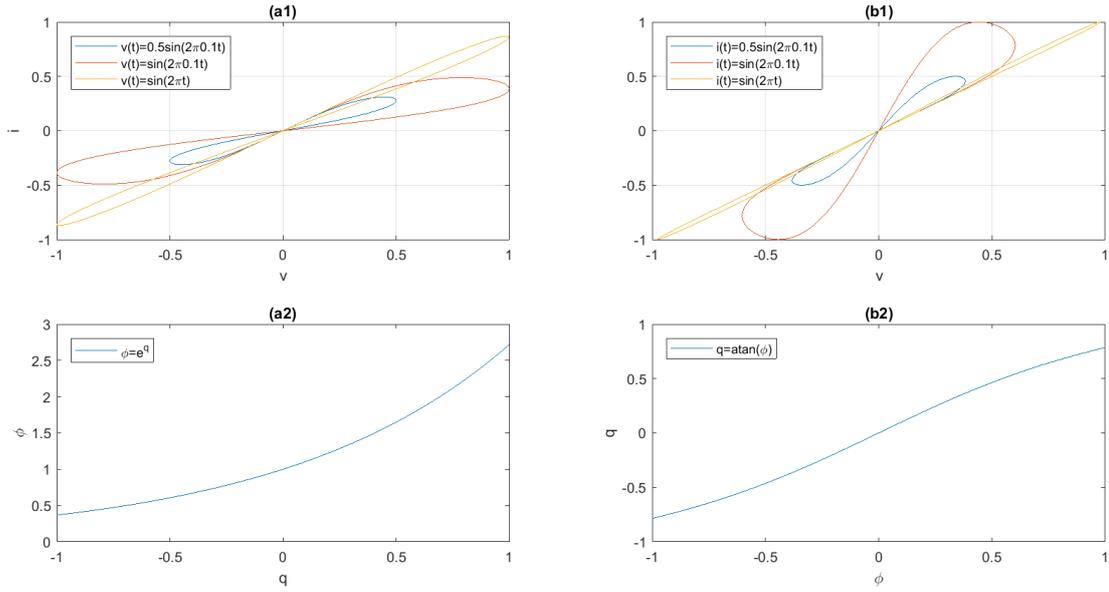


FIGURE 2.4: Pinched-loop v-i curves for a charge-controlled memristor (a1) and a flux-controlled memristor (b1), for different frequency and amplitude sinusoidal input signals. Flux-charge characteristics reported respectively in (a2) and (b2).

2.2 Taxonomy of memristive systems

The concept of memristor was already extended to that of memristor system back in 1976[5]. This extension was made in order to include in the definition devices whose state is not only defined by the history of charge and flux but also other (even non-electrical) quantities.

Recently in [7] Corinto *et al.* presented the subject of memristive systems classification in a systematic form which will be reported in the following. The novel terminology should not be thought as new types of memristor but as finer memristor subclasses.

In order to consider the state variables different from port variables $\varphi/q/v/i$ a vector $\mathbf{x} = (x_1, x_2, \dots, x_n)^T \in \mathbb{R}^n$ where each x_i (in general a non-electrical quantity) is either a state variable or a nonstate variable depending on port variables and included for convenience. The dynamic of \mathbf{x} is captured for a charge controlled memristor by the following (generally non-linear) ODE:

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(q, i, \mathbf{x})$$

For a flux controlled memristor obviously:

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\varphi, v, \mathbf{x})$$

Where $\mathbf{f} : \mathbb{R}^{n+2} \Rightarrow \mathbb{R}^n$.

In the following subsections the possible subclasses of current controlled memristors are reported. For voltage controlled memristors duality applies.

2.2.1 Current controlled extended memristor

A current controlled memristor has i as input variable and v as output variable. In general it is defined in $\varphi - q$ domain by 2.3.

$$\begin{cases} G(\varphi, q, i, \mathbf{x}) = 0 \\ \frac{d\mathbf{x}}{dt} = f(q, i, \mathbf{x}) \\ \frac{dq}{dt} = i \end{cases} \quad (2.3)$$

If a function $g : (q, i, \mathbf{x}) \rightarrow \varphi$ (called *static characteristic equation*) exists then 2.3 can be rewritten as 2.4.

$$\begin{cases} \varphi = g(q, i, \mathbf{x}) \\ \frac{d\mathbf{x}}{dt} = f(q, i, \mathbf{x}) \\ \frac{dq}{dt} = i \end{cases} \quad (2.4)$$

In the $i - v$ domain the pinched-loop constraint holds, hence the current controlled extended memristor can be describe by 2.5 if and only if 2.6 holds.

$$\begin{cases} v = R(q, i, \mathbf{x})i \\ \frac{d\mathbf{x}}{dt} = f(q, i, \mathbf{x}) \\ \frac{dq}{dt} = i \end{cases} \quad (2.5)$$

Where $R(q, i, \mathbf{x}) = \frac{dg}{dq}$ is the memristance.

$$\frac{\partial g}{\partial i} \frac{di}{dt} = - \sum_{k=1}^n \frac{\partial g}{\partial x_k} \frac{dx_k}{dt} \quad (2.6)$$

$\sum_{k=1}^n \frac{\partial g}{\partial x_k} \frac{dx_k}{dt}$ is a parasitic voltage source in series with the current controlled extended memristor while $\frac{\partial g}{\partial i} \frac{di}{dt}$ is the voltage response of a parasitic series inductance. They can either compensate each other or be present in experimental data [7]. In the latter case they should be modelled by adding them in series as in 2.7.

$$v = R(q, i, \mathbf{x})i + \frac{\partial g}{\partial i} \frac{di}{dt} + \sum_{k=1}^n \frac{\partial g}{\partial x_k} \frac{dx_k}{dt} \quad (2.7)$$

2.2.2 Current controlled generic memristor

The current controlled extended memristor is said generic memristor if its static characteristic does not depend on the port current i , in that case (under the assumption that $g : (q, \mathbf{x}) \rightarrow \varphi$ exists) 2.4 can be written as 2.8.

$$\begin{cases} \varphi = g(q, \mathbf{x}) \\ \frac{d\mathbf{x}}{dt} = f(q, i, \mathbf{x}) \\ \frac{dq}{dt} = i \end{cases} \quad (2.8)$$

In the $i - v$ domain 2.8 can be translated as 2.9.

$$\begin{cases} v = R(q, \mathbf{x})i \\ \frac{d\mathbf{x}}{dt} = f(q, i, \mathbf{x}) \\ \frac{dq}{dt} = i \end{cases} \quad (2.9)$$

Having φ no dependency on i it is obvious that a current controlled generic memristor has no inductive parasitic effect. It could still have a parasitic series voltage source.

2.2.3 Current controlled ideal memristor

The current controlled extended memristor is said ideal memristor if its static characteristic does not depend on the port current i and its state vector \mathbf{x} is of zero cardinality, in that case (under the assumption that $g : (q, \mathbf{x}) \rightarrow \varphi$ exists) 2.4 can be written just as 2.8.

$$\varphi = g(q) \quad (2.10)$$

In the $i - v$ domain 2.10 can be written as 2.11.

$$\begin{cases} v = R(q)i \\ \frac{dq}{dt} = i \end{cases} \quad (2.11)$$

The ideal memristor has no state vector and no parasitic effects at all. If for each x_i in \mathbf{x} exists a function $h_i : q \rightarrow x_i$ then the generic memristor reduces to an ideal memristor as proven in [7]. The existence of all $h_i(q)$ is not a necessary condition for a generic memristor to be ideal, it can also happen if g has no dependence on \mathbf{x} hence the state variable is *unobservable*.

2.3 Memristor as nonvolatile memory element

The application of memristors as analog nonvolatile memory devices is of extreme interest because they could mimick inside hardware implemented ANNs the role that synapses have in biological neural networks. The strength of the connections they implement can be tuned by applying electrical impulses via CMOS logic. Changing the width, the frequency or the amplitude of the pulses the memristance can be finely tuned. High amplitude pulses are usually used while programming the device and low voltage levels are applied during the memristor operation as analog circuit component.

Even if those weights correspond to the memristance while the device input is off, the memristance has no role in characterizing the device's non volatility property. The latter is described (for a generic memristor) only by mean of the state equation in 2.8. The nonvolatile memory states are linked to the stable equilibrium points x^* of the state equation describing the evolution of \mathbf{x} .

$$\frac{d\mathbf{x}}{dt} = f(q, i, \mathbf{x}) = h(q, \mathbf{x})i \quad (2.12)$$

In order to be classified as an analogue non volatile memory a generic memristor state equation should be factorizable as in 2.12. If this condition is satisfied then the generic memristor has a continuum of equilibrium states. The ideal memristor

satisfies by definition this condition. If only a finite set of equilibrium states are possible then the memristor is just a conventional digital memory.

2.4 Flux-charge analysis method

Comparing 2.11 and 2.10 one can notice that the $i - v$ representation of a memristive device introduces an additional ODE with respect to $\varphi - q$. This was recently the starting point for the development, made by Corinto and Forti [9], of a novel methodology for the analysis of nonlinear circuits containing memristors. The class of circuits that can be investigated at the moment using this technique is denoted by LM and is composed by ideal resistors, ideal inductors, ideal capacitors, ideal independent voltage sources, ideal independent current sources and ideal flux (or charge) controlled memristors.

All the circuits in LM start operating at a finite time instant $-\infty < t_0 < +\infty$, for any $t \geq t_0$ the topology of the circuit remains fixed. Given the voltage and the current on one device in LM one can define the incremental charge 2.13 and incremental flux 2.14. These defined incremental quantities reduce to the flux and charge if and only if $t_0 \rightarrow -\infty$ (e.g. a circuit with a fixed topology). The dynamics over $(-\infty, t_0)$ has to be considered to set independent initial conditions of a circuit switching topology at the finite instant t_0 .

$$q_k(t; t_0) = q_k(t) - q_k(t_0) = \int_{t_0}^t i_k(\tau) d\tau \quad (2.13)$$

$$\varphi_k(t; t_0) = \varphi_k(t) - \varphi_k(t_0) = \int_{t_0}^t v_k(\tau) d\tau \quad (2.14)$$

By integrating the Kirchoff Current Law (KCL) over the interval (t_0, t) the Kirchoff Charge Law (KqL) is obtained which states:

the algebraic sum of the incremental charge in a closed surface is zero

By integrating the Kirchoff Voltage Law (KVL) over the interval (t_0, t) the Kirchoff Flux Law ($K\varphi L$) is obtained which states:

the algebraic sum of the incremental flux through a closed surface is zero

The use of KqL and $K\varphi L$ conjugated with the CRs of circuits elements in LM expressed in terms of incremental charge and incremental flux are the pillars of the new analysis method.

2.4.1 Constitutive relations for two-terminal elements in $\varphi - q$ domain

In the following the CRs in $\varphi - q$ domain, taken from [9], are reported together with their equivalent circuit representation.

Ideal independent voltage source

By considering the ideal voltage source 2.15 by integrating between t_0 and $t \geq t_0$ its CR in $\varphi - q$ domain can be written as 2.16. Its equivalent circuit is reported in figure 2.5.

$$v(t) = e(t) \quad \forall i(t) \quad (2.15)$$

$$\varphi(t; t_0) = \varphi_e(t; t_0) = \int_{t_0}^t e(\tau) d\tau \quad \forall q_e(t; t_0) \quad (2.16)$$

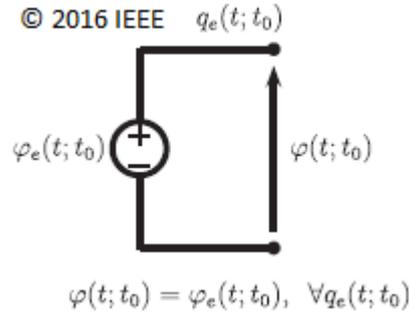


FIGURE 2.5: Ideal independent voltage source in terms of the incremental flux $\varphi_e(t; t_0)$ and charge $q_e(t; t_0)$

Ideal independent current source

By considering the ideal current source 2.17 by integrating between t_0 and $t \geq t_0$ its CR in $\varphi - q$ domain can be written as 2.18. Its equivalent circuit is reported in figure 2.6.

$$i(t) = a(t) \quad \forall v(t) \quad (2.17)$$

$$q(t; t_0) = q_e(t; t_0) = \int_{t_0}^t i(\tau) d\tau \quad \forall \varphi_e(t; t_0) \quad (2.18)$$

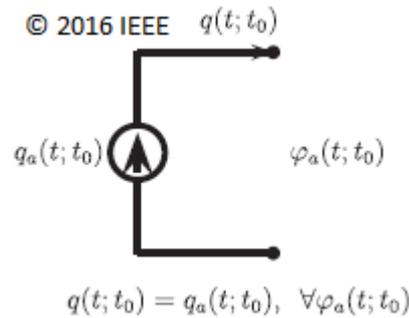


FIGURE 2.6: Ideal independent current source in terms of the incremental flux $\varphi_a(t; t_0)$ and charge $q_a(t; t_0)$

Ideal resistor

By considering the ideal resistor 2.19 by integrating between t_0 and $t \geq t_0$ its CR in $\varphi - q$ domain can be written as 2.20. Its equivalent circuit is reported in figure 2.7.

$$v_R(t) = R i_R(t) \quad (2.19)$$

$$\varphi_R(t; t_0) = R q_R(t; t_0) \quad (2.20)$$

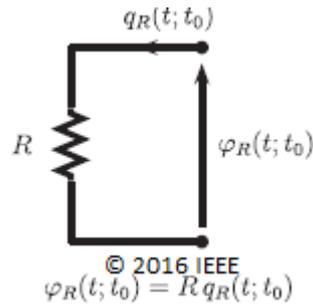


FIGURE 2.7: Ideal resistor in terms of the incremental flux $\varphi_R(t; t_0)$ and charge $q_R(t; t_0)$

Ideal capacitor

By considering the ideal capacitor 2.21 by integrating between t_0 and $t \geq t_0$ its CR in $\varphi - q$ domain can be written as 2.22. Its equivalent circuit is reported in figure 2.8.

$$i_C(t) = C \frac{dv_C(t)}{dt} \quad (2.21)$$

$$q_C(t; t_0) = C \frac{d\varphi_C(t; t_0)}{dt} - q_{C_0} \quad (2.22)$$

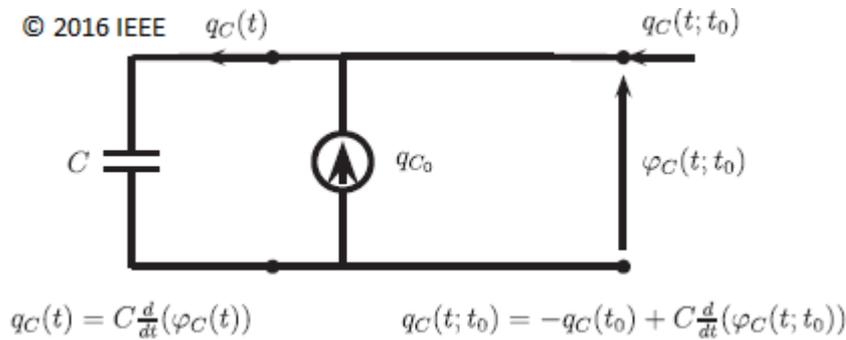


FIGURE 2.8: Ideal capacitor in terms of the incremental flux $\varphi_C(t; t_0)$ and charge $q_C(t; t_0)$

Ideal inductor

By considering the ideal inductor 2.23 by integrating between t_0 and $t \geq t_0$ its CR in $\varphi - q$ domain can be written as 2.24. Its equivalent circuit is reported in figure 2.9.

$$v_L(t) = L \frac{di_L(t)}{dt} \quad (2.23)$$

$$\varphi_L(t; t_0) = L \frac{dq_L(t; t_0)}{dt} - \varphi_{L_0} \quad (2.24)$$

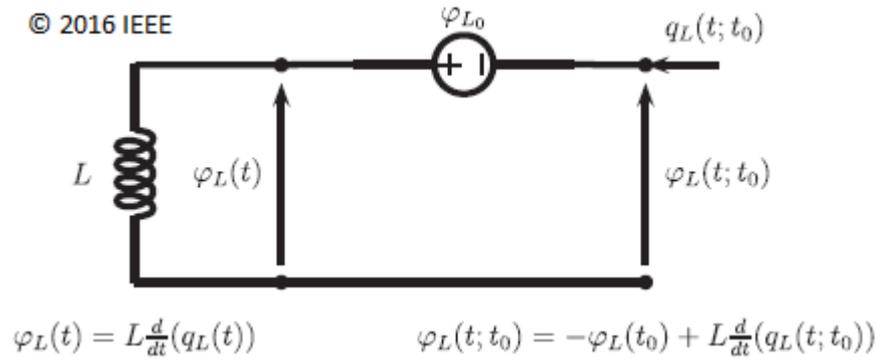


FIGURE 2.9: Ideal inductor in terms of the incremental flux $\varphi_L(t; t_0)$ and charge $q_L(t; t_0)$

Ideal flux controlled memristor

By considering the ideal flux controlled memristor 2.25 its CR is already in $\varphi - q$ domain and can be written in terms of incremental quantities and initial conditions as 2.26. Its equivalent circuit is reported in figure 2.10.

$$q_M(t) = g(\varphi_M(t)) \quad (2.25)$$

$$q_M(t; t_0) = g(\varphi_M(t; t_0) + \varphi_{M_0}) - q_{M_0} \quad (2.26)$$

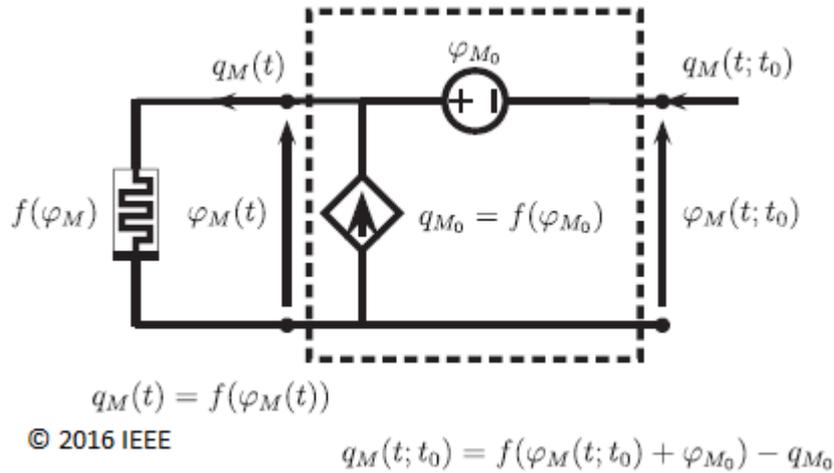


FIGURE 2.10: Ideal flux controlled memristor in terms of the incremental flux $\varphi_M(t; t_0)$ and charge $q_M(t; t_0)$

Ideal charge controlled memristor

By considering the ideal charge controlled memristor 2.27 its CR is already in $\varphi - q$ domain and can be written in terms of incremental quantities and initial conditions as 2.28. Its equivalent circuit is reported in figure 2.11.

$$\varphi_M(t) = g(q_M(t)) \quad (2.27)$$

$$\varphi_M(t; t_0) = g(q_M(t; t_0) + q_{M_0}) - \varphi_{M_0} \quad (2.28)$$

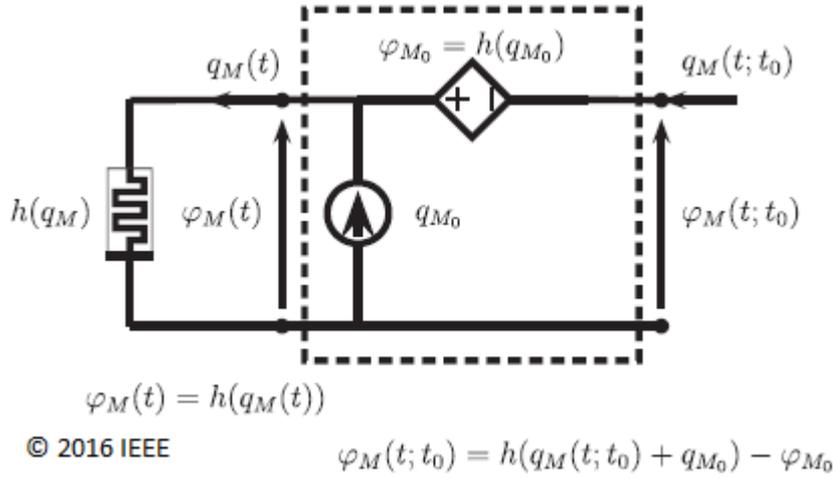


FIGURE 2.11: Ideal charge controlled memristor in terms of the incremental flux $\varphi_M(t; t_0)$ and charge $q_M(t; t_0)$

2.4.2 Incremental Kirchoff Laws

In the following the math formalizations of KqL and $K\varphi L$ as expressed in [9] are reported. Assuming to deal with l two-terminal elements in LM described in terms of *incremental charge* and *incremental flux*, and being n the number of nodes in the network, $n - 1$ cutset equations and $l - n + 1$ loop equations can be written involving the voltages 2.30 and the currents 2.29 at each element's port as the vectors $\mathbf{v}, \mathbf{i} \in \mathbb{R}^l$ multiplied by the reduced incidence matrix $\mathbf{A} \in \mathbb{R}^{(n-1) \times l}$ and reduced loop matrix $\mathbf{B} \in \mathbb{R}^{(l-n+1) \times l}$ respectively.

$$\mathbf{A}\mathbf{i}(t) = 0 \quad (2.29)$$

$$\mathbf{B}\mathbf{v}(t) = 0 \quad (2.30)$$

By integrating 2.29 and 2.30 in the interval (t_0, t) and using the definition of *incremental charge* and *incremental flux* defined in 2.13 and 2.14 the *Kirchoff Charge Law* 2.31 and *Kirchoff Flux Law* 2.32 are obtained.

$$\mathbf{A}\mathbf{q}(t; t_0) = 0 \quad (2.31)$$

$$\mathbf{B}\varphi(t; t_0) = 0 \quad (2.32)$$

This definition of charge and flux conservation laws comes useful because they are independent from initial conditions which sometimes (e.g. $q_L(t_0)$ for an inductor and $\varphi_C(t_0)$ for a capacitor) may be unavailable or difficult to obtain.

2.4.3 DAEs in the flux-charge domain

For an l elements network of devices in LM class by putting the l equations of $K\varphi L$ and KqL that set the topological constraints on $q(t; t_0)$ and $\varphi(t; t_0)$ together with the

l CRs of the elements in the network, $2l$ DAEs are obtained. The solution of those DAEs for $q(t_0; t_0) = 0$ and $\varphi(t_0; t_0) = 0$ gives the evolution of $q(t; t_0)$ and $\varphi(t; t_0)$ for any $t \geq t_0$. The solution still depends on the initial conditions for the capacitors $q_{C_{k_0}}$, the inductors $\varphi_{M_{k_0}}$ and the charge (flux) controlled memristors $q_{M_{k_0}}$ ($\varphi_{M_{k_0}}$). Those appear as constant inputs.

The method of analysis based on DAEs is fundamental in numerical simulation of circuit either linear and nonlinear.

2.4.4 State equations in flux-charge domain

Qualitative nonlinear phenomena are more easily analyzed using the state equation (SE) formulation. In this section the $\varphi - q$ domain SEs derivation is reported as obtained in [9] taking the incremental fluxes on the capacitors $\varphi_{C_0}(t; t_0)$ and the incremental charges in the inductors $q_{L_0}(t; t_0)$ as state variables.

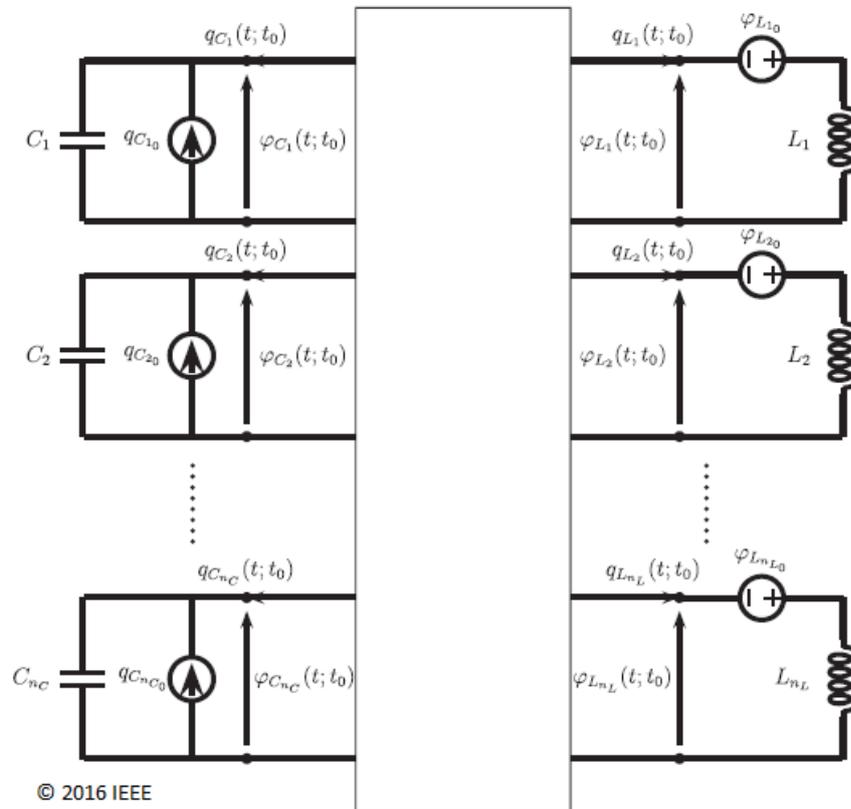


FIGURE 2.12: Representation of a circuit made of components in LM class in order to derive the state equations in terms of the incremental flux $\varphi_C(t; t_0)$ and incremental charge $q_L(t; t_0)$

Considering a circuit in LM represented as in figure 2.12 where all capacitors and inductors are connected to an algebraic non linear $(n_C + n_L)$ -port containing ideal memristors (both flux and charge controlled), ideal resistors, ideal current and voltage sources. All the elements are described by CRs in $\varphi - q$ domain. The SEs for the circuit can be written as in 2.33.

$$\begin{cases} \mathbf{C} \frac{d}{dt}(\boldsymbol{\varphi}(t; t_0)) = \mathbf{q}_{C_0} - h_a(\boldsymbol{\varphi}_C(t; t_0), \mathbf{q}_L(t; t_0), \boldsymbol{\varphi}_E(t), \mathbf{q}_A(t), \boldsymbol{\varphi}_{M_0}, \mathbf{q}_{M_0}) \\ \mathbf{L} \frac{d}{dt}(\mathbf{q}(t; t_0)) = \boldsymbol{\varphi}_{C_0} - h_b(\boldsymbol{\varphi}_C(t; t_0), \mathbf{q}_L(t; t_0), \boldsymbol{\varphi}_E(t), \mathbf{q}_A(t), \boldsymbol{\varphi}_{M_0}, \mathbf{q}_{M_0}) \\ \boldsymbol{\varphi}_C(t_0; t_0) = 0 \\ \mathbf{q}_L(t_0; t_0) = 0 \end{cases} \quad (2.33)$$

Where $\boldsymbol{\varphi}_C(t; t_0)$ is the vector of state associated with the fluxes on capacitors, $\mathbf{C} = \text{diag}[C_1, \dots, C_{n_C}]$, $\mathbf{q}_L(t; t_0)$ is the vector of state associated with the charges inside inductors, $\mathbf{L} = \text{diag}[L_1, \dots, L_{n_L}]$, \mathbf{q}_{C_0} is the vector of initial charges in the capacitors, $\boldsymbol{\varphi}_{L_0}$ is the vector of initial fluxes on the inductors, $\boldsymbol{\varphi}_E$ is the vector of independent flux sources, \mathbf{q}_A is the vector of independent charge sources, $\boldsymbol{\varphi}_{M_0}$ is the vector of initial fluxes on flux controlled ideal memristors and \mathbf{q}_{M_0} is the vector of initial charges in charge controlled ideal memristors.

Criteria to prove the existence of SEs 2.33 and the hybrid representation 2.12 can be derived according to the rigorous approach presented in[2].

2.5 Traditional voltage-current analysis method

The traditional formulation of circuit equations for a circuit in class *LM* can be derived either by differentiation of the DAEs (or SEs) in the $\varphi - q$ domain or by using the KVLs, the KCLs and the CRs of each component in the $i - v$ domain.

By observing the translation of the ideal memristor CR from $\varphi - q$ 2.10 domain to $v - i$ 2.11 domain, it is clear that each additional memristor introduces an associate ODE increasing the computational effort for solving the IVP. This is because while the memristor is a dynamical element in $v - i$ domain it plays in $\varphi - q$ domain the same role of a nonlinear resistor in $v - i$ domain. Still it has a constant flux and a constant charge generators that hold the past history of its voltage and current.

Voltages and currents in the circuit can be easily found by deriving the incremental fluxes and incremental charges after solving the IVP $\varphi - q$ domain. The flux (or charge) on the ideal flux (charge) controlled memristor is obtained easily by definition 2.34.

$$\varphi_M(t) = \varphi_M(t; t_0) + \varphi_{M_0} \quad (2.34)$$

2.6 Bifurcation phenomena in memristive circuits

The use of memristors, devices that are by nature programmable, allows to design circuits in which bifurcation phenomena can take place without changing the circuit parameters but just the initial conditions. Practically speaking the memristor behaves like an electronically tunable potentiometer, this opens the door to electronic programming of the analog circuit parameters.

2.6.1 Invariant manifolds

An *invariant manifold* is a region of the phase-space of the SE in the $(i - v)$ domain on which the evolution of the state variables takes place. Their geometric structure is essential to unfold the bifurcation phenomena due to initial conditions. There are ∞^1 non-intersecting manifolds spanning the whole phase space of the SE in $(i - v)$ domain, on each of these manifolds there is a foliation of the SEs.

2.6.2 Fixed-invariant manifold

If the manifold is fixed (e.g. the set of initial conditions satisfies a specific relation) then bifurcations of the ODEs might occur only if the circuit parameters are changed (e.g. varying a potentiometer).

2.6.3 Bifurcations without parameters

How period-doubling and Hopf bifurcations are induced by varying the initial conditions for the dynamic elements in the $(i - v)$ domain can be made clear by using the *Flux Charge Analysis Method* (FCAM) as shown in [8]. If the circuit parameters are fixed and bifurcation phenomena can be induced by varying the initial condition for the state variables in $(v - i)$ domain then the bifurcation is called *without parameters*. The advantage that FCAM offers is to analitically study those phenomena more easily using a lower order set of ODEs with smoother nonlinearities.

Chapter 3

Memristive Chua circuit

3.1 Memristive Chua circuit

In figure 3.1 is reported the circuit whose study is the main focus of this chapter. It is a simple circuit containing two capacitors, one non-ideal inductor, one resistor and one memristor.

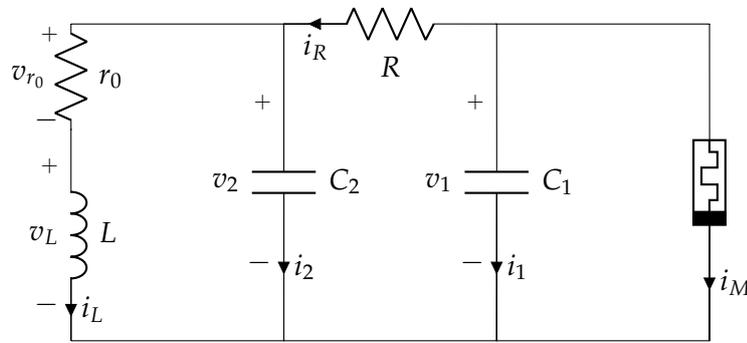


FIGURE 3.1: Memristive Chua circuit topology.

Considering an ideal locally active flux-controlled memristor whose CR in $(\varphi - q)$ domain is 3.1.

$$q_M = f(\varphi_M) = -a\varphi_M + b\varphi_M^3 \quad (3.1)$$

3.1.1 Analysis in (v-i) domain

The CR 3.1 can also be written in $(v - i)$ domain as 3.2.

$$\begin{cases} i_M = (-a + 3b\varphi_M^2)v_M \\ \frac{d\varphi_M}{dt} = v_M \end{cases} \quad (3.2)$$

The SEs describing the system in $(i - v)$ domain are the ones in 3.3.

$$\begin{cases} \frac{dv_1}{dt} = \frac{1}{C_1} \left[\frac{v_2 - v_1}{R} - (-a + 3b\varphi_M^2)v_1 \right] \\ \frac{dv_2}{dt} = \frac{1}{C_2} \left(\frac{v_1 - v_2}{R} - i_L \right) \\ \frac{di_L}{dt} = \frac{1}{L} (v_2 - i_L r_0) \\ \frac{d\varphi_M}{dt} = v_1 \end{cases} \quad (3.3)$$

The circuit in figure 3.1 has a line equilibrium point which is dependent on the memristor's initial condition. Since one of the eigenvalues of 3.3 is always equal to zero, the stability of the line equilibrium point cannot be explicitly distinguished just

with respect to the nonzero eigenvalues. In $(i - v)$ domain only numerical simulations are possible to investigate the phenomenon of multistability.

3.1.2 Analysis in Flux-Charge domain

Using the $\varphi - q$ CR of elements in class LM the circuit in figure 3.1 can be represented as in figure 3.2.

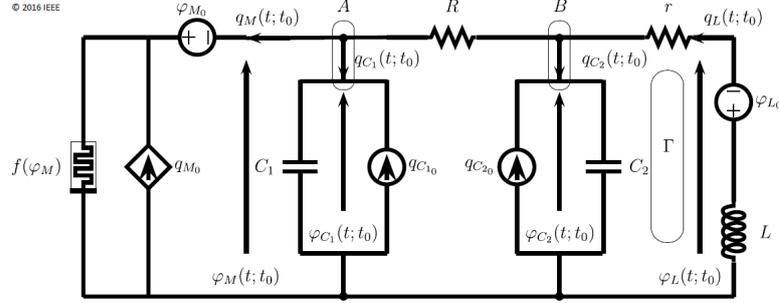


FIGURE 3.2: Memristive Chua circuit $(\varphi - q)$ domain equivalent.

The SEs for the circuit 3.2 can be written in terms of the state variables $\varphi_{C_1}(t; t_0)$, $\varphi_{C_2}(t; t_0)$ and $q_L(t; t_0)$ as 3.4.

$$\begin{cases} C_1 \frac{d\varphi_{C_1}(t; t_0)}{dt} = \frac{1}{R}(\varphi_{C_2}(t; t_0) - \varphi_{C_1}(t; t_0)) - f(\varphi_{C_1}(t; t_0) + \varphi_{M_0}) + f(\varphi_{M_0}) + q_{C_{1_0}} \\ C_2 \frac{d\varphi_{C_2}(t; t_0)}{dt} = -\frac{1}{R}(\varphi_{C_2}(t; t_0) - \varphi_{C_1}(t; t_0)) + q_L(t; t_0) + q_{C_{2_0}} \\ L \frac{dq_L(t; t_0)}{dt} = -rq_L(t; t_0) - \varphi_{C_2}(t; t_0) - \varphi_{L_0} \\ \varphi_{C_1}(t_0; t_0) = 0 \\ \varphi_{C_2}(t_0; t_0) = 0 \\ q_L(t_0; t_0) = 0 \end{cases} \quad (3.4)$$

Assuming the normalization values in table 3.1.

R_0	C_0	L_0	T_0	V_0	I_0	Q_0	φ_0
$1k\omega$	$1nF$	$1mH$	$1\mu s$	$1V$	$1mA$	$1nAs$	$1\mu Vs$

TABLE 3.1: The normalization values for resistance (R_0), for capacitance (C_0), for inductance (L_0), for voltages (V_0), for currents (I_0), for charges (Q_0), for time (T_0) and for fluxes (φ_0)

Using the parameters in 3.5.

$$\begin{cases} \tau = \frac{t}{RC_2} \\ \alpha = \frac{C_2}{C_1} \\ \beta = \frac{R^2 C_2}{L} \\ \gamma = \frac{Rr_0 C_2}{L} \end{cases} \quad (3.5)$$

The equations 3.4 can be rewritten as 3.6.

$$\begin{cases} \frac{d\varphi_{C_1}(\tau; \tau_0)}{d\tau} = \alpha[\varphi_{C_2}(\tau; \tau_0) - \varphi_{C_1}(\tau; \tau_0) - Rf(\varphi_{C_1}(\tau; \tau_0) + \varphi_{M_0}) + Rf(\varphi_{M_0}) + Rq_{C_{1_0}}] \\ \frac{d\varphi_{C_2}(\tau; \tau_0)}{d\tau} = -(\varphi_{C_2}(\tau; \tau_0) - \varphi_{C_1}(\tau; \tau_0)) + Rq_L(\tau; \tau_0) + Rq_{C_{2_0}} \\ \frac{d(Rq_L(\tau; \tau_0))}{d\tau} = -\gamma(Rq_L(\tau; \tau_0)) - \beta\varphi_{C_2}(\tau; \tau_0) - \beta\varphi_{L_0} \end{cases} \quad (3.6)$$

By denoting the variable τ as t and applying to 3.6 the change of variables 3.7 the third-order system 3.8 is obtained.

$$\begin{cases} x(t) = \varphi_{C_1}(t; t_0) + \varphi_{M_0} \\ y(t) = \varphi_{C_2}(t; t_0) + \frac{\gamma}{\beta+\gamma}(\varphi_{M_0} - Rq_{C_{2_0}}) + \frac{\beta}{\beta+\gamma}\varphi_{L_0} \\ z(t) = Rq_L(t; t_0) + \frac{\beta}{\beta+\gamma}(\varphi_{L_0} - \varphi_{M_0} + Rq_{C_{2_0}}) \end{cases} \quad (3.7)$$

$$\begin{cases} \frac{dx(t)}{dt} = \alpha[-x(t) + y(t) - n(x(t)) + X_0] \\ \frac{dy(t)}{dt} = x(t) - y(t) + z(t) \\ \frac{dz(t)}{dt} = -\beta y(t) - \gamma z(t) \\ x(t_0) = \varphi_{M_0} \\ y(t_0) = \frac{\gamma}{\beta+\gamma}(\varphi_{M_0} - Rq_{C_{2_0}}) + \frac{\beta}{\beta+\gamma}\varphi_{L_0} \\ z(t_0) = \frac{\beta}{\beta+\gamma}(\varphi_{L_0} - \varphi_{M_0} + Rq_{C_{2_0}}) \end{cases} \quad (3.8)$$

Where $n(x(t))$ is 3.9.

$$n(x(t)) = Rf(x(t)) = -Rax(t) + Rbx^3(t) = -m_0x(t) + m_1x^3(t) \quad (3.9)$$

And X_0 is 3.10.

$$X_0 = \frac{\beta}{\gamma + \beta}(\varphi_{M_0} - Li_{L_0}) + \frac{\gamma}{\gamma + \beta}RC_2v_{C_{2_0}} + n(\varphi_{M_0}) + RC_1v_{C_{1_0}} \quad (3.10)$$

Once the IVP for 3.8 is solved, the state variables in $(i - v)$ domain are easily derived by

$$\begin{cases} v_{C_1}(t) = \alpha[-x(t) + y(t) - n(x(t)) + X_0] \\ v_{C_2}(t) = x(t) - y(t) + z(t) \\ i_L(t) = -\frac{1}{R}(\beta y(t) + \gamma z(t)) \\ \varphi_M(t) = x(t) \end{cases} \quad (3.11)$$

All the trajectories of 3.8 is contained in one among the ∞^1 positively-invariant manifolds identified, as shown below, each one by a different value of $X_0 \in \mathbb{R}$.

$$M(X_0) = \{(v_{C_1}(t), v_{C_2}(t), i_L(t), \varphi_M(t)) \in \mathbb{R}^4 :$$

$$\frac{\beta}{\gamma + \beta}(\varphi_M(t) - Li_L(t)) + \frac{\gamma}{\beta + \gamma}RC_2v_{C_2}(t) + n(\varphi_M(t)) + RC_1v_{C_1}(t) = X_0\}$$

Assuming $\gamma = 0$ and $R = 1$ for simplicity's sake the SEs 3.6 have the simplified expression in 3.12.

$$\begin{cases} \frac{dx(t)}{dt} = \alpha[y(t) - x(t) - n(x(t)) + X_0] \\ \frac{dy(t)}{dt} = x(t) - y(t) + z(t) \\ \frac{dz(t)}{dt} = -\beta y(t) \\ x(t_0) = \varphi_{M_0} \\ y(t_0) = \varphi_{L_0} \\ z(t_0) = -\varphi_{M_0} + \varphi_{L_0} + q_{C_{2_0}} \end{cases} \quad (3.12)$$

The simplified expression of X_0 takes the form reported in 3.13.

$$X_0 = \varphi_{M_0} - Li_{L_0} + n(\varphi_{M_0}) + C_1 v_{C_{1_0}} \quad (3.13)$$

The ∞^1 invariant manifolds take the form.

$$\begin{aligned} M(X_0) = \{ & (v_{C_1}(t), v_{C_2}(t), i_L(t), q_M(t)) \in \mathbb{R}^4 : \\ & \varphi_M(t) - Li_L(t) + n(\varphi_M(t)) + C_1 v_{C_1}(t) = X_0 \} \end{aligned}$$

The equilibrium points of 3.12 take the form $P = (x^*, 0, -x^*)$ where x^* can be found by solving the algebraic equation 3.14.

$$x^* + n(x^*) = m_1 (x^*)^3 - (m_0 - 1)x^* = X_0 \quad (3.14)$$

Given $M^* = \frac{2}{3}(m_0 - 1)\sqrt{\frac{m_0 - 1}{3m_1}}$ the following cases for equilibrium points can be derived:

- If $|X_0| < M^*$ three equilibrium points exist
- If $|X_0| = M^*$ two equilibrium points exist
- If $|X_0| > M^*$ only one equilibrium point exists

Two alternatives exist for bifurcations to occur:

1. Having β , m_0 and m_1 fixed. By fixing the initial conditions $(v_{C_{1_0}}, v_{C_{2_0}}, i_{L_0}, \varphi_{M_0})$ the constant X_0 identifying the manifold $M(X_0)$ is fixed and bifurcation take place on that manifold by changing the parameter α .
2. Having β , α , m_0 and m_1 fixed. By varying the initial condition, in such a way that X_0 changes, the bifurcation phenomena of the ODEs take place without parameters change.

Assuming $\beta = 15$, $m_0 = \frac{8}{7}$, $m_1 = \frac{4}{63}$ the two possibilities are explored in the following.

In figure 3.3 and figure 3.4 are reported the phase portraits (both in $(\varphi - q)$ and $i - v$ domains) of the circuit on a fixed manifold $M(X_0 = 0.0347)$. To be noted how changing the bifurcation parameter α in range $[8.6, 8.9]$ a series of Hopf bifurcations occurs followed by a period doubling cascade which leads soon ($\alpha \approx 9.2$) to chaotic dynamic typical of a Rössler attractor. For $\alpha \approx 9.4$ a window of periodicity is visible, increasing further the single scroll attractor turns into a double scroll attractor.

By setting $\alpha = 9.5$ and varying X_0 (by changing φ_{M_0}) bifurcation phenomena can be obtained without tuning the parameters value. This is shown in figure 3.5 and figure 3.6.

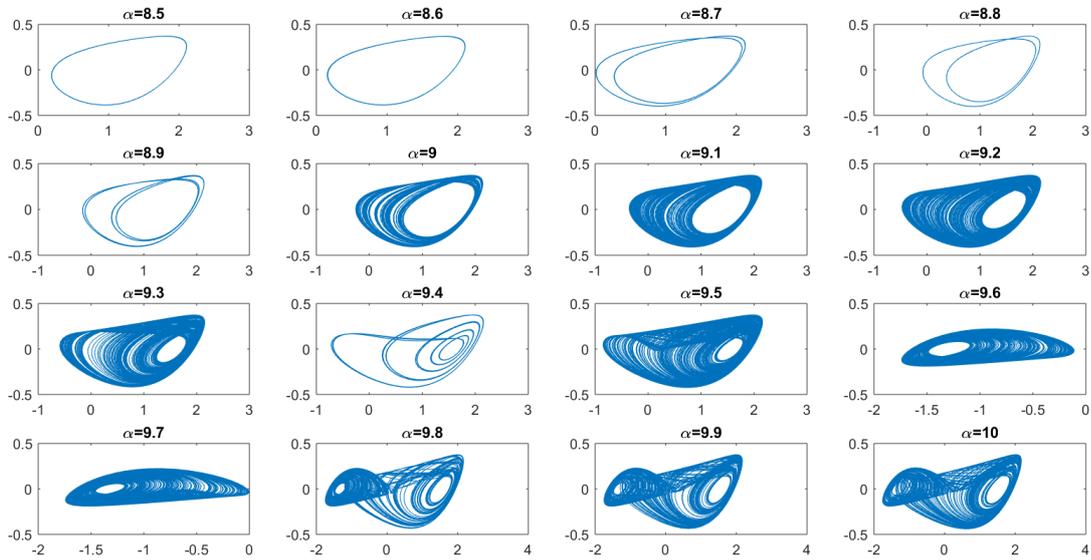


FIGURE 3.3: Phase portraits projection on $x - y$ plane in $\varphi - q$ domain showing bifurcations on a fixed manifold $M(X_0 = 0.0347)$

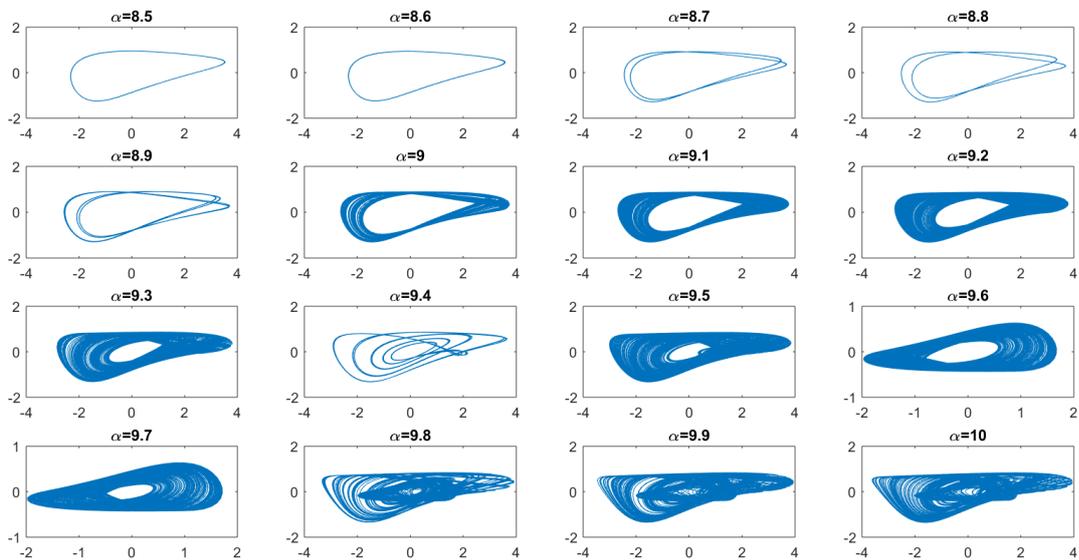


FIGURE 3.4: Phase portraits projection on $v_{C_1} - v_{C_2}$ plane in $v - i$ domain showing bifurcations on a fixed manifold $M(X_0 = 0.0347)$

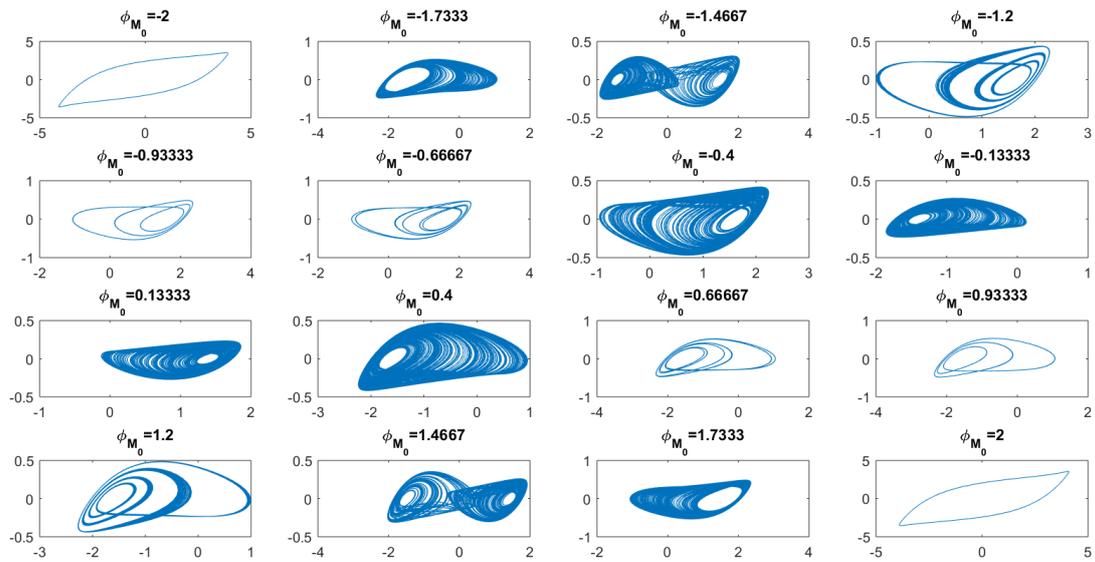


FIGURE 3.5: Phase portraits projection on $x - y$ plane in $\varphi - q$ domain showing bifurcations without parameters

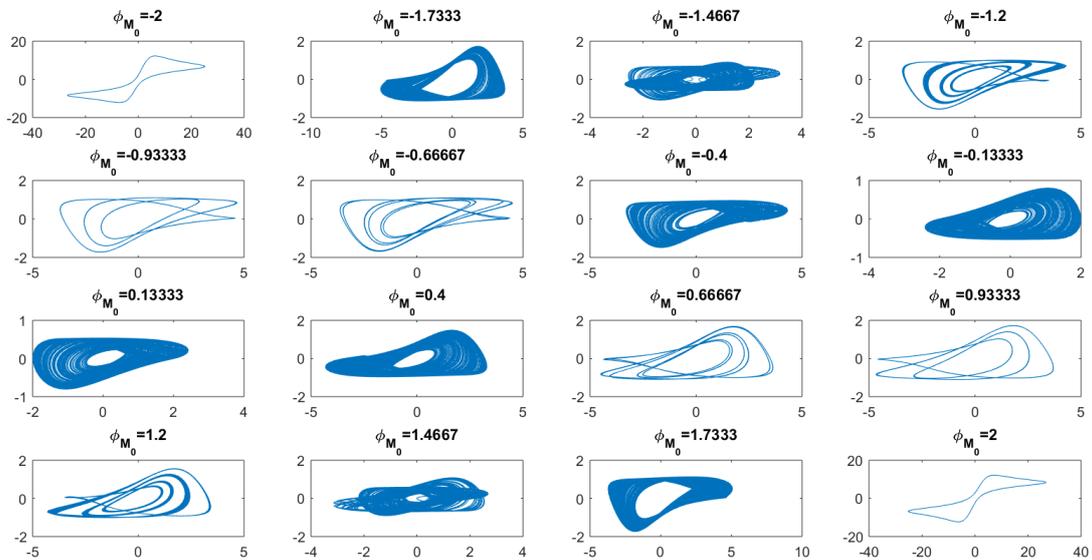


FIGURE 3.6: Phase portraits projection on $v_{C_1} - v_{C_2}$ plane in $v - i$ domain showing bifurcations without parameters

3.2 Circuit implementation

Taking inspiration from [18] a possible implementation of the previously described circuit is proposed in the following section.

3.2.1 Inductance emulation

In order to have really small series resistance ($\gamma \approx 0$) and a precise inductance value the use of an inductance emulator is mandatory. The choice is the Antoniou's circuit as described in [17]. Its topology is reported in figure 3.7.

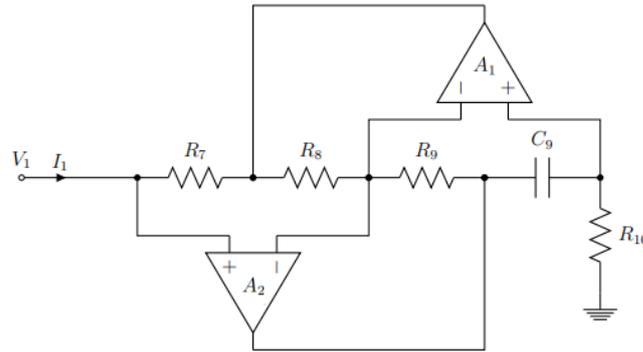


FIGURE 3.7: Antoniou's inductance emulator. [17]

The emulated inductance value 3.15 can be tuned by changing different resistors value.

$$L = \frac{C_9 R_7 R_9 R_{10}}{R_8} \quad (3.15)$$

3.2.2 Ideal memristor emulation

The ideal memristor emulator topology was taken from [18] and it is reported in figure 3.8. It makes use of an infinite gain integrator, a voltage follower, two analog multipliers and a negative resistance.

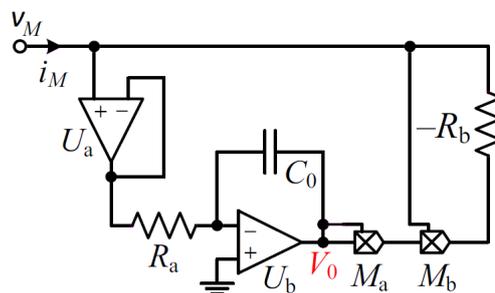


FIGURE 3.8: Ideal memristor emulator topology.[18]

It is easy to derive the $i_M - v_M$ relation 3.16 and the associated ODE for the circuit in figure 3.8. Where $g = g_a * g_b$ is the total attenuation introduced by the analog multipliers. The negative resistance can be implemented as shown in the following subsection.

$$\begin{cases} i_M(t) = (-\frac{1}{R_B} + \frac{g}{R_B} V_0^2(t))v_M(t) \\ \frac{dV_0(t)}{dt} = -\frac{1}{R_a C_0} v_M(t) \end{cases} \quad (3.16)$$

3.2.3 Negative resistance

The negative resistance can be implemented by a negative impedance converter (NIC) as shown in figure 3.9.

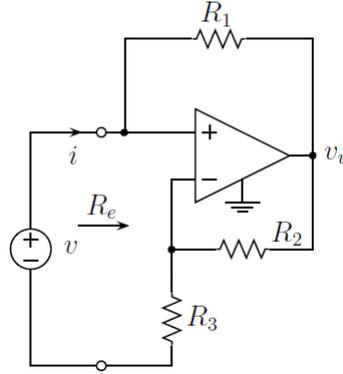


FIGURE 3.9: NIC implementing a negative resistance $R_e = -\frac{R_1 R_3}{R_2}$

The circuit shows a negative input resistance between its two input terminals for the single ended voltages on each terminal under the power supply level V_{DD} and a differential input level $|v| < \frac{R_3}{R_2+R_3} V_{DD} = B_p$. Out of the operating range the circuit behaves like a resistor R_1 in series to a voltage generator $\pm V_{DD}$. Thus $R_3 \gg R_2$ must hold in order to have a large operating range B_p .

3.2.4 Current-Voltage state equations

Using the memristor emulator in subsection 3.2.2, the inductance emulator in subsection 3.2.1 and the negative resistor in subsection 3.2.3 the SEs for the memristive Chua circuit implemented can be written in $(v - i)$ domain as 3.17.

$$\begin{cases} \frac{dv_{C_1}}{dt} = \frac{1}{C_1} \left[\frac{V_{C_2} - V_{C_1}}{R} - \left(-\frac{1}{R_B} + \frac{g}{R_B} V_0^2 \right) v_{C_1} \right] \\ \frac{dv_{C_2}}{dt} = \frac{1}{C_2} \left(\frac{v_{C_1} - v_{C_2}}{R} - i_L \right) \\ \frac{di_L}{dt} = \frac{1}{L} v_{C_2} \\ \frac{dV_0}{dt} = -\frac{1}{R_a C_0} v_{C_1} \end{cases} \quad (3.17)$$

The SEs 3.17 has a line equilibrium point which is dependent on the ideal memristor emulator's initial condition. Since one of the eigenvalues of 3.3 is always equal to zero, the stability of the line equilibrium point cannot be explicitly distinguished just with respect to the nonzero eigenvalues. In $(i - v)$ domain only numerical simulations are possible to investigate the phenomenon of multistability.

3.2.5 Flux-Charge state equations

In order to write the SEs in $(\varphi - q)$ domain the ideal memristor emulator CR the $i_M - v_M$ characteristic 3.16 should be integrated over time to find the incremental

charge $q_M(t; t_0)$ 3.18 as function of the incremental flux $\varphi_M(t; t_0)$ and internal initial condition $V_0(t_0)$.

$$q_M(t; t_0) = \int_{t_0}^t i_M(\tau) d\tau = \int_{t_0}^t \frac{1}{R_B} (-1 + gV_0^2(\tau)) v_M(\tau) d\tau \quad (3.18)$$

By the substitution of the ODE 3.16 in 3.18 the 3.19 is obtained.

$$q_M(t; t_0) = \frac{1}{R_B} \int_{V_0(t_0)}^{V_0(t)} (1 - gV_0^2) R_a C_0 dV_0 \quad (3.19)$$

Omitting the intermediate steps the CR in $(\varphi - q)$ domain is 3.20.

$$q_M(t; t_0) = \frac{1}{R_B} (-1 + gV_0^2(t_0)) \varphi_M(t; t_0) - \frac{g}{R_B R_a C_0} V_0(t_0) \varphi_M^2(t; t_0) + \frac{g}{3R_B (R_a C_0)^2} \varphi_M^3(t; t_0) \quad (3.20)$$

In the memristive Chua circuit of subsection 3.1 $\varphi_M(t; t_0) = \varphi_{C_1}(t; t_0)$. The SEs are written in $\varphi - q$ domain as 3.21.

$$\begin{cases} \frac{d\varphi_{C_1}(t; t_0)}{dt} = \frac{1}{C_1} \left(\frac{\varphi_{C_2}(t; t_0) - \varphi_{C_1}(t; t_0)}{R} - q_M(t; t_0) \right) + v_{C_1}(t_0) \\ \frac{d\varphi_{C_2}(t; t_0)}{dt} = \frac{1}{C_2} \left(\frac{\varphi_{C_1}(t; t_0) - \varphi_{C_2}(t; t_0)}{R} - q_L(t; t_0) \right) + v_{C_2}(t_0) \\ \frac{dq_L(t; t_0)}{dt} = \frac{1}{L} \varphi_2(t; t_0) + i_L(0) \end{cases} \quad (3.21)$$

Using the normalizations in 3.22.

$$\begin{cases} \tau = \frac{t}{RC_2} \\ x(t) = \frac{\varphi_{C_1}(t; t_0)}{B_P RC_2} \\ y(t) = \frac{\varphi_{C_2}(t; t_0)}{B_P RC_2} \\ z(t) = \frac{q_L(t; t_0)}{B_P C_2} \\ \eta_0 = \frac{V_0(t_0)}{B_P} \\ \eta_1 = \frac{v_{C_1}(t_0)}{B_P} \\ \eta_2 = \frac{v_{C_2}(t_0)}{B_P} \\ \eta_3 = \frac{i_L(t_0)R}{B_P} \end{cases} \quad (3.22)$$

And defining the parameters in 3.23.

$$\begin{cases} \alpha = \frac{C_2}{C_1} \\ \beta = \frac{R^2 C_2}{L} \\ a_4 = \frac{R}{R_B} \\ a_3 = a_4 g B_P^2 \\ a_2 = a_3 \frac{RC_2}{R_a C_0} \\ a_1 = \frac{1}{3} \frac{RC_2}{R_a C_0} a_2 \end{cases} \quad (3.23)$$

The SEs in 3.21 can be writtend in adimensional form as 3.24 with $h(x) = -a_1x^3 + a_2\eta_0x^2 + (a_4 - a_3\eta_0^2)x$.

$$\begin{cases} \frac{dx(t)}{dt} = \eta_1 + \alpha[y - x + h(x)] \\ \frac{dy(t)}{dt} = \eta_2 + x - y - z \\ \frac{dz(t)}{dt} = \eta_3 + \beta y \end{cases} \quad (3.24)$$

The equilibrium points are readily derived from 3.24 and are in the form 3.25.

$$P = (x^*, -\frac{\eta_3}{\beta}, x^* + \frac{\eta_3}{\beta} + \eta_2) \quad (3.25)$$

Where the x^* can be found as roots of the cubic equation 3.26.

$$-a_1x^3 + a_2\eta_0x^2 + (a_4 - a_3\eta_0^2 - 1)x + (\frac{\eta_1}{\alpha} - \frac{\eta_3}{\beta}) = 0 \quad (3.26)$$

Defining $a = -a_1$, $b = a_2\eta_0$, $c = (a_4 - a_3\eta_0^2 - 1)$ and $d = \frac{\eta_1}{\alpha} - \frac{\eta_3}{\beta}$ and the quantities in 3.27 the general formula for the roots is reported in 3.28.

$$\begin{cases} \Delta_0 = b^2 - 3ac \\ \Delta_1 = 2b^3 - 9abc + 27a^2d \\ C = \sqrt[3]{\frac{\Delta_1 \pm \sqrt{\Delta_1^2 - 4\Delta_0^3}}{2}} \\ \zeta = -\frac{1}{2} + \frac{1}{2}\sqrt{3}i \end{cases} \quad (3.27)$$

$$x_k^* = -\frac{1}{3a}(b + \zeta^k C + \frac{\Delta_0}{\zeta^k C}) \quad k \in \{0, 1, 2\} \quad (3.28)$$

The Jacobian matrix is 3.29.

$$J(x^*) = \begin{pmatrix} \alpha[-1 + h'(x^*)] & \alpha & 0 \\ 1 & -1 & -1 \\ 0 & \beta & 0 \end{pmatrix} \quad (3.29)$$

Where $h'(x) = -3a_1x^2 + 2a_2\eta_0x + (a_4 - a_3\eta_0^2)$. By evaluating the sign of the eigenvalues the stability of the equilibrium points can readily be determined. In $i - v$ domain the presence of the zero eigenvalue does not allow the study of equilibrium points. In the $\varphi - q$ domain the line equilibrium point is converted into some determined equilibrium points. The locations and stabilities of those can be determined as function of the initial conditions $\eta_0, \eta_1, \eta_2, \eta_3$ that identify the manifold.

Numerical simulations

By setting $a_1 = \frac{18}{10}$, $a_2 = \frac{36}{10}$, $a_3 = \frac{24}{10}$, $a_4 = \frac{11}{10}$, $\alpha = 9.6$, $\beta = 15$ and assuming $\gamma = 0$ the numerical simulations are carried out.

In the upper subplots of figures 3.10 and 3.13 are reported the equilibrium points x^* (in 3.25 the whole 3D position) of 3.24 as function of η_0 for $\eta_1 = -0.0173$ and $\eta_1 = 0.0345$ respectively. For the same values of η_1 in the lower subplots of the same figures the maximum Lyapunov exponent (computed using [11]) is reported as function of η_0 .

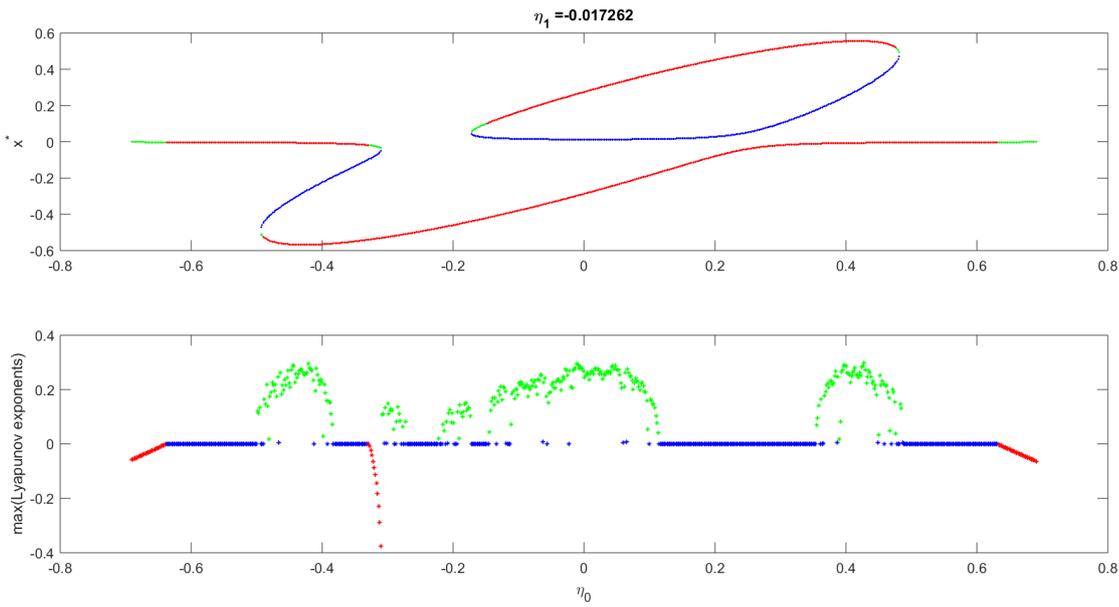


FIGURE 3.10: Equilibrium points and maximum Lyapunov exponent of 3.24 for $\eta_1 = -0.0173$ and varying η_0 . Stable equilibrium points in green, unstable equilibrium point in red (2 positive eigenvalues) or blue (only 1 positive eigenvalue). Maximum Lyapunov exponent in red if negative, in green if positive and in blue if ≈ 0 (periodic regime).

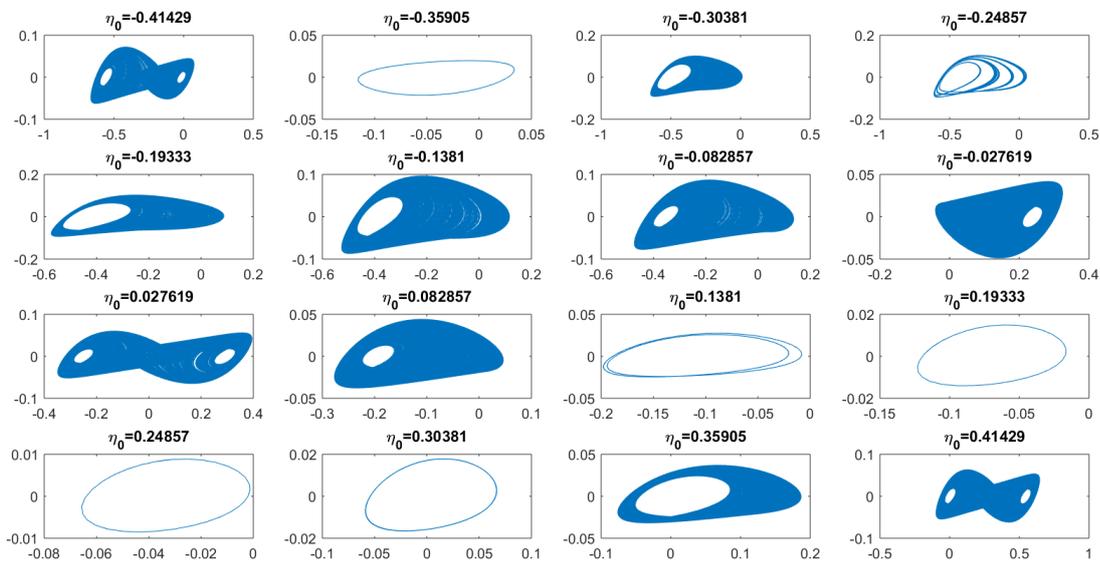


FIGURE 3.11: Phase portraits' projections on $x - y$ plane of 3.24 in $(\varphi - q)$ domain for $\eta_1 = -0.0173$ and varying η_0 .

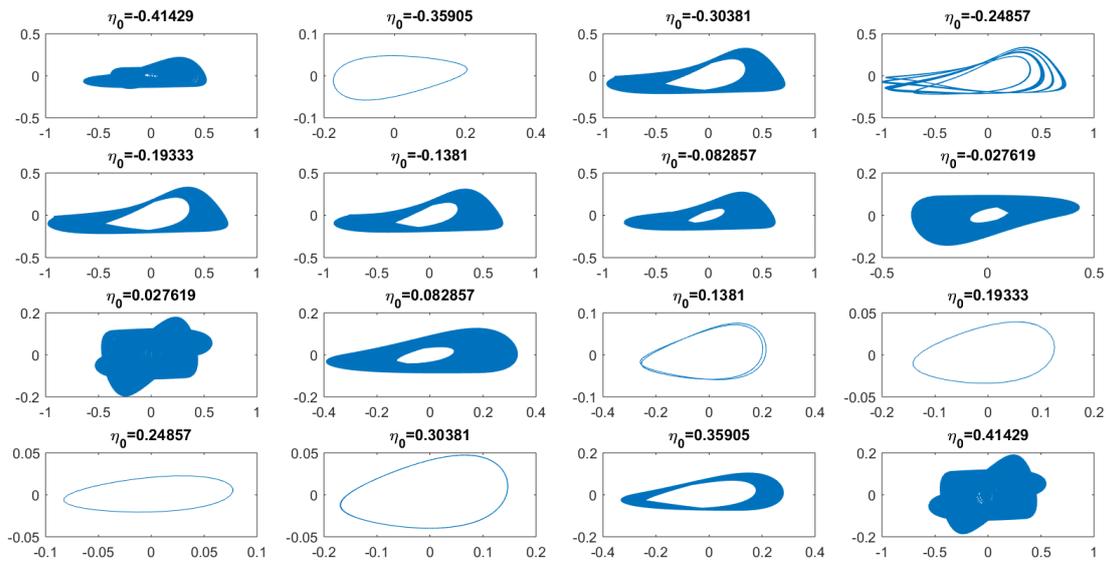


FIGURE 3.12: Phase portraits' projections on $v_{C_1} - v_{C_2}$ plane of 3.24 in $(i - v)$ domain for $\eta_1 = -0.0173$ and varying η_0 .

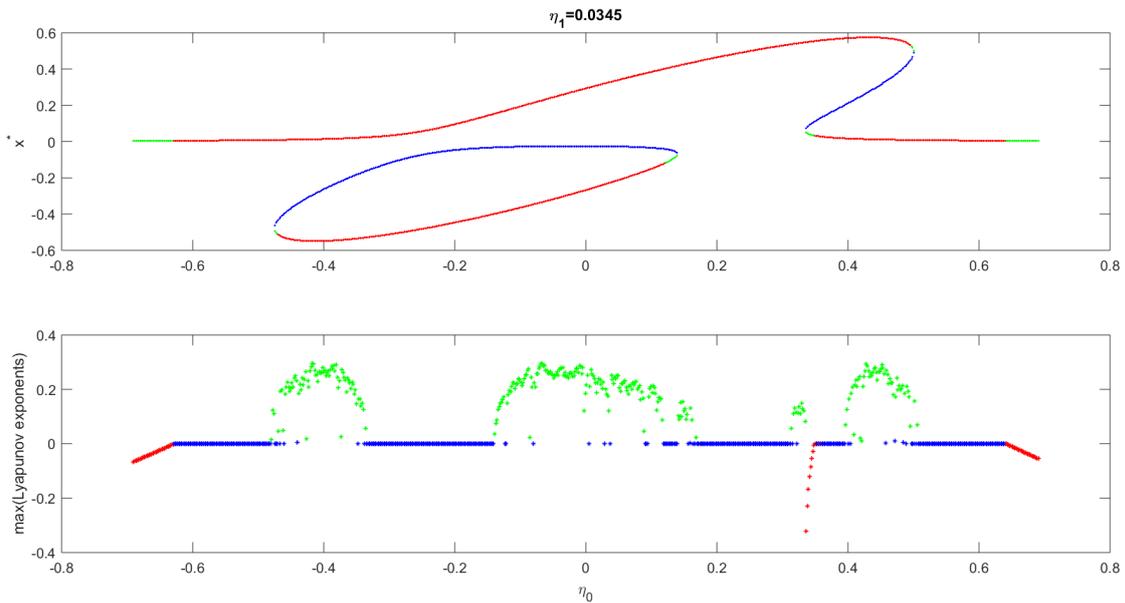


FIGURE 3.13: Equilibrium points and maximum Lyapunov exponent of 3.24 for $\eta_1 = 0.0345$ and varying η_0 . Stable equilibrium points in green, unstable equilibrium point in red (2 positive eigenvalues) or blue (only 1 positive eigenvalue). Maximum Lyapunov exponent in red if negative, in green if positive and in blue if ≈ 0 (periodic regime).

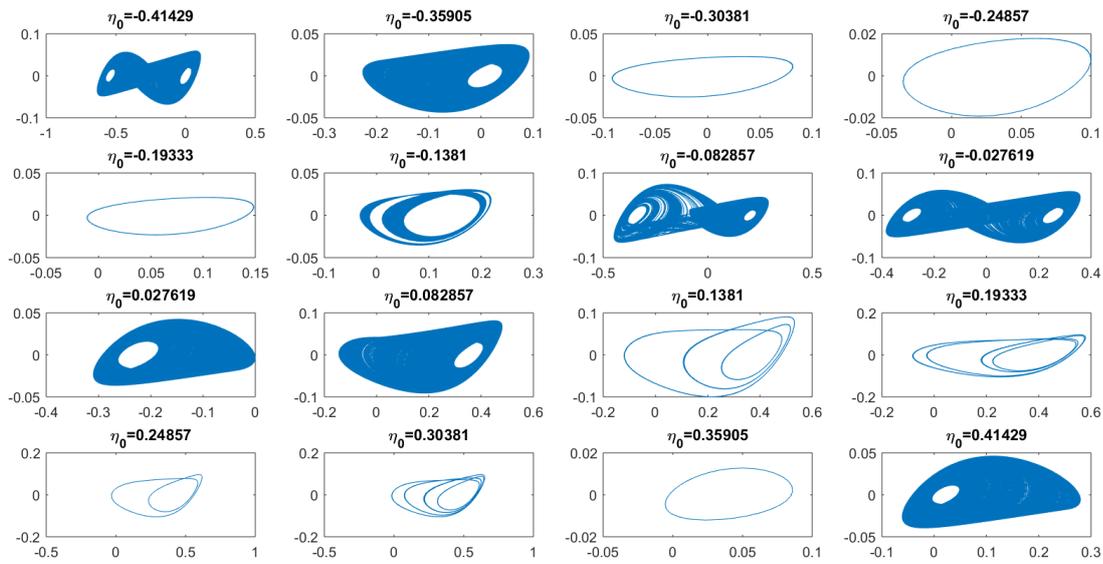


FIGURE 3.14: Phase portraits' projections on $x - y$ plane of 3.24 in $(\varphi - q)$ domain for $\eta_1 = 0.0345$ and varying η_0 .

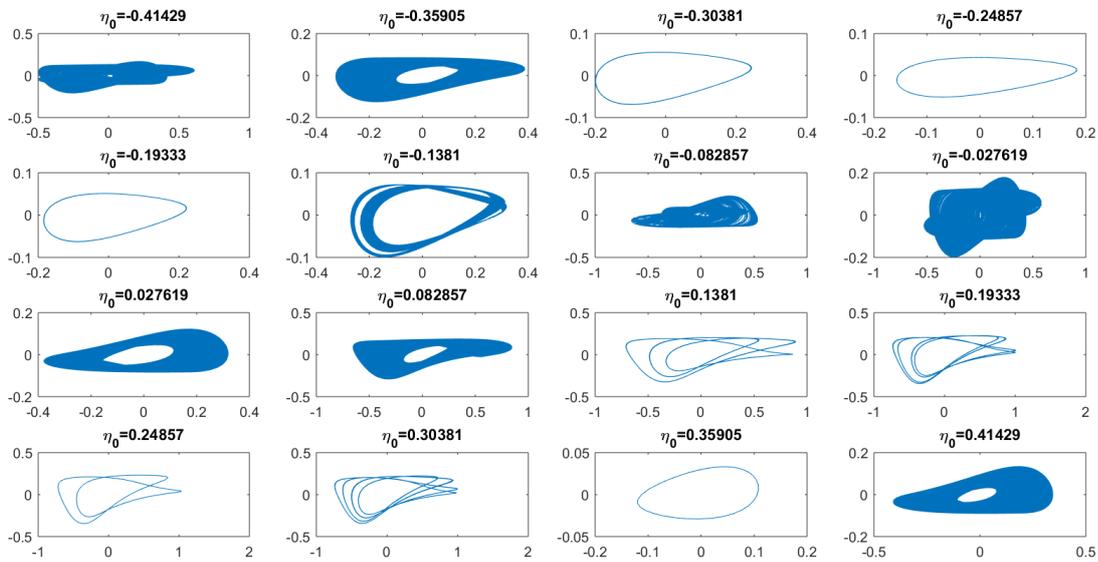


FIGURE 3.15: Phase portraits' projections on $v_{C_1} - v_{C_2}$ plane of 3.24 in $(i - v)$ domain for $\eta_1 = 0.0345$ and varying η_0 .

Circuit simulation

A circuit implementing the SEs 3.17 for the parameters' values assumed in "numerical simulations" is reported in figure 3.16. The chosen SPICE simulator is LTSpice by Linear Technology.

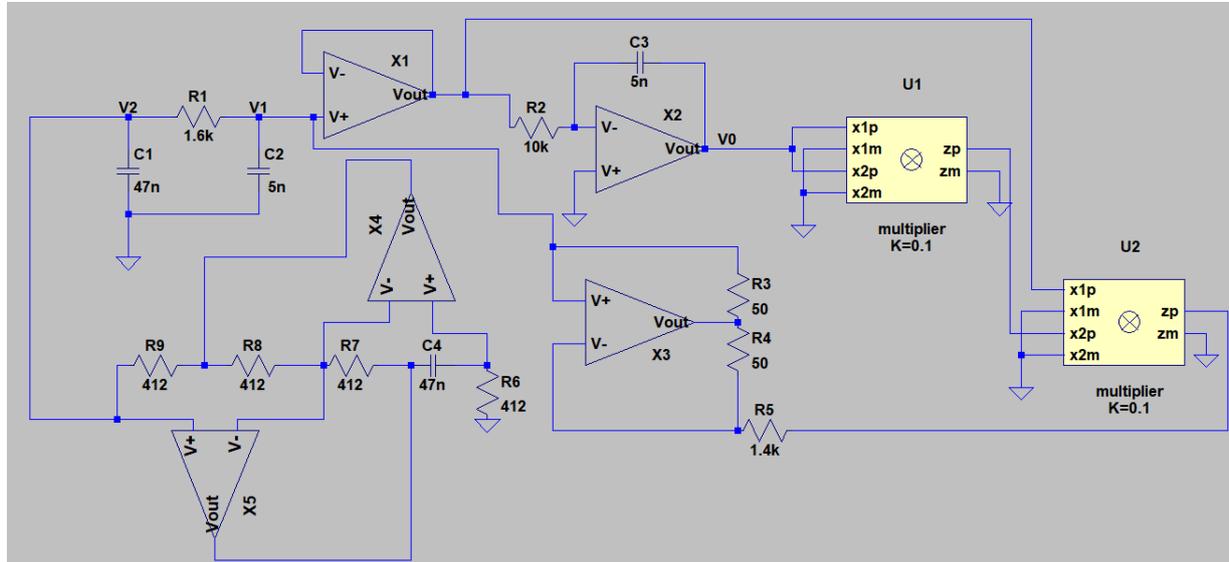


FIGURE 3.16: Schematic for the circuit implementing the set of equations 3.17.

The multiplier block is behaviourally implemented in SPICE syntax as follows.

```
.subckt multiplier x1p x1m x2p x2m zp zm params: K=0.1
Eres zp zm value={K*(V(x1p)-V(x1m))*(V(x2p)-V(x2m))}
.ends multiplier
```

The ideal opamp is implemented as reported in figure 3.17.

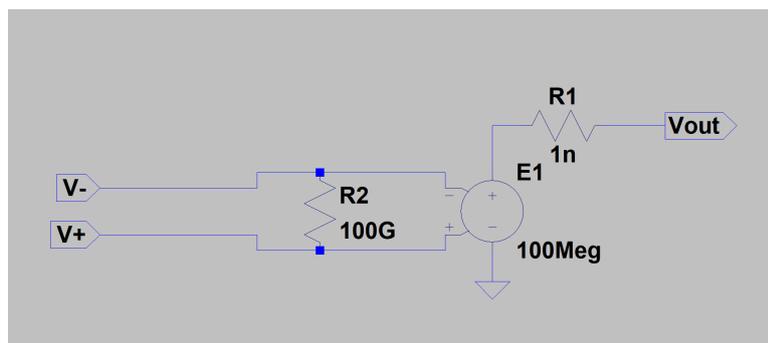


FIGURE 3.17: Implementation of an ideal operational amplifier.

The coexistence of multiple chaotic attractors and stable periodic regimes is confirmed by the circuitual simulations as reported in figures 3.18, 3.19, 3.20, 3.21, 3.22, 3.23. Fixed point attractors are not reported even if encountered because not visible on the $v_{C_1} - v_{C_2}$ plot.

The simulations were performed keeping $\eta_1 = 0.0345$ fixed and varying only η_0 . The required maximum time step to have accurate results is $T_{step,max} = 1ns$.

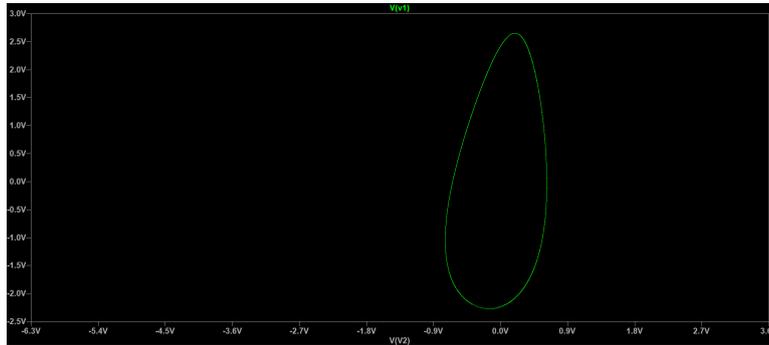


FIGURE 3.18: LTspice simulation for $\eta_0 = -0.2071$ and $\eta_1 = 0.0345$.

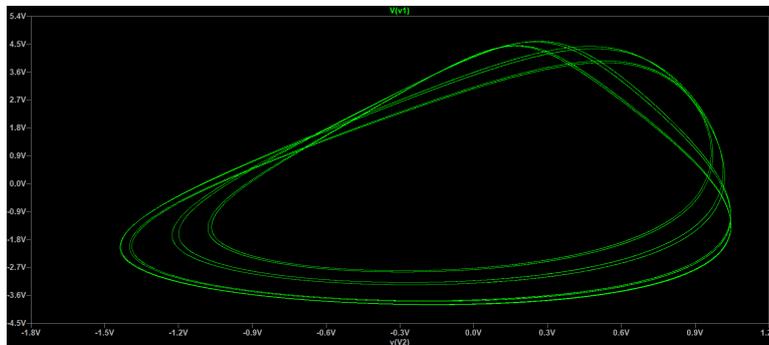


FIGURE 3.19: LTspice simulation for $\eta_0 = -0.1036$ and $\eta_1 = 0.0345$.

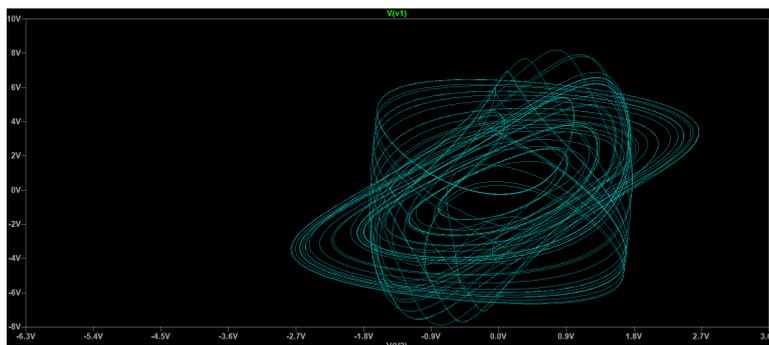


FIGURE 3.20: LTspice simulation for $\eta_0 = 0$ and $\eta_1 = 0.0345$.

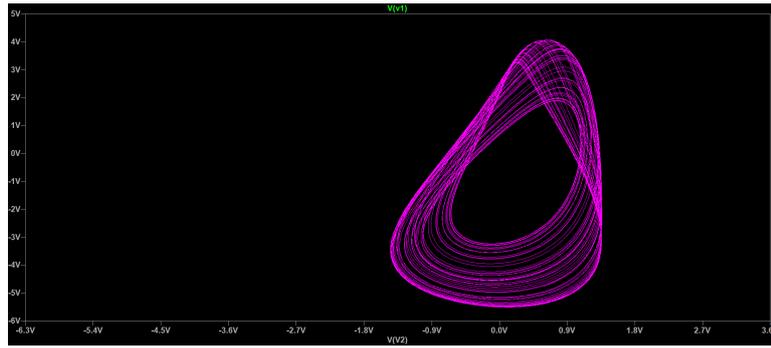


FIGURE 3.21: LTspice simulation for $\eta_0 = 0.0690$ and $\eta_1 = 0.0345$.

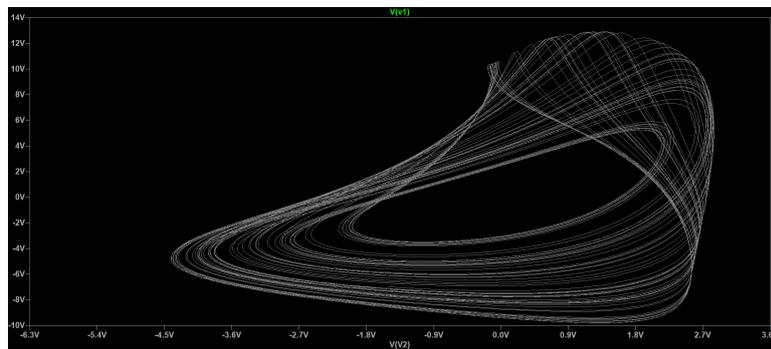


FIGURE 3.22: LTspice simulation for $\eta_0 = 0.1380$ and $\eta_1 = 0.0345$.

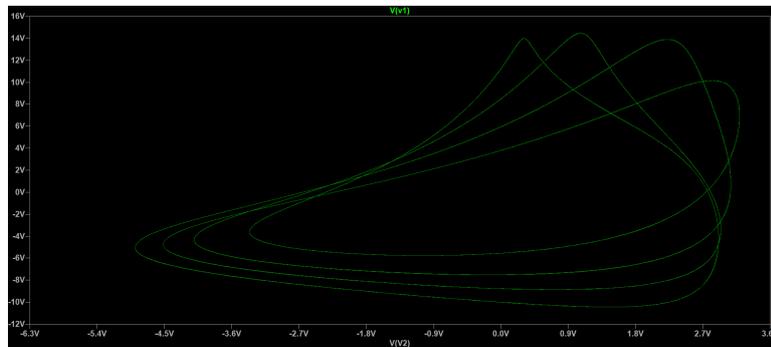


FIGURE 3.23: LTspice simulation for $\eta_0 = -0.2071$ and $\eta_1 = 0.0345$.

Pulse programming

As proposed in [10] the chaotic memristive Chua circuit can be programmed via pulses. In order to do that the system has to become non-autonomous by introducing independent voltage (or current) generators in the circuit. This addition allows to change dynamically the manifold on which the evolution of state variables takes place. Here just as example an induced bifurcation is reported. This was obtained by adding a voltage source with an high R_{off} series electrically controlled switch as reported in figure 3.24.

The projection of the phase portait on $v_{C_1} - v_{C_2}$ plane is reported in figure 3.25.

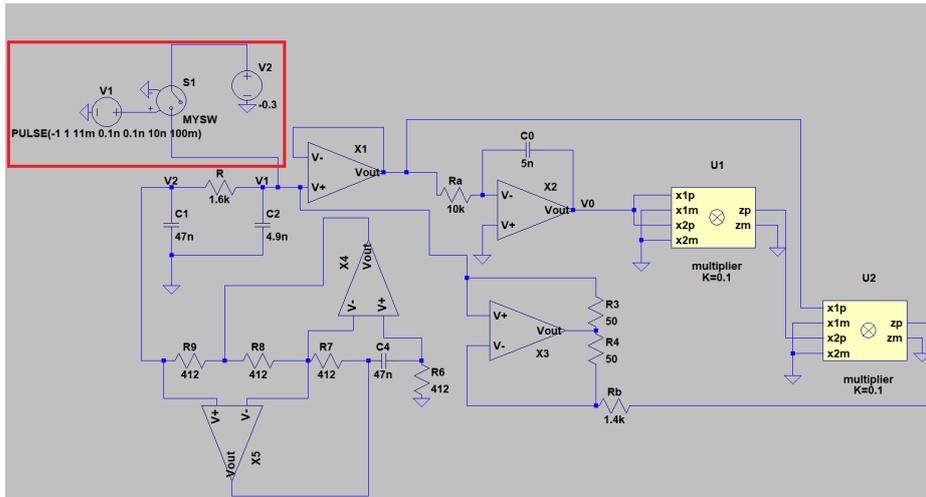


FIGURE 3.24: Same circuit as in figure 3.16 just with the addition of an electrically controlled switch, an independent controlling voltage source and a constant voltage source.

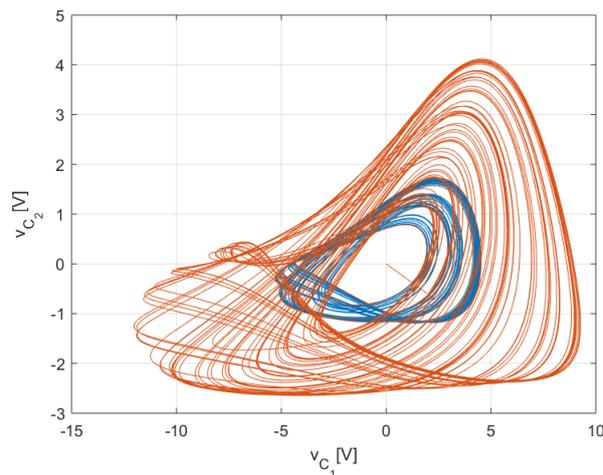


FIGURE 3.25: LTspice simulation for $\eta_0 = 0.0690$ and $\eta_1 = 0.0345$ in blue. Bifurcation without parameters induced by a voltage pulse of duration $\tau_{pulse} = 1.33e - 4$ and amplitude $A = 0.021$ in orange.

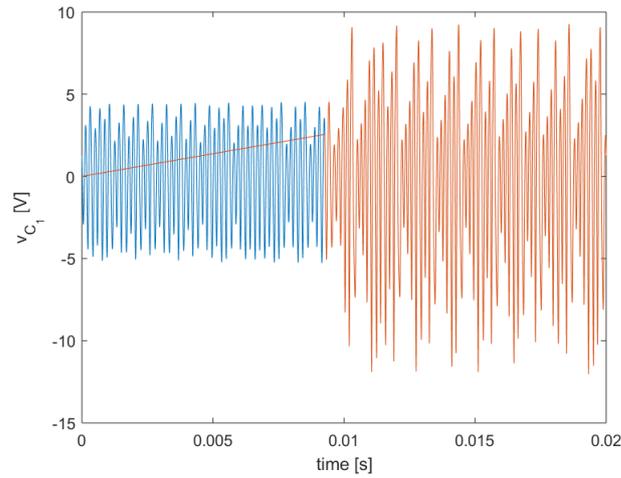


FIGURE 3.26: LTspice simulation waveforms for $\eta_0 = 0.0690$ and $\eta_1 = 0.0345$ in blue. Bifurcation without parameters induced by a voltage pulse of duration $\tau_{pulse} = 1.33e - 4$ and amplitude $A = 0.021$ in orange.

3.2.6 Physical implementation

The implementation of the circuit as it is in figure 3.16 was attempted using commercial operational amplifiers (AD711K) and commercial analog multipliers (AD633J) as in [18]. The choice of using the same components was dictated by the fact that only those analog multipliers were found with readily usable SPICE models. Even though LTspice models are available for those components they show convergence problems. The only solution to carry simulations of the whole circuit was to use the Analog Devices proprietary simulation tool SIMetrix.

Dealing with nonidealities

What is clear by looking at 3.16 is the fact that an infinite gain amplifier is used in the ideal memristor emulator circuit. This is a problem because any parasitic constant offset injected into the integrator by the preceding voltage follower is integrated over time and causes the saturation of the internal memristor state V_0 .

If on the one hand the use of a negative feedback resistor is mandatory on the other one this resistor heavily changes the memristor characteristic. The modelling of how the SEs are modified by the introduction of the feedback resistor is left to future work.

Given the two integrator topologies as in figure 3.27 the crossover frequency f_c 3.30(unity gain frequency) is the same for both of them.

$$f_c = \frac{1}{2\pi R_{in} C} \quad (3.30)$$

The $-3dB$ cut frequency f_{3dB} of the finite gain integrator is given by 3.31.

$$f_{3dB} = \frac{1}{2\pi R_f C} \quad (3.31)$$

The effect of increasing R_f is graphically showed in 3.28. When increasing the feedback resistance the pole is moved towards the origin and the in-band gain is increased.

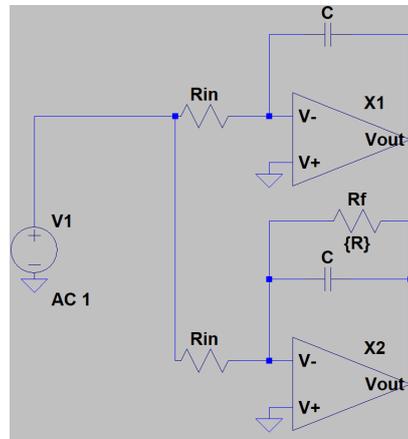


FIGURE 3.27: Infinite gain integrator (upper one) and finite gain integrator (lower one).



FIGURE 3.28: Transfer function of a finite gain amplifier in green for various values of R_f . Transfer function of an infinite gain amplifier in blue.

Circuit simulation

The analogue of 3.16 simulated on SIMetrix is reported in 3.29. The Antoniou inductance emulation circuit could not be included in the SIMetrix simulation because of licence limitations thus the corresponding ideal inductance was used instead.

Many different values of feedback resistance R_f were tried in the search of the same extreme multistable behaviour shown in ideal circuit simulation. All the simulation were run for initial conditions $i_L(0) = 0A$, $V_{C_2}(0) = 0V$, $V_{C_1} = 1V$ and varying $V_0(0)$.

For $R_f \leq 2k\Omega$ the system shows only stable point attractors (not reported here). Simulations in figures 3.30, 3.31, 3.32, 3.33 are for $3k\Omega \leq R_f \leq 6k\Omega$, those values of R_f do not show any sensitivity with respect to $V_0(0)$. In fact for each value of R_f

only one kind of attractor was found varying $V_0(0)$.

For $R_f = 7k\Omega$ different initial evolutions are detected, as reported in figures 3.34, 3.35 and 3.36, but those are overwhelmed by the saturation of internal state of memristor emulator V_0 due to input parasitic voltage to the integrator. The result of setting $R_f = 7k\Omega$ is a square wave generator.

The final conclusion is that the implementation of the ideal circuit in figure 3.16 is not feasible using real components because of the presence of parasitic voltage and current generators in the circuit which cause the ideal integrator's divergence. The first thought solution of using a feedback resistance, in order to make the integrator gain finite, is not successful. This is because on the one hand if the feedback is strong then the memristor emulator behaves just like a non-linear resistor loosing its $v - i$ memory characteristic and making possible bifurcations only on a fixed manifold, on the other hand if the feedback is weaker then the ideal memristor emulator suffers from the saturation of internal variable V_0 .

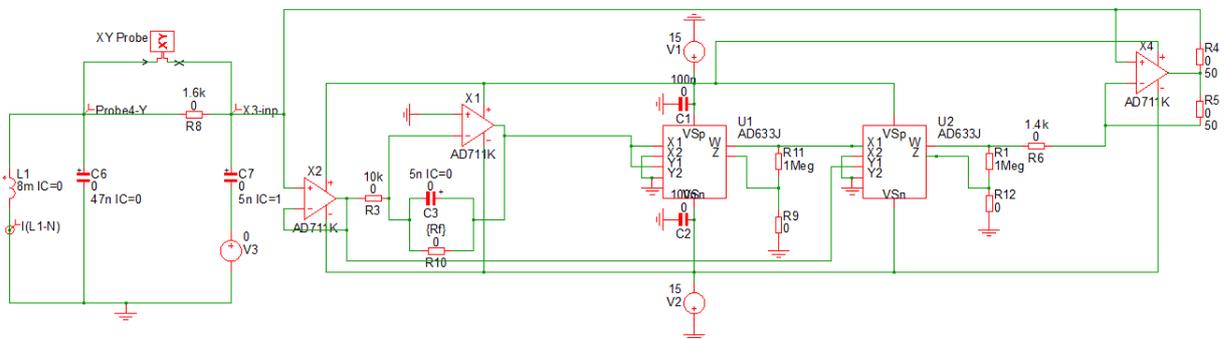


FIGURE 3.29: Schematic for the circuit in figure 3.16 implemented with real components.

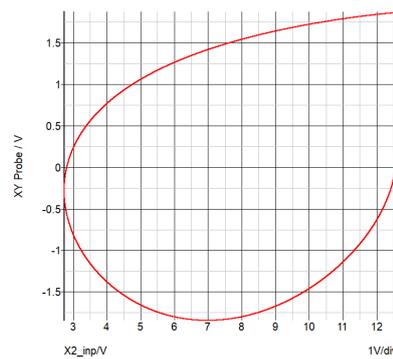


FIGURE 3.30: SIMetrix simulations for $R_f = 3k\Omega$, $V_{C_1}(0) = 1V$ and varying $V_0(0)$.

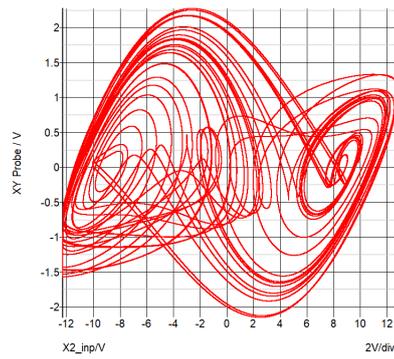


FIGURE 3.31: SIMetrix simulations for $R_f = 4k\Omega$, $V_{C_1}(0) = 1V$ and varying $V_0(0)$.

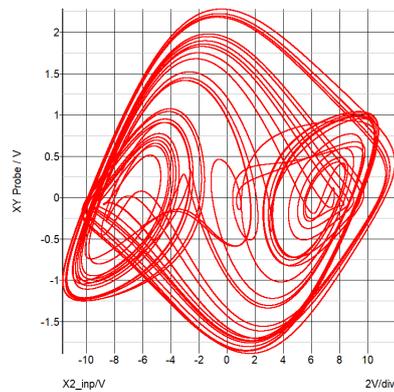


FIGURE 3.32: SIMetrix simulations for $R_f = 5k\Omega$, $V_{C_1}(0) = 1V$ and varying $V_0(0)$.

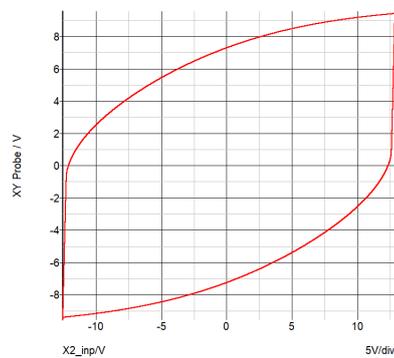


FIGURE 3.33: SIMetrix simulations for $R_f = 6k\Omega$, $V_{C_1}(0) = 1V$ and varying $V_0(0)$.

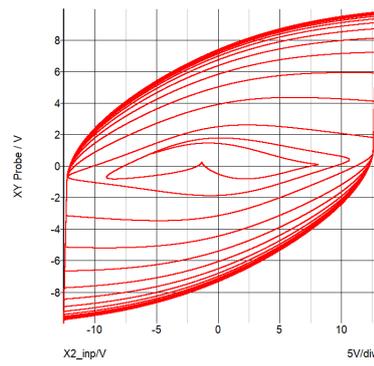


FIGURE 3.34: SIMetrix simulations for $R_f = 7k\Omega$, $V_{C_1}(0) = 1V$ and $V_0(0) = -2V$.

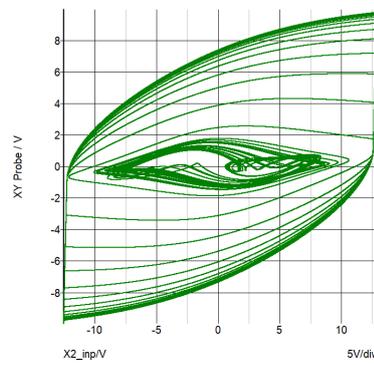


FIGURE 3.35: SIMetrix simulations for $R_f = 7k\Omega$, $V_{C_1}(0) = 1V$ and $V_0(0) = -1V$.

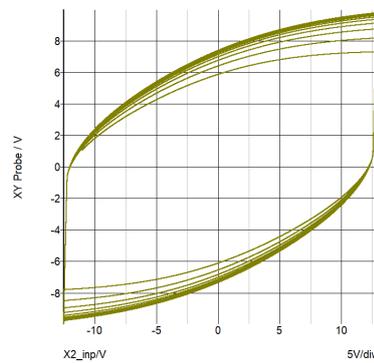


FIGURE 3.36: SIMetrix simulations for $R_f = 7k\Omega$, $V_{C_1}(0) = 1V$ and $V_0(0) = 1V$.

Laboratory acquired measurements

The circuit in figure 3.29 was actually mounted on a breadboard and tested in laboratory as in figure 3.37. Differently from circuit in figure 3.29 the implemented circuit used Antoniou's circuit as described in [17] (bottom right part of the picture).

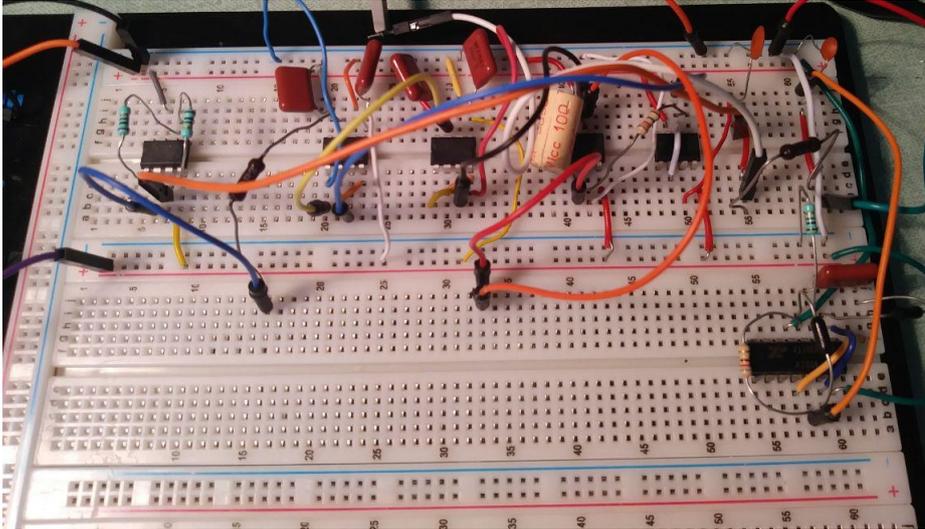


FIGURE 3.37: Picture of the mounted circuit.

The circuit components parameters are measured as: $L = 7.87\text{mH}$, $C_2 = 37.4\text{nF}$, $C_1 = 4.3\text{nF}$, $C_0 = 4.9\text{nF}$, $R_a = 9.811\text{k}\Omega$, $R = 1.709\text{k}\Omega$, $R_B = 1.747\text{k}\Omega$ and $R_f = 6.734\text{k}\Omega$.

Using the oscilloscope Agilent 2024A available in laboratory the waveforms in figure 3.38 and the phase space projection in figure 3.39 were obtained.

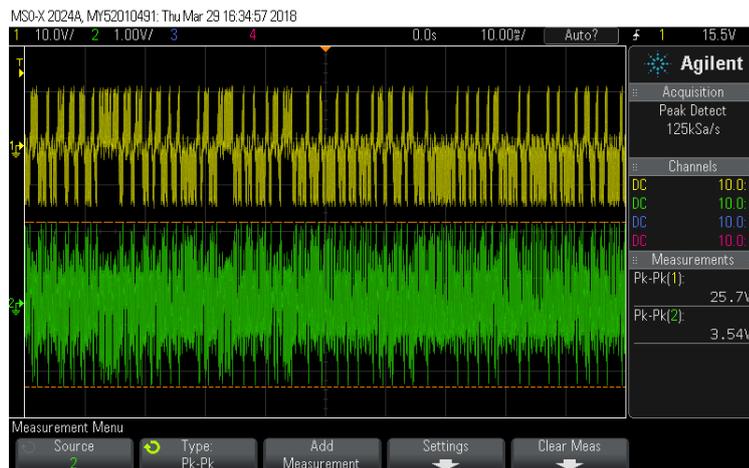


FIGURE 3.38: In green v_{C_1} and in yellow v_{C_2} .

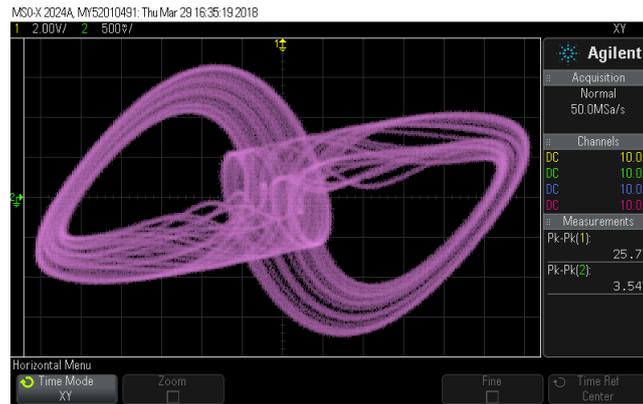


FIGURE 3.39: Projection on the $v_{C_1} - v_{C_2}$ plane of the chaotic attractor in the phase space.

The measured behaviour recalls what was obtained in simulation for low levels of feedback resistance as in figure 3.31. Changing the R_f resistance value using a potentiometer the same saturation effects were recognizable: saturated purely deterministic waveforms for high values of R_f and limit cycles for low values of R_f .

Chapter 4

Application of chaos

4.1 Artificial Neural Networks

Artificial neural networks (ANNs) are computing systems inspired by the biological neural networks. They learn new tasks by considering examples, generally without any a priori knowledge about the task itself. From the processed learning material ANNs evolve their specific set of relevant characteristics.

ANNs are based on a set of connected units or nodes called artificial neurons. All the connections between artificial neurons can transmit a signal from one to another. The artificial neuron receiving the signal processes it and then transmits it to the connected artificial neurons.

Usually the signal transmitted along a connection is a real number, and each artificial neuron returns in output a non-linear function of the weighted sum of its inputs. Connections usually have an associated weight that is adjusted during the learning phase. The weight is an attenuation/amplification factor of the signal transmitted on the connection. Artificial neurons could have a threshold, this means that only if the weighted sum of the inputs is above that threshold the signal is transmitted.

The original naive goal of ANNs was to emulate the function of human brain as a generic problem solver. Still today the research field of general artificial intelligence is in its embryonic phase mainly because of the lack of understanding of how complex animal brains work. Nonetheless the interest on ANNs has enormously grown in the last decade because of their capability of exceeding the human abilities in specific tasks such as computer vision, speech recognition, interlanguage translation, playing complex games and medical diagnosis.

4.2 Recurrent Neural Networks

A recurrent neural network (RNN) is a type of ANN whose artificial synapses form an oriented graph along a sequence. This feature makes it a proper dynamical system. RNNs use their internal state to process sequences of data. This characteristic makes them suitable for connected handwriting recognition or speech recognition.

A FIR RNN is an oriented acyclic graph which can be unrolled and substituted by a feedforward ANN, an IIR RNN is an oriented cyclic graph which cannot be unrolled.

4.3 Hopfield Neural Networks

A Hopfield neural network is a class of RNNs that can work as content-addressable memory (CAM). Under certain assumptions they are guaranteed to converge to a

local minimum, sometimes they may converge to a wrong local minimum instead. The artificial neurons in Hopfield networks are binary threshold units. Each neuron can only take two different values (usually ± 1) for their status.

4.3.1 Hopfield NN topology

Every pair of distinct artificial neurons in a Hopfield network has a connecting synapse. The connections in a Hopfield net typically have the following restrictions:

- $w_{ii} = 0 \quad \forall i$: no unit has a connection with itself.
- $w_{ij} = w_{ji} \quad \forall i, j$: connections are symmetric.

The symmetry of weights guarantees the monotonical decrease of the energy function while updating the network state. If weights are asymmetric periodic or chaotic behaviour can occur.

An example of a simple Hopfield neural network is reported in figure 4.1.

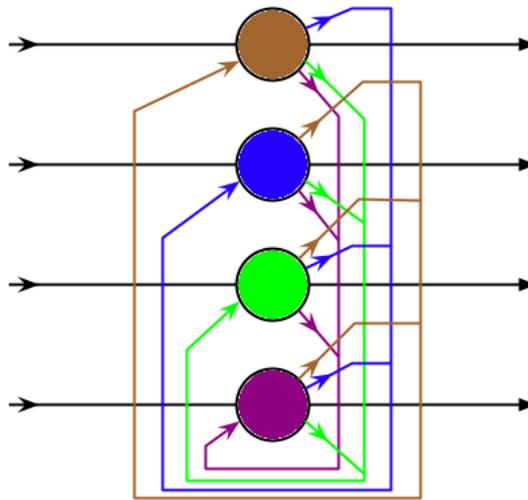


FIGURE 4.1: Hopfield network made of 4 neurons.

4.3.2 State update

Hopfield Networks are initialized by setting the values of the neurons to a desired start pattern which can be for example a corrupted pattern to reconstruct. Updates are repeated until the NN converges to an attractor state. The rule used in order to update the network state is 4.1.

$$S_i \leftarrow \begin{cases} +1 & \text{if } \sum_j w_{i,j} S_j \geq \theta_i \\ -1 & \text{otherwise} \end{cases} \quad (4.1)$$

- $w_{i,j}$: the weight of the connection from unit j to unit i .
- S_j : the state of neuron j .
- θ_i : threshold of unit i .

The rule 4.1 can also be written as a discrete dynamical system IVP 4.2.

$$\begin{cases} S_{i+1} = \text{sign}(WS_i - \theta) \\ S_0 = X \end{cases} \quad (4.2)$$

- W : the matrix of weights.
- S_i : the state vector at iteration i .
- θ : the threshold vector.
- X : the input pattern.

Two different ways of updating the network are possible:

- **Asynchronous**: neurons are updated one by one sequentially in random or pre-defined order.
- **Synchronous**: neurons are updated all at once. This way is considered as less realistic because it does not reflect analogous physical or biological systems.

If $w_{i,j} > 0$ 4.1 implies:

- if $S_j = 1$ then S_i is pulled by S_j towards $S_i = 1$
- if $S_j = -1$ then S_i is pulled by S_j towards $S_i = -1$

If the weight of a synapse connecting two neurons is positive then they will converge. They will diverge otherwise. This concept is often summarized as "*Neurons that fire together, wire together. Neurons that fire out of sync, fail to link*" [12].

4.3.3 Energy function

Hopfield NN have an associated function called "energy" 4.3 which associates to each state of the network a scalar value.

$$E = -\frac{1}{2} \sum_{i,j} w_{i,j} S_i S_j - \sum_i \theta_i S_i \quad (4.3)$$

If neurons' update order is randomly chosen, the energy function E will either lower or stay constant. By iterated updates the NN will converge to a state that is a local minimum of the energy function.

4.3.4 Training

Hopfield networks are trained by lowering the energy of states that should become attractors. In this way the NN can work as a CAM system: the Hopfield net will converge to a "learnt" state just by providing it with an incomplete part of the data. The network associates the most similar trained state to a distorted input. It can recover memories on the basis of similarity. In order to correctly train the network the energy of states which the network should remember must be local minima.

Hebbian learning rule

For Hopfield Networks the Hebbian learning rule is implemented as 4.4.

$$w_{ij} = \frac{1}{n} \sum_{\mu=1}^n \epsilon_i^\mu \epsilon_j^\mu \quad (4.4)$$

Where:

- n is the number of patterns to store.
- ϵ_i^μ is bit i from pattern μ .
- w_{ij} is the weight of the synapse connecting neuron i and neuron j .

If the bits i and j are equal in pattern μ , then $\epsilon_i^\mu \epsilon_j^\mu \geq 0$. This has a positive effect on the weight w_{ij} and the values of units i and j will tend to converge. If the bits i and j are different the opposite happens.

Storkey learning rule

Storkey [14] proved that a Hopfield network trained using his proposed rule has a greater capacity than one trained with the Hebbian rule. The weights are set according to 4.5. The Storkey rule uses more information from the training set and weights than the Hebbian rule, because of the effect of the local field 4.6.

$$w_{i,j}^v = w_{i,j}^{v-1} + \frac{1}{n} \epsilon_i^v \epsilon_j^v - \frac{1}{n} \epsilon_i^v h_{j,i}^v - \frac{1}{n} \epsilon_j^v h_{i,j}^v \quad (4.5)$$

$$h_{i,j}^v = \sum_{k=1, k \neq i, j}^n w_{i,k}^{v-1} \epsilon_k^v \quad (4.6)$$

Spurious patterns

It sometimes happens that the network converges to patterns not contained in the training set, those are called *spurious patterns*. The spurious patterns are also local minimum of the energy function E . For example if x is a stored pattern, then $-x$ is a spurious pattern. Linear combinations of an odd number of retrieval states can also be spurious patterns.

Network capacity

The number of neurons and synapses within an Hopfield network determines the amount of patterns that can be stored, this amount is called the network capacity. It was shown [13] that approximately 138 patterns can be correctly retrieved from storage for every 1000 neurons. Many mistakes can occur if a large number of vectors is stored.

4.4 Introduction of chaos in Hopfield NN

The utility of chaotic signals in hardware accelerators has been shown by Kumar, Strachan and Williams [16]. Inclusion of tunable chaos generated by a memristive circuit can enhance computational efficiency in problems requiring global minimization. This enhancement comes with little energy overhead and no additional latency. The goal of displacing a system from a local minimum can also be achieved with a computer-generated pseudo-random sequence of numbers (CGPRS) instead of a chaotic signal. The ease of integration of memristors into ASIC accelerators is the fundamental improvement over CGPRS.

In the following to confirm the results obtained by Kumar et Al. [16] an Hopfield neural network for corrupted data reconstruction was implemented in Python Programming Language. A chaotic signal obtained via circuit simulation was introduced in the threshold level of each neuron in order to show that chaos can enable, by displacing from a local minimum, convergence towards the global minimum.

4.4.1 The dataset, the network and the training

The dataset used in the simulations is a set of 12 characters encoded as 10x10 binary bitmaps reported in figure 4.2. Those are coded using bipolar binary code: white pixels are coded with -1 and black pixels with 1 .

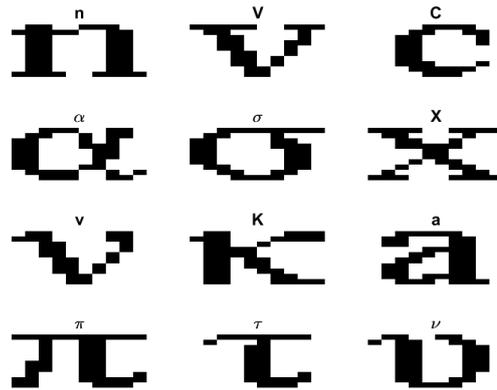


FIGURE 4.2: The used dataset of 12 images encoded as 10x10 binary bitmaps.

Because of the images dimension (10x10) the network is made of 100 neurons and 10,000 synapses.

The correct working of the network was checked by inputting the uncorrupted stored patterns and verifying their correct self association. Training the network with Hebbian learning rule only 6 patterns could be stored. By upgrading to the Storkey learning rule up to 11 patterns could be stored and correctly recalled excluding the ν which is too similar to v .

4.4.2 Simulation methodology

At the beginning of any simulation a random updating order was computed. Then the states were updated in that random sequential order. In every update, the chaotic

signal (v_{C_1}) obtained from the memristive Chua circuit was sampled at periodic intervals. Assuming a single chaotic source whose output signal is dispatched to all the thresholding units only one chaotic waveform, reported in figure 4.5, was sufficient. Given an updating order the propagation delay of the chaotic signal between one neuron and the next one was assumed to be the sampling time $T_s = 5\mu s$.

Upon simulating the NN, the energy of the system decreased until a minimum was reached. If chaos is not included, the energy converged to a minimum and stayed constant through all the following updates as shown in figure 4.3.

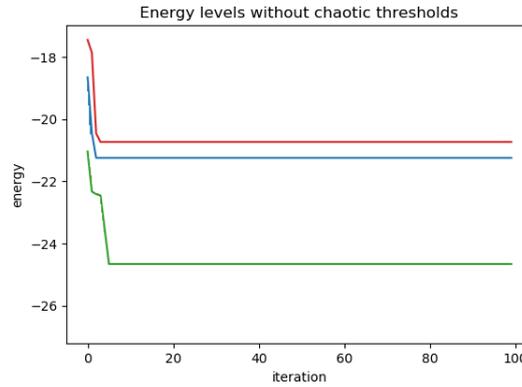


FIGURE 4.3: The energy levels evolution in 3 different simulations without chaotic thresholds and 30% corruption level.

Using the same initial input patterns and order of updating, upon inclusion of chaotic signal on thresholding units, the energy monotonically reduces and at the same time undergoes fluctuations. Those correspond to instantaneous variation of states due to chaotic threshold levels. The energy on average evolves to values below the case with no chaotic thresholds hence escaping local minima as shown in figure 4.4.

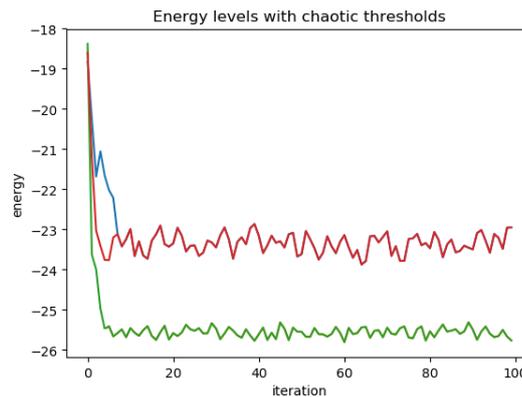


FIGURE 4.4: The energy levels evolution in 3 different simulations with chaotic thresholds scaled to 8% of the waveform in figure 4.5 and 30% corruption level.

Because of fluctuations there is no strict convergence, this is a problem for the verification of the correct pattern recovery. This problem was addressed by introducing a 5th order FIR filter on each state output, the verification is made when

the programmed updates are over (250 steps) on the average of the last 5 values assumed by each unit.

A block scheme of the simulation setup is reported in figure 4.6.

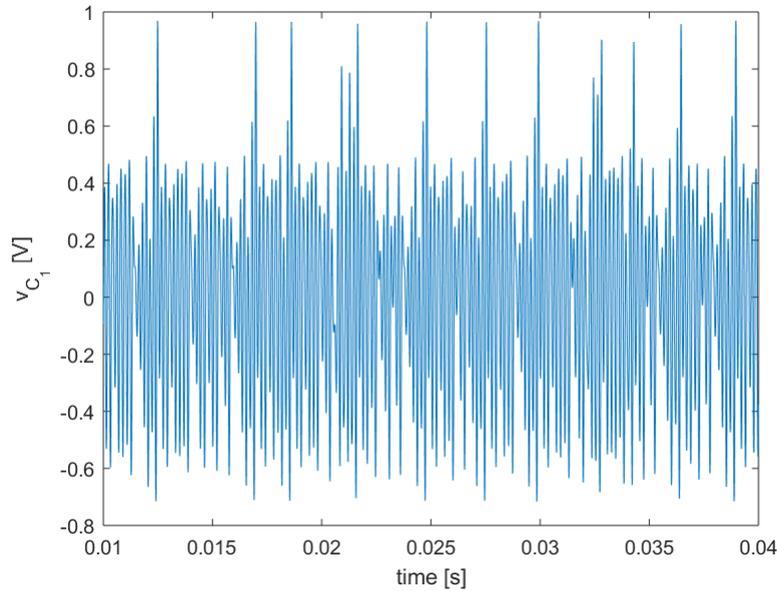


FIGURE 4.5: The chaotic waveform injected into the thresholding units.

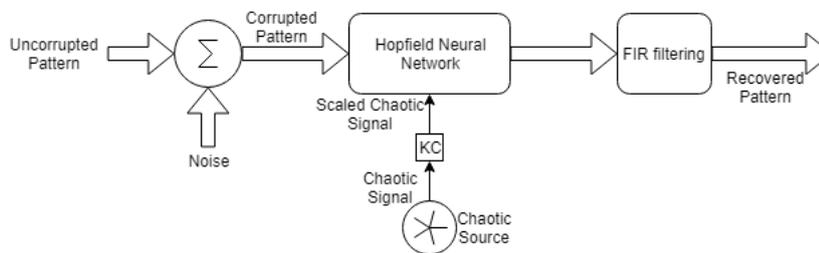


FIGURE 4.6: The simulation setup.

The amplitude of fluctuations

An important parameter to consider is the magnitude of chaotic signal that is scaled by the coefficient KC . By properly setting the scaling factor a reduction in the energy of the solutions is enabled.

For high magnitude of fluctuations (e.g. $KC = 0.3$), the range of energies is significantly higher compared to the case with no chaos ($KC = 0$). The high magnitude of chaotic thresholds drives the NN across a wide range of outputs, thus producing a wide range of energies.

The case with an intermediate magnitude of chaos ($KC \in [0.05, 0.15]$), enables a reduction in the energy of most of the solutions with respect to $KC = 0$ case. This is because of the magnitude of chaotic signal being high enough to displace the NN out of a local minimum delimited by low energy barriers, while the same magnitude is not enough to displace the NN out of a global minimum.

Those results are visualized in figure 4.7.

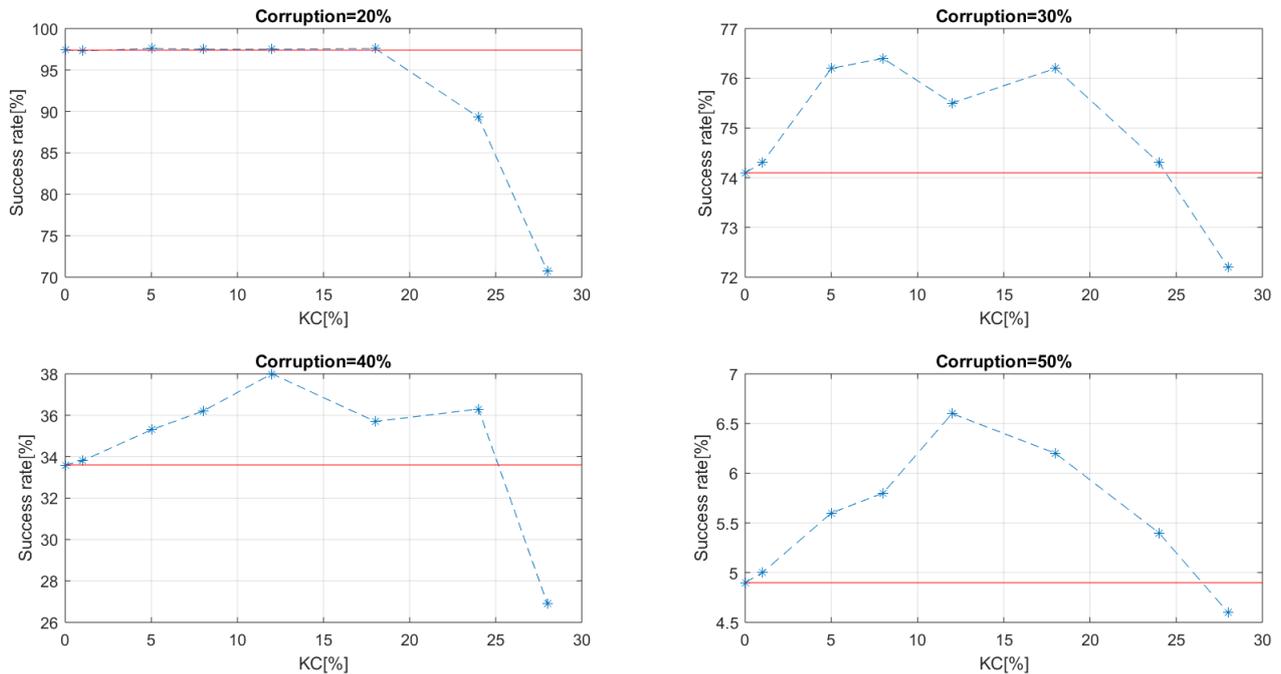


FIGURE 4.7: Success rates for different values of corruption (20%, 30%, 40% and 50%) and different amplitudes of chaotic thresholds (KC). In red the line representing the level of KC = 0

The results reported in figure 4.7 were obtained simulating the Hopfield neural network, as previously described, for 5 batches of 200 corrupted patterns. Before start feeding a new batch to the network, the update order of neurons was permuted. Simulations were run for different levels of KC and corruption level. The latter one is intended as $\frac{\#uncorrectpixels}{\#correctpixels}$.

Example of pattern reconstruction enhanced by chaotic thresholds

An example of the actual enhanced reconstruction capability enabled by the injection of a chaotic signal on the thresholding units is provided in figure 4.8. In that figure are reported all the patterns that could not be recovered using non-chaotic threshold levels and the corresponding fully recovered patterns obtained by injection of chaos. Those are organized in rows couples where the first of each couple is the case with KC = 0 and the other one is the case with KC = 8%.

The distorted image (30% corruption) is reported as well as initial pattern and reconstructed pattern in order to show the kind of noise the NN recovers a pattern from.

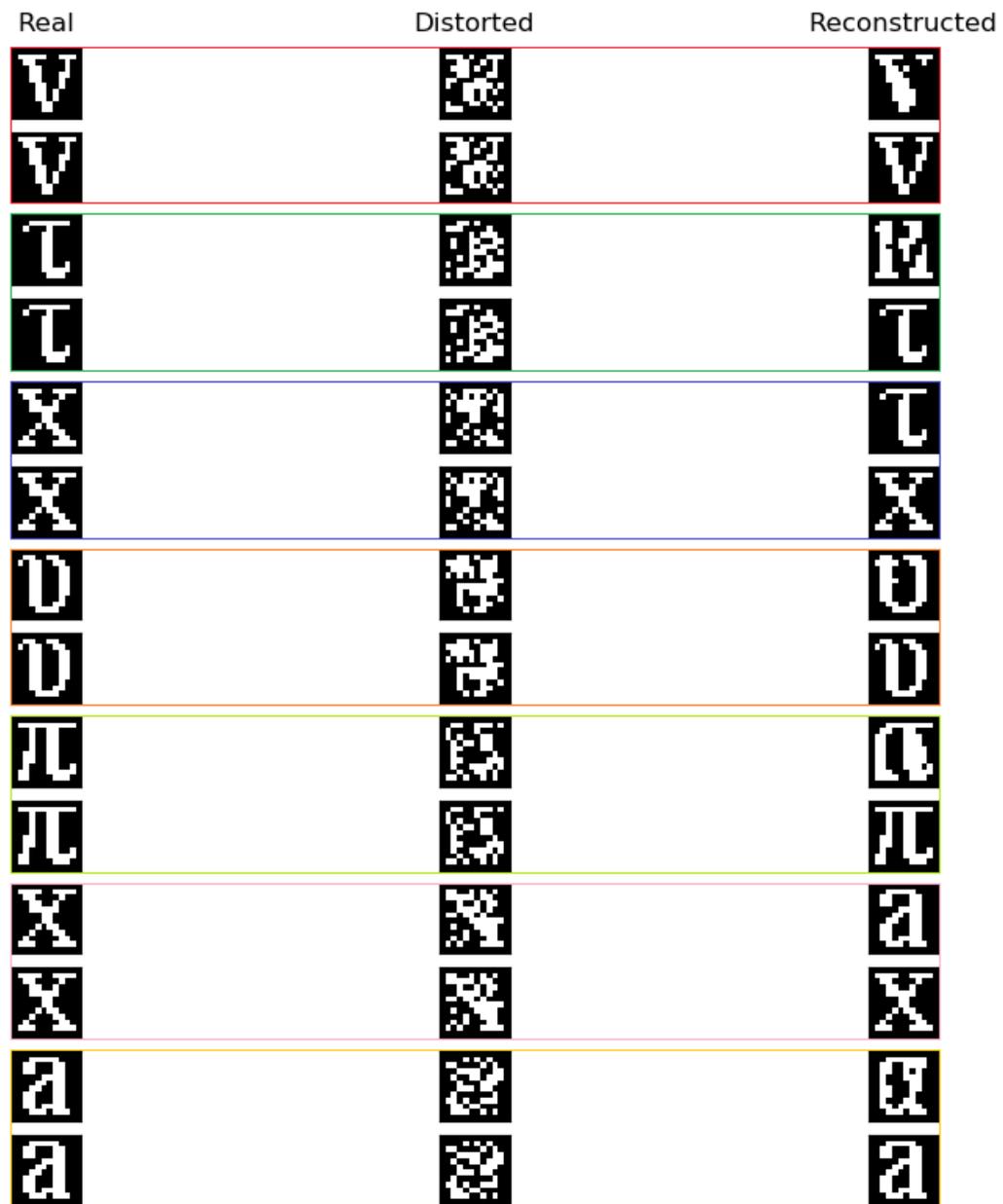


FIGURE 4.8: Example of pattern recovery enhanced by chaotic signal applied on neuron threshold. $KC = 8\%$ and corruption level 30% on second, forth and all even rows. $KC = 0$ and corruption level 30% on first, third and all odd rows.

Chapter 5

Conclusions and future work

5.1 The obtained results

The goals of this thesis work were: (1) to show how the use of a memristive component inside a chaotic oscillator could allow dynamical programming of the latter on different chaotic attractors, and (2) to show how chaos could enhance the performance of a bio-inspired analogue computational system.

The first goal was partially achieved. The study of the ideal memristive Chua circuit ODEs and the circuital simulation of its ideal implementation confirmed the extreme multistability property. By applying pulses it was shown the possibility of switching between different chaotic attractors. Those results eventually faced the non-ideality of real components and proved that the ideal memristor emulator proposed in [18] is not valid as approach for implementing extreme multistability in physical chaotic oscillating circuits.

The second goal was fully achieved: for highly distorted patterns the injection of a chaotic signal on the threshold level of the artificial neurons leads to an increase of successful reconstruction rate up to 4% and average reduction in energy level for most of the applied input patterns.

5.2 Future work

The formal study of the non-ideal memristor emulator and how to obtain, using it, the desired extreme multistability could be an interesting topic of future research work.

The study of how the use of different chaotic attractors affects the enhancement of successful reconstruction rate and the comparison with stochastic noisy signals could also be another interesting topic to explore.

The study of the global architecture implemented as an actual memristive neural network and its electrical simulation can also be subject worth of attention.

Bibliography

- [1] Axelle Amon. *Nonlinear dynamics and chaos*. <http://blogperso.univ-rennes1.fr/axelle.amon/index.php/pages/Teaching>.
- [2] L. O. Chua. "Dynamic nonlinear networks: State-of-the-art". In: (1980). URL: <http://ieeexplore.ieee.org/document/1084745/>.
- [3] L. O. Chua. "If it's pinched it's a memristor". In: (2014). URL: <http://iopscience.iop.org/article/10.1088/0268-1242/29/10/104001/pdf>.
- [4] L. O. Chua. "Memristor-The Missing Circuit Element". In: (1971). URL: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.189.3614&rep=rep1&type=pdf>.
- [5] S. M. Chua L. O.; Kang. "Memristive devices and systems". In: (1976). URL: <https://doi.org/10.1109/2FPROC.1976.10092>.
- [6] Robert L. Devaney. "An Introduction to Chaotic Dynamical Systems (2nd ed.)" In: (2003). URL: <https://books.google.com/books?id=CjAnY99LwTgC>.
- [7] Leon O. Chua Fernando Corinto Pier Paolo Civalleri. "A Theoretical Approach to Memristor Devices". In: (2015). URL: <https://doi.org/10.1109/2FPROC.1976.10092>.
- [8] Fernando Corinto; Mauro Forti. "Memristor Circuits: Bifurcations without Parameters". In: (2017). URL: <http://ieeexplore.ieee.org/document/7827101/>.
- [9] Fernando Corinto; Mauro Forti. "Memristor Circuits: Flux—Charge Analysis Method". In: (2016). URL: <http://ieeexplore.ieee.org/document/7547974/>.
- [10] Fernando Corinto; Mauro Forti. "Memristor Circuits: Pulse Programming via Invariant Manifolds". In: (2017). URL: <http://ieeexplore.ieee.org/document/8023796/>.
- [11] Vasilii Govorukhin. *Calculation Lyapunov Exponents for ODE*. MathWorks File Exchange. <https://it.mathworks.com/matlabcentral/fileexchange/4628-calculation-lyapunov-exponents-for-ode>. Mar. 2004.
- [12] Donald Olding Hebb. "The organization of behavior: A neuropsychological theory." In: (2002). URL: http://s-f-walker.org.uk/pubsebooks/pdfs/The_Organization_of_Behavior-Donald_O._Hebb.pdf.
- [13] Anders S. Krogh Hertz John A. and Richard G. Palmer. *Introduction to the theory of neural computation*. Vol. 1. Westview press, 1991.
- [14] Amos Storkey. "Increasing the capacity of a Hopfield network without sacrificing functionality." In: (1997). URL: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.33.103&rep=rep1&type=pdf>.
- [15] Gregory S.; Stewart Duncan R.; Williams R. Stanley Strukov Dmitri B.; Snider. "The missing memristor found". In: (2008). URL: <http://www.ece.ucsb.edu/~strukov/papers/2008/Nature2008.pdf>.

-
- [16] John Paul Strachan & R. Stanley Williams Suhas Kumar. *“Chaotic dynamics in nanoscale NbO₂ Mott memristors for analogue computing.”* In: (2017). URL: <https://www.nature.com/articles/nature23307>.
- [17] Valentin Siderskiy. *The Antoniou Inductance-Simulation Circuit Derivation*. <http://www.chuacircuits.com/PDFs/Antoniou%20Inductance-Simulation%20Circuit.pdf>.
- [18] Mo Chen; Mengxia Sun; Bocheng Bao; Huagan Wu; Quan Xu; Jiang Wang. *“Controlling extreme multistability of memristor emulator-based dynamical circuit in flux–charge domain”*. In: (2017). URL: <https://doi.org/10.1007/s11071-017-3952-9>ssss.