To those I love and those who love me

"*Education does not end at any point in our lives;*
it is an ongoing journey to be carried with us
everyday throughout our lives."
Thomas Powell

Synopsis

The pioneering works of Prof. Leon O. Chua constructing electrical circuit (named *Chua's circuit*) which can display chaotic behaviors open a period with fruitful researches about nonlinear circuits. A considerable amount of different nonlinear circuits has been introduced and their numerous applications have developed instantaneously. Complex dynamical characteristics of nonlinear circuits are used to model complex systems, to generate random numbers or to secure informations. There are three nonlinear systems which have received much attention recently: time–delay system, Cellular Neural Network and memristive system. In this thesis we investigate how the properties of these nonlinear systems can be efficiently exploited to build novel chaotic systems and to observe novel complex phenomena.

Acknowledgement

"You raise me up, so I can stand on mountains;
You raise me up, to walk on stormy seas;
I am strong, when I am on your shoulders;
You raise me up... To more than I can be."
(Brendan Graham/Rolf Lovland, 2002)

It is my great pleasure to have this opportunity for showing my appreciation to all people who supported me during my PhD course in University of Catania.

Foremost, deep gratitude goes to my supervisor Prof. Luigi Fortuna, without him, I could not have come and studied here. I wish to express my thanks to him for his invaluable supports, supervisions, giving me guidance and valuable advice. Without any of these, it would not have been possible to complete this PhD thesis.

I would like to show special thanks my co–supervisor Prof. Mattia Frasca. His constructive feedbacks, comments and suggestions at ev-

ery stages encouraged me to pursue and complete interesting challenge topics.

I also want to thank Prof. Thang Manh Hoang who gives me prestigious suggestions during my visiting period at SIPLab, HUST.

I would like to thank all members of Automatic Control Laboratory, DIEEI, University of Catania. I have learned a lot from them through the memorial three–year period. The warm and continuous support they have given me since I came to Catania is the thing I will not forget. I am really appreciated for what they have done for me.

Finally, but not mean the least, I would like to thank members in my family for their attention and loving care.

Contents

1

Introduction

This chapter briefly presents the background of the three nonlinear systems which constitute the framework in which our new nonlinear circuits have been built and new phenomena have been observed. After that, objectives of the thesis are introduced. Finally we introduce an overview of the content and the contribution of this thesis.

1.1 Problem statements

There are amounts of phenomena that could be described as nonlinear systems. There, firstly we review three widely–used systems in the literature: time–delay system, Cellular Neural Network (CNN) and memristive system.

The systems with time–delays have been observed in various fields from physical systems, biological systems to engineering systems. In addition, the use of delay differential equations to model these systems has a long history. On one hand, with the presence of time–delay, the

dynamics of time–delay systems are more complex than ones without delays. Using this feature, chaos can be created by using a system describing by only one delay differential equation. Applications of the time–delay chaotic systems can be found in random generator, secure communication, modelling complex system and so on. On the other hand, the presence of time–delay is able to reduce the dynamics of systems. Noticeable application is time–delay feedback control which can stabilize the chaotic behavior to one of unstable fixed points (UFPs) or unstable periodic orbits (UPOs) embedded within chaotic attractor. Time–delay feedback control has become one of the most popular methods applied successfully in a variety of biological systems, electrical circuits, lasers and magneto–elastic systems. For these reasons, design of autonomous time–delay chaotic circuits has received a significant amount of attention, the authors of many works on this subject however often built separate circumstances by changing the nonlinearities. Thus, a general design procedure is required. Moreover, although a network of T–type LCL filters was often used as the delay unit in recorded circuits, the drawbacks of parasitic resistive effects of inductances and large size make the difficulties in design progress. As the result, finding other ways to implement precise time–delay units, for example operational amplifier–based circuitries or solutions based on a digital circuitry, are open problems.

CNN is a nonlinear system which combines the advanced features of neural networks and cellular automata, hence its architecture is suitable to VLSI implementation. Applications of CNN grow continuously,

for instance in fields like signal processing, pattern formation, or modelling of complex systems. There are a lot of generalizations of CNN architectures such as State Controlled–CNN, discrete–time CNN, universal machine CNN introduced just to this aim etc. . However, finding new effective architectures is still an attractive area.

Although memristor and memristive systems have been introduced a long time ago by Chua, applications of them have developed recently after the invention of the nano–scale HP memristor. As the result, the discovery of novel memristive systems, and especially chaotic ones, is an interesting topic. While some implemented memristive circuits have exhibited chaotic behaviors, the question if it is possible to find simpler memristive chaotic circuits, i.e., 2–element circuit including a memristor and a capacitor, because of the rich of dynamics of memristor is open and very interesting.

In addition, noise can effect and modify the dynamics of original systems. Hence estimating the influence of noise on nonlinear systems, in particular novel systems or complex networks, is a very important area of research which is still ongoing.

1.2 Objectives

The aim of our thesis is to study nonlinear circuits built starting from devices or architectures with special features. Models of new systems are proposed and analysed. Then theoretical models are confirmed by numerical simulations and circuital experiments using off–the–shelf

analog components or Field Programmable Gate Array. Because of the rich variety of dynamics, these systems are able to be used in various applications like chaotic circuits, autowave generators or investigation of other complex phenomena. Starting from the conventional systems such as time–delay system, CNN, and memristor system, novel systems are discovered by two approaches. Changing or improving conventional systems is the method pursued in the first approach, for instance, we investigate the general methodology to design time–delay continuous–time chaotic circuit or the procedure to realize chaotic circuits with a digital time–delay block. Combining conventional systems into new ones is the main idea of the second approach, for example, the thesis focuses on studying the characteristics of memristive CNN and memristive time–delay systems. Furthermore, the presences of noise in chaotic systems are also examined.

1.3 Contributions and contents

The thesis presents the obtained results relating to the novel nonlinear circuits. The structure of the thesis is illustrated in Fig. 1.1.

Chapter 2 reviews time–delay systems based on first–order delay differential equation. We summarize the nonlinearities and time-delay blocks which were employed in the literature. The design procedure for time–delay chaotic circuits is also repeated to provide a general way for construction of different time–delay circuits in the next chapter.

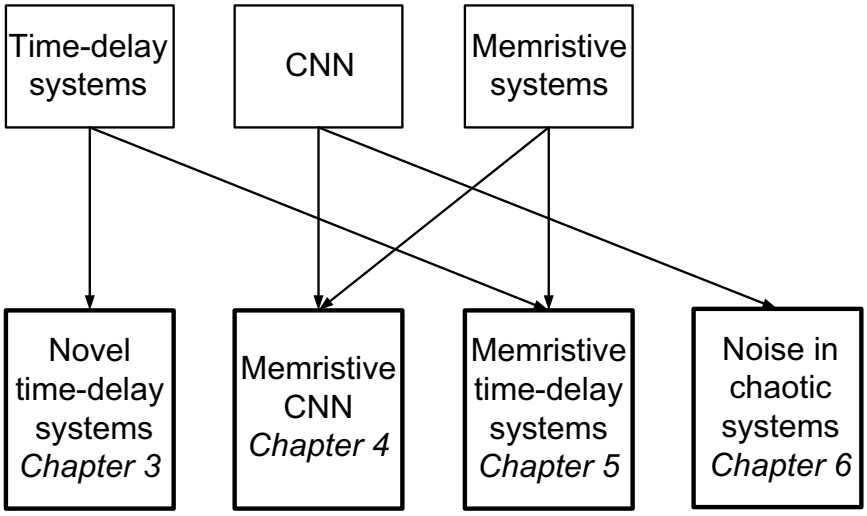


Fig. 1.1: Structure of the thesis which shows relations between three conventional nonlinear systems and our novel studies.

A digital approach for the implementation of time–delay blocks in chaotic circuits is discussed in Chapter 3. Because of the presence of an ADC and a DAC in the scheme, the effects of these devices are also analysed. Furthermore a general method to evaluate the effects of these parameters and/or other sources of errors or parametric differences between implementation and model are found.

CNN based on memristive cells consisting of three–component circuits is introduced in Chapter 4. Due to the fact that each cell is connected with four neighbours by four linear resistors, a diffusion–reaction

process is created. The simulations and experimental results show that our memristive CNN is able to generate autowaves.

In Chapter 5, some simple memristive time–delay systems are proposed. It is worth noting that chaos can be observed in very simple circuit configurations, for example in the 2–element circuit.

Finally, results relating to effects of noise on chaotic systems are presented in Chapter 6. There are two systems which have experimentally investigated: a chaotic system with only one stable equilibrium and a network of four Chua's circuits.

2

Design of time–delay chaotic electronic circuits

This chapter summarises nonlinear blocks and delay blocks which were used to construct time–delay chaotic circuits in the literature. We also review the approach allows us to design and implement a new class of time–delay chaotic circuits with simple components, like resistors, capacitors, and operational amplifiers.

2.1 Introduction

Dynamical systems with time delays are observed in various fields [118]. The presence of time delay is unavoidable because of the limit calculation speed, memory effects, finite transmission velocity etc.

On one hand, delay can be utilized to stabilize chaotic systems. Time–delay feedback control (TDFC) [83] is a highly effective control method which has applied in various systems [66, 52, 100]. The most advantage of this method is that it does not require the prior knowledge about the model. Different to the conventional feedback control,

the feedback signal is proportional to the difference of output signal and its delayed version. TDFC can stabilize the chaotic behavior to one of unstable fixed points (UFPs) or unstable periodic orbits (UPOs) embedded within chaotic attractor. There are two control parameters: the feedback gain and the feedback time delay. It is worth noticing that the feedback time–delay is often different from the intrinsic delay of time–delayed system. Two parameters have been selected by trial–and–error procedures or Lyapunov stability analysis approaches. In particular, for stabilizing of the UPO, the controller time delay has to be chosen as an integer multiple of the period of the desired UPO.

On the other hand, delay could make complex behaviors which do not exist in original systems. For example, the systems described by first order delay differential equations (DDE) can exhibit chaos [65]. These systems have attracted more attention because of their complex chaotic attractors as well as their feasibilities. Differential kinds of time–delay chaotic systems [84, 73, 125, 97, 98] have been investigated with the same structure as shown in Fig. 2.1, containing nonlinear block, delay block, and a RC filter. Observe from Fig. 2.1 that nonlinear unit has played a very important role in making diverse features of designed systems. Moreover, the normal approaches to realize delay block still faces some problems. For example, the implementation based on T–type LCL filters has the drawback of parasitic resistive effects of inductances which can not be neglected since they produce a strong attenuation of the delayed signal. The aim of this chapter is to summarize studied time–delay chaotic systems, concentrating on nonlinearity

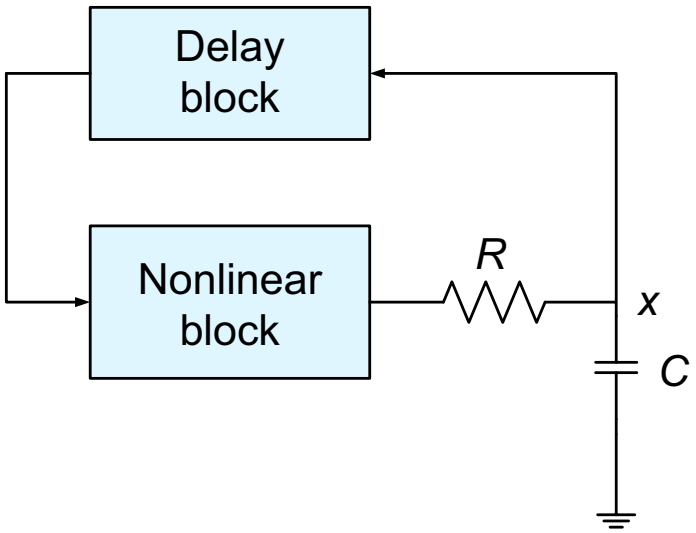


Fig. 2.1: General architecture of the time–delay chaotic circuit.

block and time–delay block, and to review a new procedure to design time–delay chaotic circuits. That provides the background to support the next chapter where novel time–delay chaotic circuits are presented.

2.2 Nonlinear block

In the literature, time–delay chaotic oscillator is often presented by a DDE

$$x\left(t\right) = -ax\left(t\right) + bF\left(x\left(t-\tau\right)\right), \tag{2.1}$$

where a, b are parameters, τ is the time delay and $F(x)$ is a nonlinear function. Authors focused on constructing suitable nonlinear blocks emulating $F(x)$. Some typical nonlinear functions are reported as follows.

Mackey–Glass model [66] for haematologic disorders was given by

$$F(x) = \frac{Ax}{1 + x^{10}}, \tag{2.2}$$

with $A = 10$. This function was produced by coupling in a special way two complementary junction field–effect transistors [73].

Voss model [110] used a single–humped smooth nonlinearity

$$F(x) = A_1 x^3 + A_2 x^2 + A_3 x + A_4, \tag{2.3}$$

with $A_1 = -10.44$, $A_2 = -13.95$, $A_3 = -3.63$, and $A_4 = 0.85$. The nonlinearity was built in essence from a transistor and four resistors [51].

In Ucar's works [106, 107], he developed a model containing cubic nonlineariry

$$F(x) = Ax + Bx^3, \tag{2.4}$$

with $A = 1$ and $B = -1$. Obviously, the equation (2.4) is simpler than Eq. (2.3). Two different methods [69] were tested in order to design the cubic nonlinearity, using log–antilog operational amplifiers and analog multipliers.

Banerjee [11] defined a nonlinearity with the function

$$F(x) = 0.5A(|x| + x) + B\tanh(Cx), \tag{2.5}$$

where $A = 1.15$, $B = 0.97$ and $C = 2.19$. This function was realized easily by a half–wave rectifier and an operational amplifier.

In the neural model of Duan [38], a non–monotonously increasing transfer function was employed as activation function

$$F(x) = \sum_{i=1}^{2} A_i \left[\tanh(x + B_i) - \tanh(x - B_i)\right], \tag{2.6}$$

where A_1, $A_2 = 1.5$, $B_1 = 1$, and $B_2 = 4/3$. The active function was implemented as a linear combinations of four translated hyperbolic tangent functions obtained by dual–transistor pairs.

Lu and He [65] introduced an odd piecewise linear function as

$$F(x) = \begin{cases} 0 & x \leq -\frac{4}{3} \\ Ax - 2 & -\frac{4}{3} < x \leq -0.8 \\ Bx & -0.8 < x \leq 0.8 \\ Ax + 2 & 0.8 < x \leq \frac{4}{3} \\ 0 & x > \frac{4}{3}, \end{cases} \tag{2.7}$$

with $A = -1.5$ and $B = 1$. Similar to Mackey–Glass nonlinear function, it saturates to zero at large $|x|$.

Another three–segment nonlinear function, which were obtained from two diodes, and an operational amplifier, has been studied [103]. It was described as

$$F(x) = \begin{cases} B(x + 1) + A & x < -1 \\ Ax & -1 \leq x \leq 1 \\ B(x - 1) + A & x > 1, \end{cases} \tag{2.8}$$

where $A = 2.15$ and $B = -4.3$. In contrast to the previous function (2.7), it does not saturate to zero at large $|x|$.

In order to increase the complex dynamics behaviours of chaotic oscillators, Wang [111] considered the activation function took a reflection symmetric piecewise linear function of the form

$$F(x) = Ax + 0.5(A-B)\left[(|x+m| - |x-m|) - (|x+n| - |x-n|)\right], \tag{2.9}$$

where $A = 4.3$, $B = -5.8$, $m = 1.1$ and $n = 3.3$. The nonlinear function could be built conveniently by operational amplifiers and constant voltage sources.

Interestingly, a piecewise linear function with a threshold controller [97] was applied in the following form

$$F(x) = AF^* - Bx, \tag{2.10}$$

where

$$F^* = \begin{cases} -x^* & x < -x^* \\ x & -x^* \le x \le x^* \\ x^* & x > x^*, \end{cases} \tag{2.11}$$

here $A = 5.2$, $B = 3.5$ and $x^* = 0.7$ is the controllable threshold value. The nonlinearity was implemented by using only two diodes and few operational amplifiers.

Moreover, first–order DDE capable of creating multiscroll chaotic attractors was researched [119, 56]. Yalcin [119] proposed a nonlinearity based on a hard limited function

$$F(x) = \sum_{i=1}^{M_x} g_{(-2i+1)/2}(x) + \sum_{i=1}^{N_x} g_{(-2i+1)/2}(x), \tag{2.12}$$

where

$$g_\theta(\zeta) = \begin{cases} 1 & \zeta \ge \theta,\ \theta > 0 \\ 0 & \zeta < \theta,\ \theta > 0 \\ 0 & \zeta \ge \theta,\ \theta < 0 \\ 1 & \zeta < \theta,\ \theta < 0, \end{cases} \tag{2.13}$$

The system exhibited n–scroll chaotic attractor for suitable values of M_x and N_x. For example, three–, four–, five–, and six–scroll attractors obtained when $\{M_x = 1, N_x = 1\}$, $\{M_x = 1, N_x = 2\}$, $\{M_x = 0, N_x = 4\}$, and $\{M_x = 1, N_x = 4\}$, respectively. Nonlinear block comprised voltage comparators whose total number depending on the number of scrolls.

Recently, Kilinc [56] represented a oscillator employed the nonlinearity as a function of hysteresis series

$$F(x) = \sum_{i=1}^{N} (h(x - 2i) - i) + \sum_{j=0}^{M} (h(x + 2j) + j), \tag{2.14}$$

where $h(x)$ is the basic hysteresis function

$$h(x) = \begin{cases} -1 & x < 0.5 \\ 1 & x > -0.5. \end{cases} \tag{2.15}$$

Similar to the previous system (2.12), by choosing the appropriate values of M and N, $N + M + 2$ scrolls could be generated. For instance, the three–, four–, and five–scroll attractors could be observed when $\{N = 0, M = 1\}$, $\{N = 0, M = 2\}$, and $\{N = 0, M = 3\}$, respectively. Here hysteresis comparators were realized employing positive feedback around the classical operational amplifiers.

2.3 Time–delay block

In the reported low–frequency time–delay chaotic circuits, the typical values of the delay were relatively large (in the order of magnitude of

milliseconds). Therefore, time–delay blocks were implemented by delay lines or analog circuits.

A bucket brigade delay–line device (MN3011 integrated circuit with 3328 stages) were employed [110, 51, 60] as illustrated in Fig. 2.2. The delay time may be varied by changing the clock frequency which is provided by a function generator or a special integrated circuit MN3101 that is ideally suited for driving MN3011. In fact, due to the fact that the voltages of the chaotic circuits are not within the input range of MN3011, it required additional subparts to adapt voltages. Furthermore, as the delay–line has a high frequency switch operation, a low–pass filter is required to remove the high frequency noise from the delayed voltage.

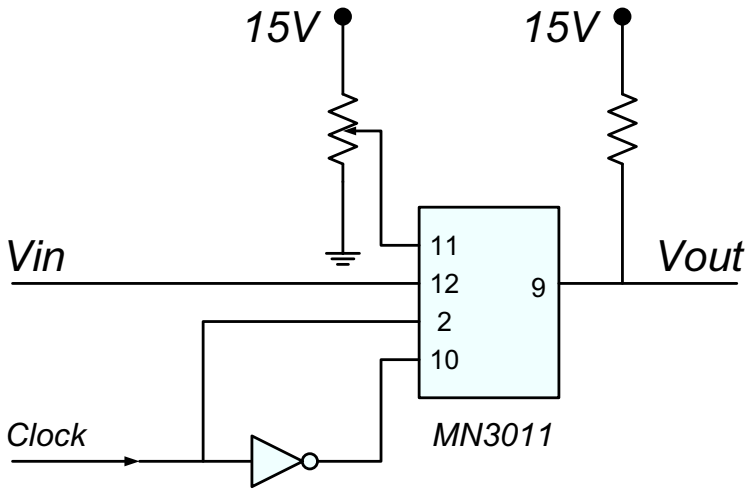


Fig. 2.2: General architecture of the delay bock based on a bucket brigade delay–line device MN3011.

A delay block consisting of T–type LCL filters (see Fig. 2.3) was often used. This approach was introduced firstly by Namajunas [73] with matching resister at the input and output. The delay can be change by increasing or decreasing the number of LCL filters. The total delay can be calculated approximately as

$$T_{delay} = n\sqrt{2LC}, \tag{2.16}$$

if n filters are utilized. Because the delay unit is frequency dependent, the cutoff frequency is only $3kHz$.

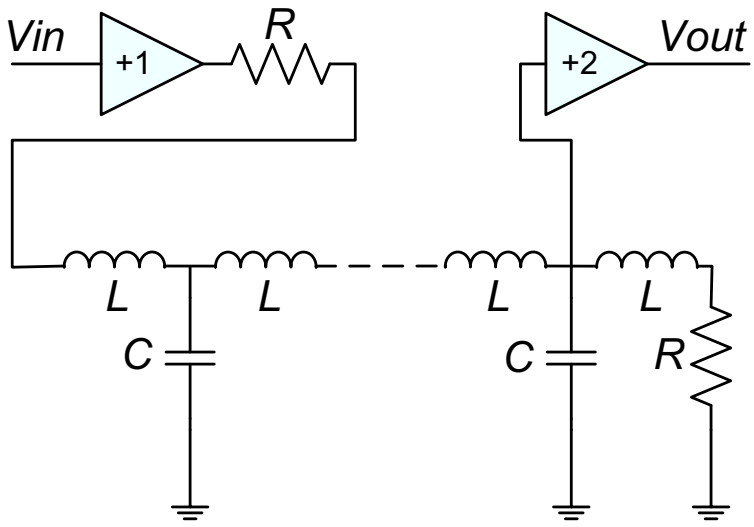


Fig. 2.3: Delay block based on T–type LCL filter with $R = 190\Omega$, $L = 9.5mH$ and $C = 525nF$ [73].

The tuning delay block can be implemented by first order all pass filter [11] as displayed in Fig. 2.4. The transfer function of an all pass filter has the following form

$$H(s) = -a_1 \frac{s - \omega_0}{s + \omega_0}, \tag{2.17}$$

where the flat gain $a_1 = 1$ is determined by R_1, R_2 and the frequency at which the phase shift is $\pi/2$ is calculated as $\omega_0 = \frac{1}{RC}$. In this case, the output power level does not depend upon input signal frequency; however, the phase depends on the input signal frequency. Because each all pass filter contributes with a delay of value equal to RC, the total delay of the delay block is

$$T_{delay} = nRC, \tag{2.18}$$

where n is the number of all pass filters. It is clear to see that by changing the value of resistor R, the amount of delay can vary, as the results the resolution of the delay block can be controlled easily.

It is possible to implement the time–delay by using a cascade of n low–pass second–order Bessel filters [21]. Each filter is characterized by the Sallen–Key topology [89] as shown in Fig. 2.5 and by the following transfer function:

$$H(s) = \frac{1}{1 + C_1(R_1 + R_2)s + C_1C_2R_1R_2s^2} \tag{2.19}$$

The values of the filter components have been chosen in order to realize a Bessel filter with $3dB$ frequency equal to $f_c \simeq 1kHz$ and taking into account off–the–shelf component values. The time–delay introduced by this filter in the band up to f_c can be calculated as $\tau_i = -\frac{d\Phi(\omega)}{d\omega}$. In the $3dB$ band, $\tau_i \simeq C_1(R_1 + R_2)$. For the values of the

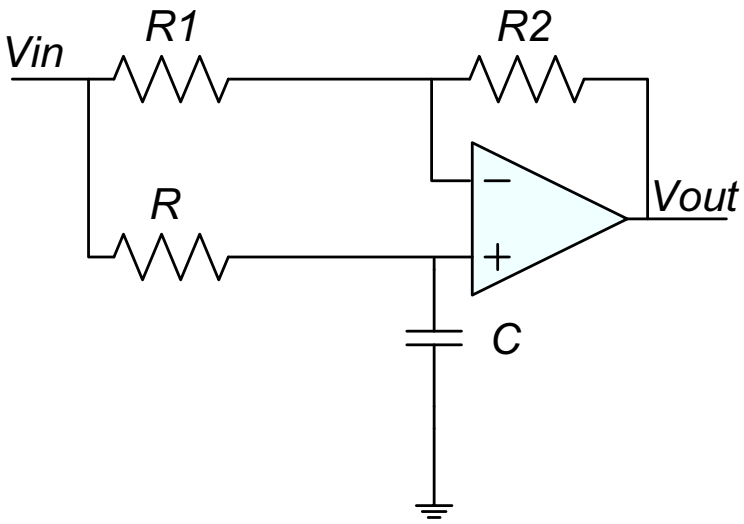


Fig. 2.4: First–order all pass filter with $R_1 = R_2 = 2.2k\Omega$ and $C = 10nF$ [11].

components in Fig. 2.5 $\tau_i \simeq 0.2ms$. Larger delays are realized by taking into account a cascade of n filters.

The approximation of an ideal delay through a cascade of multiple second–order filters or all pass filters allowed to design an efficient and simple circuitry which avoids the drawback of usual delay devices, like delay lines and LCL T–type filters. Furthermore, the filter realization with operational amplifiers, resistors and capacitors is more compact compared to LCL filters as inductors are bulky for large time–delays, although power dissipation is larger.

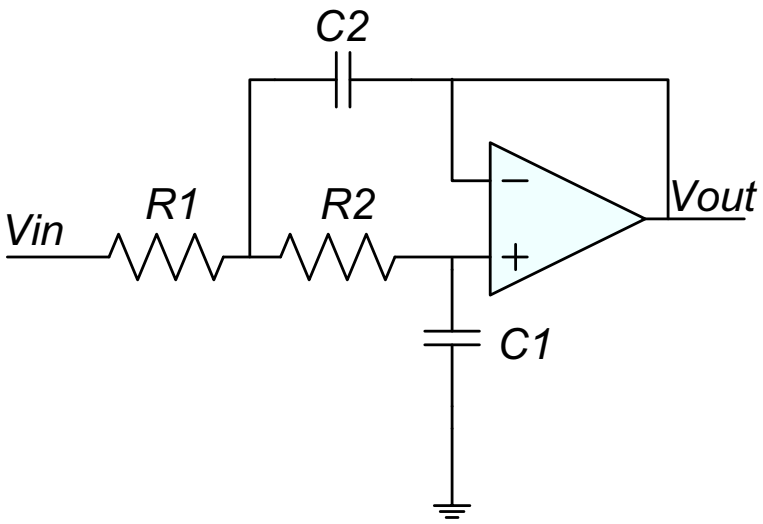


Fig. 2.5: Schematic of the Sallen–Key low–pass active filter implementing a low–pass Bessel filter, here $R_1 = R_2 = 10$ kΩ, $C_1 = 10nF$ and $C_2 = 22nF$.

2.4 Design procedure for time–delay chaotic system

In this Section, we summarize the procedure to design time–delay chaotic systems using only off–the–shelf common components [21]. The core is the feedback scheme shown in Fig. 2.1 is investigated. It represents a simple feedback scheme of a nonlinear system containing all the elements strictly needed to have the possibility that chaotic behavior emerges. It consists of three blocks: a nonlinearity, a RC circuit, and a time–delay block. It represents a minimal configuration to observe chaotic dynamics in autonomous time–delay circuits. The nonlinearity is needed since chaos is a prerogative of nonlinear circuits; the RC

circuit implements the dynamics of the single state variable; and the presence of the delay makes the system infinite–dimensional, allowing chaos to be observed in a system with a single state variable.

In terms of dimensionless equations, the dynamics of the system shown in Fig. 2.1 can be expressed as follows:

$$\dot{x}(t) = k(-ax(t) - bh(x(t-\tau))) \tag{2.20}$$

where $x(t) \in \mathbb{R}$ is the circuit state variable, $h(x) : \mathbb{R} \to \mathbb{R}$ is the nonlinear function, $\tau \in \mathbb{R}^+$ is the time–delay, k is a scaling factor, and a and b are system parameters. b represents the gain multiplying the nonlinearity, while ka is the pole of the RC circuit. System (2.20) represents a nonlinear system in Lur'e form with dynamical linear part given by $L(s) = \frac{ke^{-s\tau}}{s+ka}$ and feedback nonlinear part given by $\mathcal{N} = bh(x)$.

For systems in Lur'e form, it is possible to analytically derive approximate conditions for the existence of chaotic behavior. They are summarized in the following criterion [45]:

Criterion 2.1 (For the Existence of Chaos in Lur'e Systems): The conditions required for the existence of chaos in a Lur'e system are:

1. existence of a stable predicted limit cycle;
2. existence of a separate unstable equilibrium point;
3. interaction between the unstable equilibrium point and the stable predicted limit cycle.

Additionally, it is required that the linear part has filtering properties, which in our case is always satisfied, since $L(s) = \frac{ke^{-s\tau}}{s+ka}$.

The existence of a stable predicted limit cycle is derived by applying the harmonic balance method [45]. A limit cycle solution of the type $y_0(t) = A\sin(\omega t) + B$ exists if the amplitude A and the bias B of y_0 satisfy the following equations:

$$\begin{aligned} 1 + L(0)N_0(A, B) &= 0 \\ 1 + L(j\omega)N_1(A, B) &= 0 \end{aligned} \tag{2.21}$$

where N_0 and N_1 are static and dynamic describing functions [9] approximations of the nonlinearity $\mathcal{N}$. Solving Eqs. (2.21) means to express A as a function of B, i.e., $A(B)$ from the static equation, and, then, to consider the intersections between the curve $L(j\omega)$ and the curve $-1/N_1(A(B), B)$ in the complex plane. Each intersection corresponds to a limit cycle with a given frequency and amplitude.

The stability properties of the predicted limit cycle y_0 are inferred by applying the limit cycle criterion [93]:

Criterion 2.2 (Limit cycle criterion) Each intersection point of the curve $L(j\omega)$ and the curve $-1/N_1(A(B), B)$ corresponds to a limit cycle. If the points near the intersection, corresponding to the considered limit cycle, along the curve $-1/N_1(A(B), B)$ for increasing values of B are not encircled by the curve $L(j\omega)$, then the limit cycle is stable. Otherwise the limit cycle is unstable.

Let focus on the second condition of criterion 2.1. The equilibrium points $x(t) = x(t - \tau) = cost = \bar{x}$ of the system can be calculated by solving:

$$a\bar{x} + bh(\bar{x}) = 0 \tag{2.22}$$

The criterion for the existence of chaos in Lur'e systems requires the existence of an equilibrium point which has to be separate from the limit cycle and unstable. We now briefly discuss how the stability properties of the equilibrium points can be checked. Since in the following piece–wise linear (PWL) nonlinearities will be used, the stability of the generic equilibrium point E_i can be analyzed taking into account the following system:

$$\dot{x}(t) = k(-ax(t) - b'x(t-\tau) + u) \tag{2.23}$$

which represents the dynamics of the system (2.20) in the PWL region to which the equilibrium point under examination belongs, i.e., $b' = bm_i$ where m_i is the slope of the PWL nonlinearity in the region of the equilibrium point E_i. u is a generic constant input. Alternatively, if $h(x)$ is not a PWL function, linearization around E_i could be considered.

For system (2.23) two different types of asymptotic stability can be defined [74]: delay–independent (if the system (2.23) is stable for all the values of $\tau \in R^+$) and delay–dependent (if the system (2.23) is stable for some values of τ and unstable for other values of τ) stability. To check them, the following criteria [74] can be applied.

Criterion 2.3 (Delay–independent stability): The equilibrium point E_i is delay–independent stable if and only if $a + b > 0$ and $a \geq |b|$.

Criterion 2.4 (Delay–dependent stability): The equilibrium point E_i is delay–dependent stable if and only if $b > |a|$. The equilibrium point E_i is stable for $\tau \leq \tau^* = \frac{\cos^{-1}(-a/b)}{k\sqrt{(b^2-a^2)}}$.

Finally, the third condition of criterion 2.1, i.e., the interaction between the stable predicted limit cycle and the unstable equilibrium point E, can be expressed as follows:

$$A \geq |E - B| \tag{2.24}$$

The system represented in Fig. 2.1 thus exhibits chaotic behavior if conditions 1)–3) of criterion 2.1 are satisfied, i.e., if Eqs. (2.21) and (2.24) admit a solution and a separate equilibrium point satisfying neither criterion 2.3 nor criterion 2.4 exists. These conditions can be satisfied by a proper choice of the nonlinearity $h(x)$ and of the parameters a, b and τ of system (2.20).

The time–delay τ plays an important role in conditions 1)–3) of the criterion 2.1. τ may change the stability properties of equilibrium points (if they are delay-dependent stable) and may also affect the Nyquist diagram of $L(s)$ (and thus the conditions on the existence of the limit cycle and on its interaction with the equilibrium point). In general, when criterion 2.1 admits a solution, for a given nonlinearity and for fixed values of a and b, one obtains a range of values of τ, i.e., $\tau_{min} \leq \tau \leq \tau_{max}$ for which the system exhibits a chaotic behavior. Once fixed τ in this range, i.e., $\tau = \bar{\tau}$, the problem of the implementation of a circuitry introducing this delay arises. Taking into account the typical scaling factor k in discrete–components circuit implementations, this delay is in the order of magnitude of milliseconds and thus its implementation requires the definition of an appropriate strategy.

The idea underlying the approach is to implement the time–delay block with a cascade of n Bessel filters which are low-pass filters with a maximally flat magnitude and a maximally linear phase response [35]. Since a Bessel filter with transfer function $H(s)$ introduces a time–delay up to the $3dB$ frequency equal to $\tau_i = -\frac{d\Phi(\omega)}{d\omega}$, a given delay $\bar{\tau}$ can be obtained with n blocks in cascade according to:

$$\bar{\tau} \simeq n\tau_i \tag{2.25}$$

In this way, the time–delay can be easily tuned by changing the number n of filters in cascade.

The Bessel filter implemented is an *uncertain* Bessel filter, since its implementation is based on off–the–shelf components with standard values, which additionally are subjected to tolerance. For this reason, it is important to evaluate the error between the model and the circuit implemented with a given n, and, if necessary, use a different number of filters in cascade. Since the implemented system is chaotic and thus sensitive to parameters, the introduced approach relies on the definition of an error measure taking into account this property.

Let consider two chaotic circuits coupled through a master–slave configuration [55]. If the two circuits are identical, in general, it is possible to synchronize them, obtaining state variables which asymptotically follow the same trajectory. In the case of non–identical circuits, complete synchronization cannot be obtained, but the synchronization error is kept small if the circuits have similar parameters. Therefore,

the synchronization error can be used to evaluate the accuracy of the approximation based on Bessel filters.

More in detail, let consider

$$\begin{aligned} \dot{x}(t) &= k(-ax(t) - bh(x(t-\tau))) \\ \dot{x}_n(t) &= k(-ax_n(t) - bh(\widehat{x}(t)) + \kappa(x(t) - x_n(t))) \end{aligned} \tag{2.26}$$

where $x(t)$ is the state variable of the ideal model (2.20), i.e., with $L(s) = \frac{ke^{-s\tau}}{s+ka}$, $x_n(t)$ is the corresponding state variable of the approximated model, i.e., with $\widetilde{L}(s) = \frac{k}{s+ka}H^n(s)$, κ is the coupling strength, and $\widehat{x}(t)$ is the output of the n Bessel filters in cascade, which ideally is $\widehat{x}(t) = x_n(t-\tau)$.

Then let define the synchronization error between the ideal model (with time–delay τ) and the approximated model (with a time–delay given by n filters in cascade) as follows

$$\delta(n) = \langle |x(t) - x_n(t)| \rangle \tag{2.27}$$

where $\langle \cdot \rangle$ represents the average with respect to time. The synchronization error (2.27) is used to evaluate the accuracy of the approximated model. Moreover, n is selected to minimize this error.

The accuracy of this procedure was confirmed by experimental examples [21].

3

Implementation of chaotic circuits with a digital time–delay block

In this chapter, we investigate how to implement the time–delay block with a solution based on a digital circuitry. Using available Field Programmable Gate Arrays, a programmable time–delay block can be realized. Effect of this approach to the accuracy of the whole implementation is estimated by the synchronization error.

In some electronic circuits, delay blocks are intentionally introduced to implement specific functions. For instance, in measurement applications, the highly integrated time–multiplexing device [18], the particle detector [90], and the time analog–to–digital converter [76] make use of time–delay blocks. In radio frequency applications, in high–speed microprocessors or in memories, delay circuits play an important role in delay–locked loop [24] where they allows to enhance the performance of the integrated circuits.

As noted in the previous chapter, the presence of time–delays in nonlinear systems may also induce chaotic oscillations. In such systems, the typical values of the delay may also be relatively large. In

order to create time–delay, two main methodologies can be taken into account: analog methods and digital methods. In the former class, the time–delay is implemented by an analog circuit (usually a long chain of elementary blocks) made of resistors, capacitors, inductors, and/or operational amplifiers [73, 110, 85]. However, one of the disadvantages of these approaches is that they require a large number of components. Furthermore, the design or the parameters of the circuit have to be changed if other values of the delay should be implemented.

In order to overcome the drawbacks of analog methods, a few papers have explored digital approaches for the implementation of relatively small time–delays (in the order of magnitude of nanoseconds), e.g., [18, 90, 124].

The aim of this chapter is to investigate the use of time–delay digital blocks for the implementation of chaotic circuits. The effects of some parameters such as the sampling time and the number of bits used in the converters are also investigated.

3.1 Design of chaotic circuits with digital time–delay block

As mentioned in Chapter 2 the model of chaotic oscillator is a simple one with only three blocks connected into a feedback configuration (see Fig. 2.1). From the point of view of the physical realization, the time–delay block implementation is not trivial, because of the typical values of the time–delay used in such chaotic oscillators (order of magnitude

of milliseconds). A digital approach is here applied for the realization of this block, the alternative scheme can be represented as in Fig. 3.1 where an analog–to–digital converter (ADC) and a digital–to–analog converter (DAC) interfacing the time–delay digital block with the remaining (analog) part of the circuit are included. The analog signal is transformed into a digital one by the ADC, then the digital signal from the ADC is delayed in the time–delay block before being fed through the DAC to the remaining part of the circuit. A delayed analog signal is obtained at the output of the DAC. The signal is delayed by a quantity given by:

$$T_{delay} = T_{ADC} + T_d + T_{DAC} \tag{3.1}$$

where T_{ADC}, T_d and T_{DAC} are the delay of the ADC, the delay of the digital time–delay block and that of the DAC, respectively. The total delay can be approximated as $T_{delay} \simeq T_d$ when $T_{ADC}, T_{DAC} \ll T_d$.

The analog to digital conversion introduces some error sources whose effects are also investigated in this chapter. In particular, the effects of the sampling rate and of the number of bits used in the conversion process have been studied, in order to obtain indications on how to select the DAC and ADC devices in the implementation stage.

In general, the synchronization error grows up in the presence of parametric or structural differences, so the synchronization error between the circuit with a digital time–delay and the circuit with an analog time–delay can be used to evaluate the effects of the parameters of the conversion process.

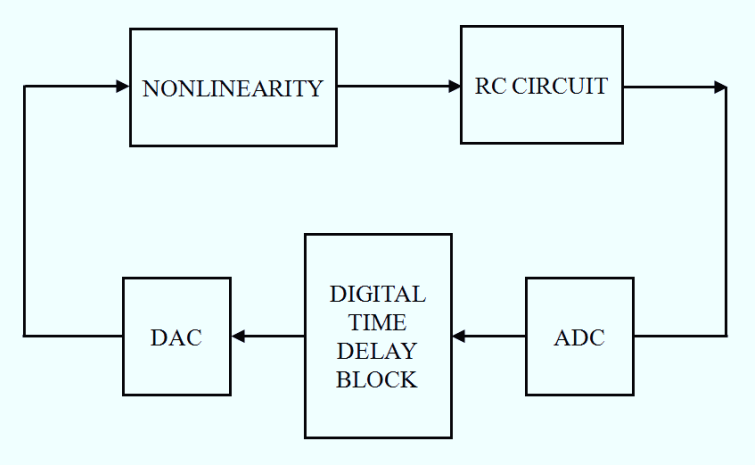


Fig. 3.1: Block scheme of the time–delay chaotic oscillator with a digital time–delay block.

In the case under investigation, larger time–delays are needed, shift register–based approach has been here introduced and applied later to the circuit implementation. Since the output of a D flip–flop holds the value of the input until the next change of the clock, this device can be considered a block with a time–delay equal to the clock duration T_{clk}. A shift register made of N D flip–flops has been used in our approach. On each rising edge of the clock, a new bit is shifted in from the input and all the subsequent contents are shifted forward. The last bit in the shift register is available at the output. According to the configuration of the shift register, there are two parameters controlling the time–delay: the clock duration T_{clk} and the number of shifted bits N_{shift}. The time–delay introduced by a shift register is given by:

$$T_{delay} = T_{clk} N_{shift} \tag{3.2}$$

An advantage of this digital implementation, compared to analog approach for delay implementation, is the possibility to use available programmable hardware to realize the desired configuration and to change its parameters on–the–fly. In the following, we describe one of such implementations, in particular based on an FPGA, and show its suitability for the implementation of chaotic circuits.

3.2 FPGA–based implementation

3.2.1 Implementation of the time–delay

High performance FPGAs containing millions of gates are currently available in the market [70]. FPGAs can be used in almost all applications including communications, digital signal processing applications or systems on chip, because of their advanced features such as embedded microprocessor and digital signal processing cores. An FPGA can be quickly configured to the desired application so that it is very convenient for the research and prototype development phase. For these reasons, in this chapter for the implementation of the time–delay block, an FPGA has been chosen. In this Section, the details of the implementation of the time–delay block are outlined.

The scheme of the time–delay block implemented in the FPGA is illustrated in Fig. 3.2. It consists of one multiplexer and nine time–delay subcircuits connected in cascade (including one subblock implementing a delay of 1 ms, one subblock implementing a delay of 0.4 ms and seven identical subblocks implementing a delay of 0.2 ms). The time–delay

subcircuits utilize shift register configurations as described in Sect. 3.1. The subcircuits are connected to the multiplexer so that a total delay in the range from 1.4 ms to 2.8 ms can be set. The tuning of the time–delay has been made accessible to the user, by implementing a routine, which thanks to control buttons, allows the user to set the multiplexer parameters, thus selecting the desired time–delay.

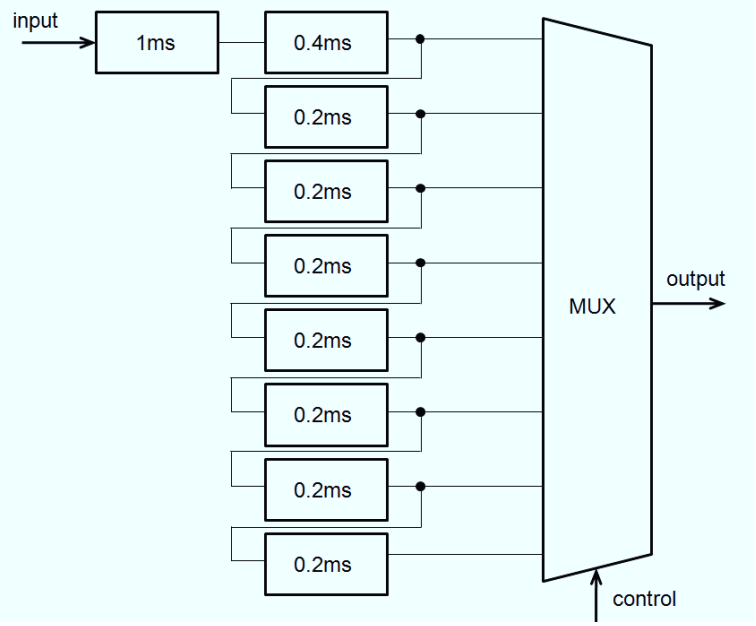


Fig. 3.2: Block diagram of the time–delay block implemented in the FPGA.

The hardware platform used in this work is the ML405 board that is based on the Virtex4 family of FPGAs. The main features of the board are the following: Virtex–4 FPGA XC4VFX20–FF672, 128 MB DDR SDRAM with a 32–bit interface running up to 400 MHz data rate, 100

MHz clock oscillator plus one extra open 3.3 V clock oscillator socket. The delay circuitry prototype has been realized in the Xilinx FPGA chip programmed through a VHDL language. By employing the RTL Viewer tool in Xilinx ISE, the schematic diagram of the delay block could be obtained as in Fig. 3.3. The main resources of Virtex4 used for our application are summarized in Table 3.1.

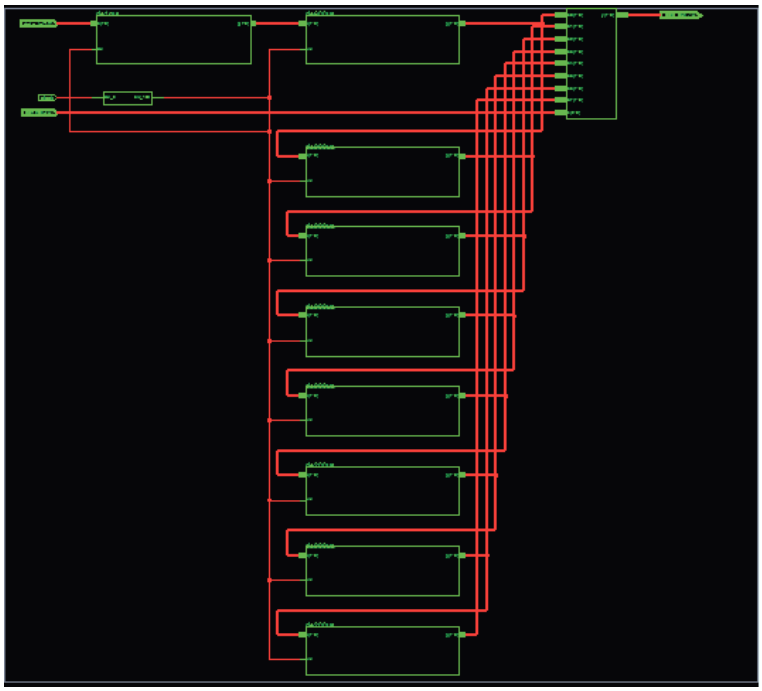


Fig. 3.3: Schematic of the time–delay block displayed by RTL Viewer tool.

It is interesting to note that the time–delay block could be tested by applying directly a digital signal input generated by the signal generator (see Fig. 3.4). Both the input and output signal were displayed on

Table 3.1: Device utilization summary

Logic utilization	**Used**	**Available**	**Utilization**
Number of Slice Flip Flops	442	17,088	2.62%
Number of 4 input LUTs	73	17,088	0.43%

the monitor of the oscilloscope, so it was convenient to see and compare results.

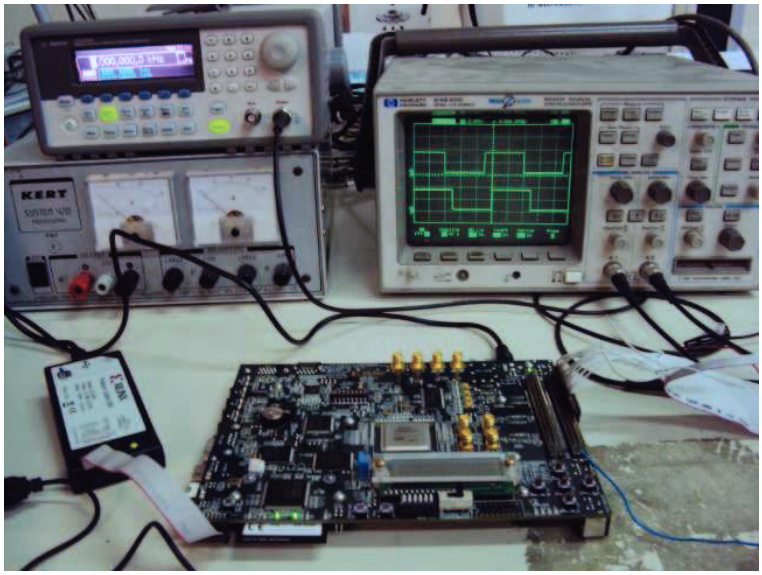


Fig. 3.4: Connection of equipments to test the FPGA–based time–delay block.

3.2.2 Implementation of the chaotic circuit

In this Section, an example of a chaotic circuit obeying (2.1) is discussed. The chaotic circuit is described by the following state equation [52]:

$$\dot{x} = -ax\left(t\right) - b\sin\left(x\left(t-\tau\right)\right), \tag{3.3}$$

where a and b are the parameters and τ is the dimensionless time–delay. Equation (3.3) represents the dynamics of an optical bistable resonator in which x is the lag of the phase of the electric field across the resonator, b is the laser power intensity injected into the system, and τ is the round trip time of the light in the resonator.

In the following, instead of implementing the sinusoidal nonlinearity appearing in (3.3), a piecewise linear (PWL) approximation has been used. PWL functions, in fact, have the advantage of ease of implementation [43].

In particular, the following PWL approximation has been used:

$$\sin x \simeq g\left(x\right) = \begin{cases} -\frac{2}{\pi}x - 2, & x \in \left[-\frac{3\pi}{2}; -\frac{\pi}{2}\right] \\ \frac{2}{\pi}x, & x \in \left[-\frac{\pi}{2}; \frac{\pi}{2}\right] \\ -\frac{2}{\pi}x + 2, & x \in \left[\frac{\pi}{2}; \frac{3\pi}{2}\right] \end{cases} \tag{3.4}$$

The nonlinear in (3.4) is also presented in Fig. 3.5

The complete schematics of the designed time–delay chaotic circuit including the PWL function (3.4), the time–delay block implemented with the FPGA approach described in Sect. 3.2.1 and the RC circuit, is shown in Fig. 3.6. The circuit makes use of TL084 operational amplifiers (U1A, U2B, U3C, and U4D), eight resistors (from R1 to R8) with 5%

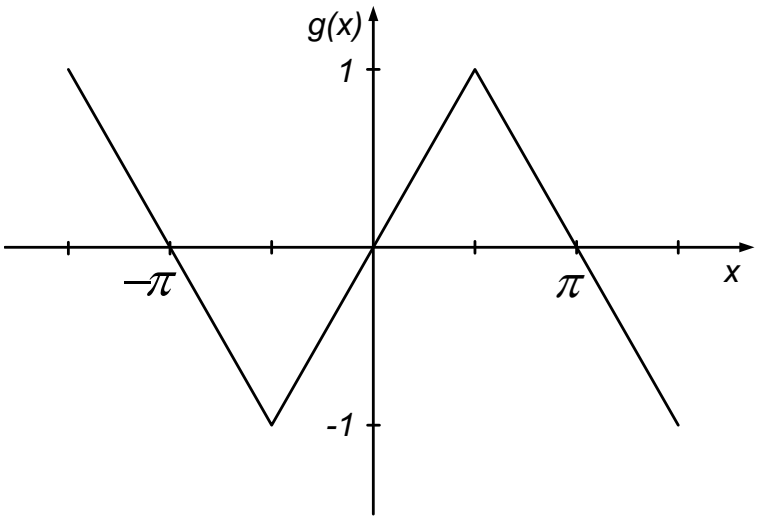


Fig. 3.5: Piece–wise linear function $g(x)$.

tolerance, one capacitor (C1), and the FPGA–based time–delay block.

The first three operational amplifiers are used to build the PWL nonlinearity by letting the operational amplifiers U1A and U2B working in the nonlinear region according to the guidelines described in [43]. The operational amplifier U4D is a buffer. The time–delay block is obtained through the FPGA which implements a time–delay τ given by:

$$\tau = \frac{T_{delay}}{R_8 C_1},$$

where T_{delay} is tuned through control buttons that set the FPGA multiplexer configuration.

By applying the Kirchhoff's laws and taking into account the parameter values reported in Fig. 3.6, the following differential equation

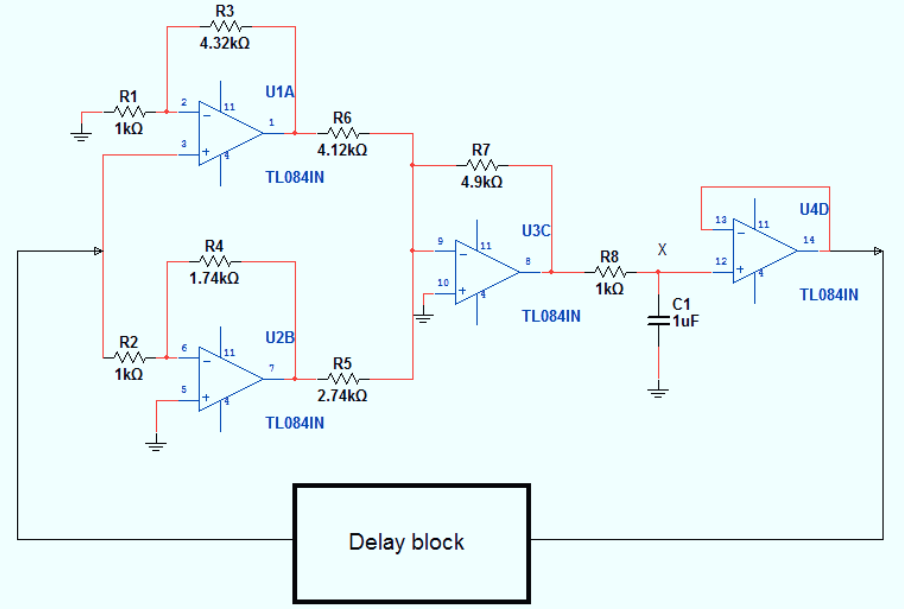


Fig. 3.6: Electrical scheme of the time–delay chaotic circuit introduced in this chapter. The delay block is implemented by means of a FPGA.

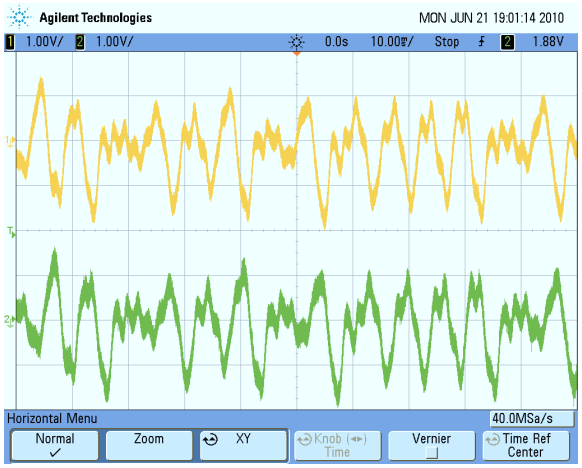
describing the circuit dynamics can be obtained:

$$\frac{dV_{C_1}}{dt} = -aV_{C_1}(t) - bg\left(V_{C_1}(t-\tau)\right)$$

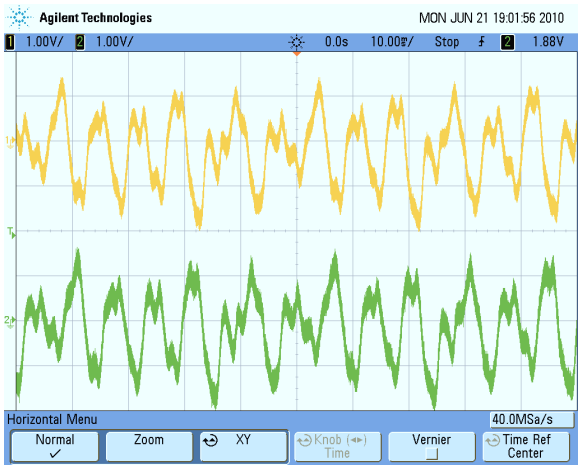
where $a = 1$, $b = 5$ and V_{C_1} is the voltage across the capacitor C_1.

In the range of t from 1.6 ms to 2.8 ms, the circuit displays a chaotic regime, as theoretically expected [22]. Figure 3.7 shows some examples of the waveforms experimentally obtained. These experimental results confirm that the FPGA–based delay block performs effectively for generating chaos.

(a)



(b)



(c)

Fig. 3.7: Waveforms generated by the chaotic circuit of Fig. 3.6 for different values of the time–delay: (a) τ =2.0 ms; (b) τ =2.4 ms; (c) τ =2.8 ms.

3.3 Accuracy of the implementation

As mentioned in Sect. 3.1, the conversion process affects the accuracy of the implementation. The accuracy of the implementation has been investigated by assuming as a reference model an analog implementation based on Bessel filters and described in [22] and investigating how the synchronization error between the analog implementation and the FPGA–based one is affected by the parameters of the conversion processes (the sampling rate and the number of the bits used).

In particular, an unidirectional coupling scheme [19] has been implemented. The first chaotic system, referred as the master, sends a scalar signal (i.e., the state variable x) to the second system, referred as the slave. At the slave, the state variables of the two systems are compared and used to build the error signal $e(t) = x_m(t) - x_s(t)$ which is then fed back to the slave, like in an observer. More in detail, the master equation and the slave equation can be written as:

$$\begin{aligned} \dot{x}_m(t) &= -a_m x_m(t) - b_m \sin\left(x_m(t-\tau)\right) \\ \dot{x}_s(t) &= -a_s x_s(t) - b_s \sin\left(x_s(t-\tau)\right) + K\left(x_m - x_s\right). \end{aligned} \tag{3.5}$$

The synchronization error has been defined as follows:

$$E = \frac{1}{N}\sum_{i=1}^{N} e(t_i) \tag{3.6}$$

where $t_1, t_2, \ldots, t_N$ are the sampling times for each time series and N is the total number of samples.

To evaluate how these parameters affect the synchronization error, both numerical simulations with respect to different parameters of the

conversion process and experiments in different conditions have been performed. As numerical simulations are concerned, two different cases have been taken into account: identical and nonidentical systems. In the case of identical systems, the parameters appearing in (3.3) have been fixed equal for the two circuits $a_m = a_s = a = 1$ and $b_m = b_s = b = 5$. In contrast, the case of nonidentical systems, the parameters have been supposed to be affected by a tolerance in the order of magnitude of 5% of the nominal value, i.e., $a_m = a$, $a_s = a(1 \pm 0.05)$, $b_m = b$, and $b_s = b(1 \pm 0.05)$. As it will be shown later, the introduction of the tolerance in the numerical simulations is needed to obtain a good match with the experimental simulations. In fact, the introduced tolerance is in the range of parameter tolerance of the circuit components used.

The results obtained have been reported in Fig. 3.8 and Fig. 3.9 showing the synchronization error versus the sampling rate and the number of bits, respectively. Figure 3.8 indicates that the sampling rate has to be chosen higher than 100 kHz in order to guarantee a low synchronization error. The synchronization error increases significantly when the sampling rate goes under 100 kHz. As regards the number of bits of the conversion process, when it increases, the synchronization error shows small decreases as illustrated in Fig. 3.9. Therefore, it is not convenient to increase the number of bits beyond 8–10 bits. The synchronization error corresponding to a sampling frequency $f_s = 10$ kHz is always higher than the synchronization error obtained with a sampling frequency $f_s = 200$ kHz, independently of the number of bits used. Based on these considerations, the ADC and DAC used have been

selected with a number of bits equal to 8 and $f_s = 200$ kHz. The results shown in Fig. 3.7 refer to this case.

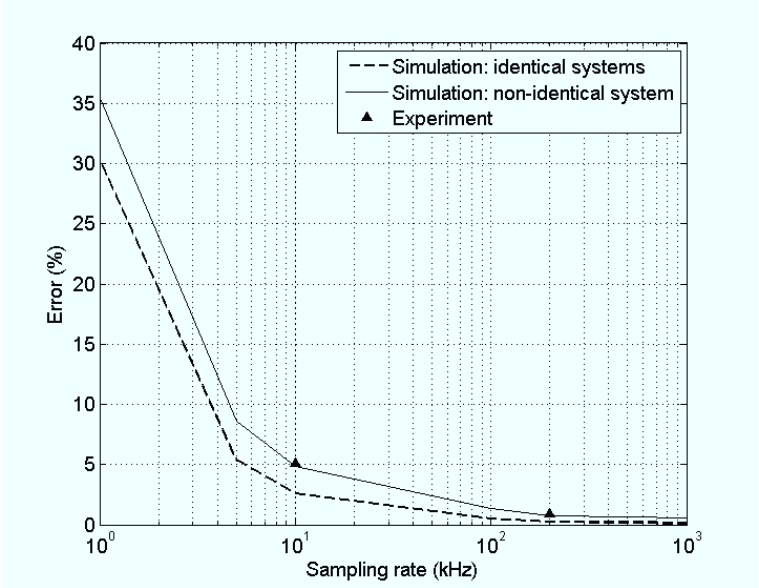


Fig. 3.8: Numerical results: synchronization error (3.6) between model with analog time–delay block and model with digital time–delay block versus sampling rate.

The data in the experiments have been acquired by using a data acquisition board (National Instruments USB–6255) with sampling frequency $f_s = 300$ kHz. As mentioned before, the accuracy of the implementation was evaluated by assuming as reference model a totally analog implementation in which the time–delay block has been realized with Bessel filters [22]. This totally analog implementation has been assumed as the master circuit, while the slave is the FPGA–based

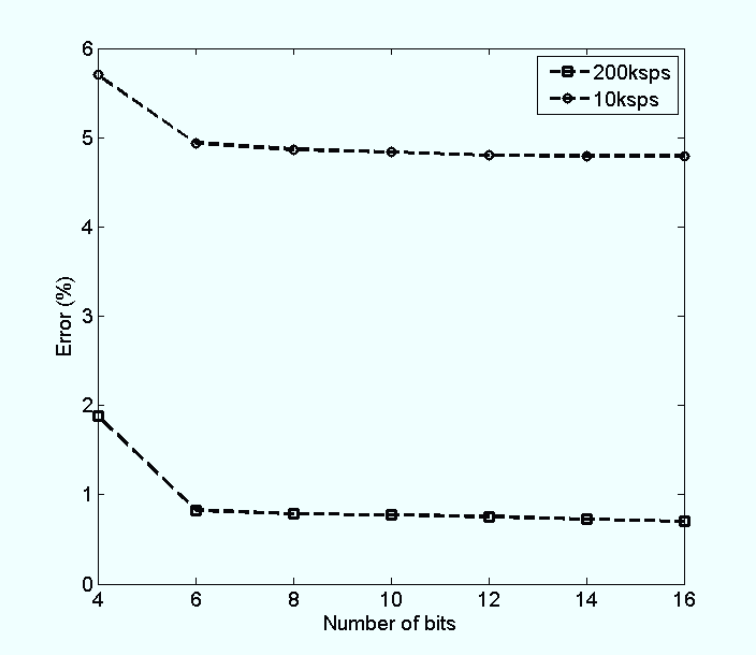


Fig. 3.9: Numerical results: synchronization error (3.6) between model with analog time–delay block and model with digital time–delay block versus number of bits.

time–delay chaotic circuit. Synchronization is observed, as shown in Fig. 3.10(a) when the high speed analog–to–digital converter and high speed digital–to–analog converter (maximum throughput of 200 kHz) are used in the slave circuit. In contrast, when the ADC and DAC with large conversion time (low throughput of 10 kHz) are utilized, a much larger synchronization error occurs as shown in Fig. 3.10(b). It is worth noting that the conversion process parameters, especially the sampling rate, critically affect dynamical features of the circuits which were represented through the synchronization error. The synchronization errors

in Fig. 3.10(a) and Fig. 3.10(b) are equal to $E = 0.8948\%$ and $E = 5.1212\%$, respectively (they have been evaluated taking into account an acquisition time window equal to 1 s). These experimental results well fit the simulation results, in the case of nonidentical systems, i.e., when the parameter tolerance of the circuit components used is taken into account, as it can be observed in Fig. 3.8.

The procedure described to evaluate the effects of the conversion parameters can be applied when a reference model does exist, which is obviously not the general case. Therefore, a more general strategy is here described. The idea is to acquire a long trajectory from the circuit and to use it to synchronize the slave circuit. Instead of the master signal x_m, the acquired signal is used. The experiment, previously described, was repeated by using this approach. First, a circuit with high speed analog–to–digital and digital–to–analog converters has been used. In this case, a low synchronization error ($E = 0.9687\%$) has been observed as shown in Fig. 3.10(c). On the opposite, when a low sampling rate 10 kHz ADCs and DACs is used, the synchronization error ($E = 5.2375\%$) grows up as shown in Fig. 3.10(d). The experimental results obtained confirmed that investigating synchronization by using a signal acquired from the chaotic circuit itself is a suitable strategy for the evaluation of the effects of the conversion parameters (or more in general of other parameters) on the accuracy of the implementation.

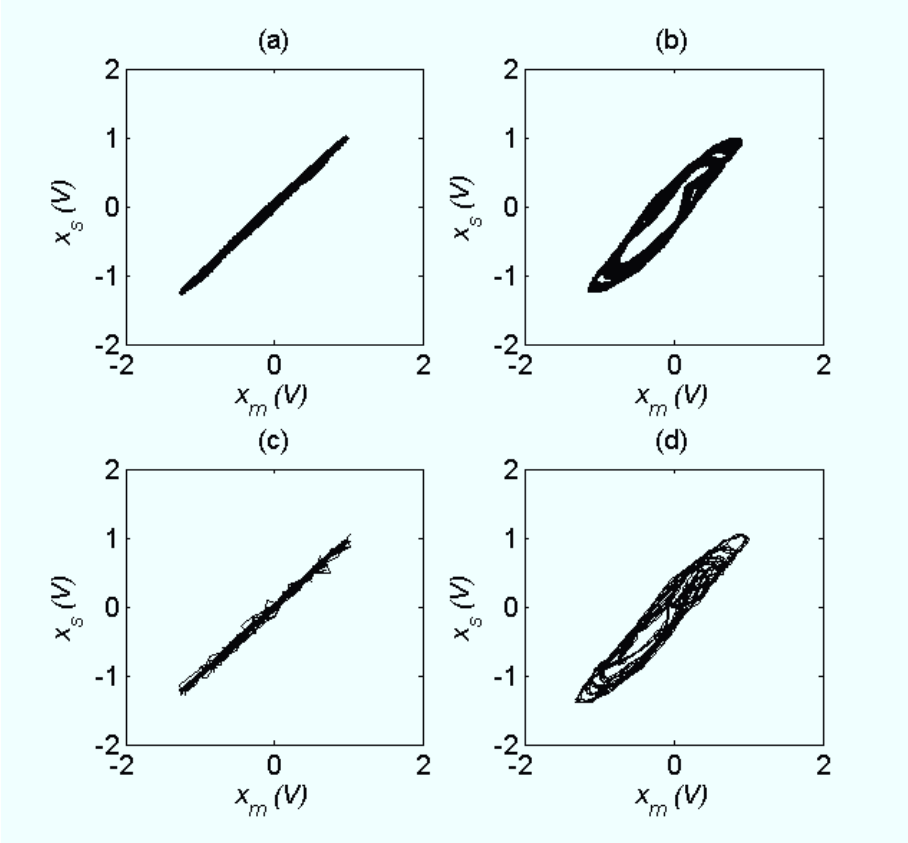


Fig. 3.10: x_s vs x_m for (a) analog master/FPGA–based slave with high speed ADC; (b) analog master/FPGA–based slave with low speed ADC; (c) FPGA–based master with high speed ADC/FPGA–based slave with high speed ADC; (d) FPGA–based master with low speed ADC/FPGA–based slave with low speed ADC.

4

Memristive Cellular Neural/Nonlinear Networks

Memristor has received a significant amount of attention after a solid state implementation of it was realized in the Hewlett–Packard laboratories, because of the potential applications of such device in different areas. In this chapter, we introduce a Cellular Neural/Nonlinear Network based on memristive cells for autowave generation. The basic cell consists of a three–component circuit (the parallel of a capacitor, an inductor and an active memristor) and displays slow–fast dynamics. Such circuit is then connected through passive resistors to other identical cells to form a reaction–diffusion system. Simulation results show that the system is able to generate autowaves and open the way to the study of other complex phenomena like spiral waves or pattern formation in memristive circuits. Moreover, we also implement such a *Memristive Cellular Neural/Nonlinear Network* (MCNN) by using Field Programmable Gate Array (FPGA). Our system consisting of a FPGA development board connected to a monitor allows us

to emulate autowave propagation in an efficient way. Experimental results show the feasibility of FPGA–based approach to implement MCNN.

4.1 Autowaves in Cellular Neural Network

Cellular Neural/Nonlinear Networks (CNNs) were invented to process real–time signals [33], especially image signals [32]. The classical CNN configuration includes a number of cells which consist of linear capacitors, linear resistors, linear and nonlinear voltage–controlled current sources and independent sources. Hence, VLSI implementation of the CNN was realized conveniently with CMOS technique [31]. Because CNNs have the ability to emulate partial differential equations, they can be utilized to simulate complex phenomena in the space such as autowaves [78], [6], spiral waves [77] or Turing patterns [48]. It is interesting to note that chaotic circuits could be also realized by generalized CNN cells [4], [5].

Recently, the nanoscale memristor has been found [99], [105], although the fundamental theory of it was already introduced [27] and generalized [30] some time ago. This discovery promised to apply for biological models [79], adaptive filters [37] or programmable analog integrated circuits [92, 28]. The potential applications of memristive systems lead to one question. How complex systems made by interacting memristor–based circuits can be constructed? There are a few studies focusing on this question. In [54], a cellular automaton and

a discrete–time CNN (DTCNN) using nonlinear passive memristors were designed. One noticeable feature of memristive DTCNN was the multitasking, since memristive DTCNN were shown to be able to perform more than one functions of the memristor cellular automaton at the same time. Another approach was the use of standard CNNs to implement memristive analog circuits which then can be utilized as basic cells to realize chaotic circuits [20]. In another work [61], the role of memristors in implementing programmable connections of the cells was investigated. Following these discoveries, in this chapter we introduce a novel memristive CNN, named MCNN, and we show that it can exhibit autowaves. This MCNN represents an universal paradigm following a process of generalization schematically shown in Fig. 4.1. In fact, the invention of the memristor leads to a new paradigm in which four components are at the basis of any electrical circuit, as shown in Fig. 4.1. Based on this new component, novel dynamic, eventually chaotic, circuits can be designed. In turn these can be used as basic cells to define novel CNN architectures with general features. In the following we describe one of such architectures and show that it can generate autowaves.

4.1.1 Model of the memristive CNN

CNNs are usually based on first–order cells [31]. Cells are directly connected with their neighbours, but the global interaction is guaranteed because of indirect effects. CNN, in fact, includes both advanced features of neural network and cellular automata such as asynchronous

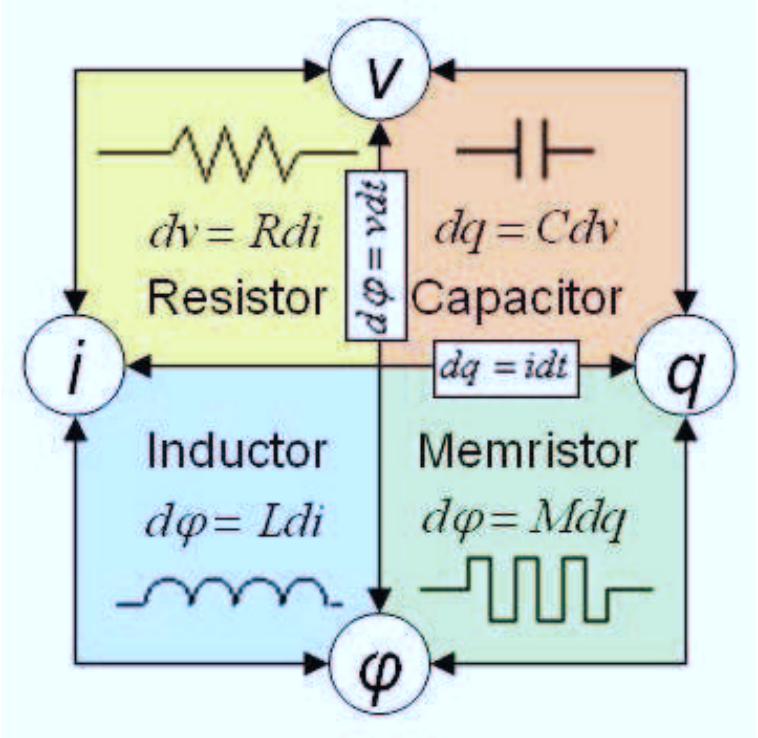


Fig. 4.1: Schematic representation of the four basic components of electrical circuits.

parallel processing, continuous–time dynamics and practical implementation. CNNs have applied in various areas: image signal processing, pattern recognition [32], bio–inspired robotic visions and biological functions [41].

When CNNs are made up of more complex cells (the cell itself can be a complex circuit, e.g., a Chua's circuit), difficult computational problems can be reformulated naturally by CNN. These CNN arrays have been examined for the generation of Turing patterns and various autowaves [29], where CNNs were used to approximate partial differ-

ential equations, especially reaction–diffusion equations. In our case, we adopt this second approach to explore complexity through memristive CNN. In this Section we represent the details of the fundamental memristive cell introduced as well as the overall configuration of our MCNN.

The memristive cell

In their work, Itoh and Chua introduced a gallery of different memristive oscillators [53]. Some of these oscillators are also able to display chaotic dynamics. Among the oscillators introduced in [53], to construct the basic cell used in our investigation, we choose a simple one, the so–called memristor–based Chua oscillator with a flux controlled memristor. The circuit of the cell is illustrated in Fig. 4.2, where it can be observed that the cell includes three elements: an inductor, a capacitor, and an active memristor. It is worth to notice that the active memristor consists of a negative conductance $(-G)$ and a passive memristor in parallel.

The dynamic equations of the cell are derived by applying Kirchhoff's circuit laws as follows

$$\begin{cases} C\dot{v} = -i - W(\varphi)\, v + Gv, \\ L\dot{i} = v, \\ \dot{\varphi} = v, \end{cases} \tag{4.1}$$

where v, i, and φ are voltage of the capacitor, current over inductor, and flux, respectively. The memristor is characterized by the memductance

$$W(\varphi) = \frac{dq(\varphi)}{d\varphi} = \begin{cases} a & |\varphi| < 1, \\ b & |\varphi| > 1, \end{cases} \tag{4.2}$$

where $q(\varphi)$ is a piecewise–linear function described by

$$q(\varphi) = b\varphi + 0.5(a-b)(|\varphi+1| - |\varphi-1|). \tag{4.3}$$

By replacing $x = v$, $y = i$, $z = \varphi$, $\alpha = 1/C$, $\beta = 1/L$, and $\gamma = G$, Eqs. (4.1) can be transformed into the following dimensionless equations:

$$\begin{cases} \dot{x} = \alpha(-y - W(z)x + \gamma x), \\ \dot{y} = \beta x, \\ \dot{z} = x, \end{cases} \tag{4.4}$$

From Eqs. (4.4) we can obtain

$$\dot{y} - \beta\dot{z} = 0. \tag{4.5}$$

Computing z in Eq. (4.5) yields

$$z = \frac{y+c}{\beta}, \tag{4.6}$$

where c is constant. Substituting Eq. (4.6) into Eq. (4.4), the following equation describing the dynamics of the memristive cell is obtained:

$$\ddot{y} + \alpha\left(W\left(\frac{y+c}{\beta}\right) - \gamma\right)\dot{y} + \alpha\beta y = 0. \tag{4.7}$$

The memristive circuit can be thus considered as a periodic second–order oscillator. Eq. (4.7) is equivalent to Eqs. (4.4). In the following, our analysis is however referred to Eqs. (4.4).

When the parameters are chosen such as $\alpha = 2$, $\gamma = 0.3$, $\beta = 1$, $a = 0.1$, and $b = 0.5$, Eqs. (4.4) exhibit the periodical signal shown

in Fig. 4.3. However, in order to get autowaves, Eqs. (4.4) should be characterized by a slow–fast dynamics [77], [7]. In the slow regime, the state of the limit cycle remains at a constant value for a considerably long period of time τ_{st}. After this long period, the state returns rapidly in a significantly short period of time τ_{ex}, where $\tau_{ex} << \tau_{st}$. By choosing appropriate parameters e.g., $\alpha = 10$, $\gamma = 0.3$, $\beta = 0.01$, $a = 0.1$, and $b = 0.5$, our memristive cell satisfies this requirement. The waveform of the signal (variable $x(t)$) is shown in Fig. 4.4, in which the slow–fast dynamics is clearly evident.

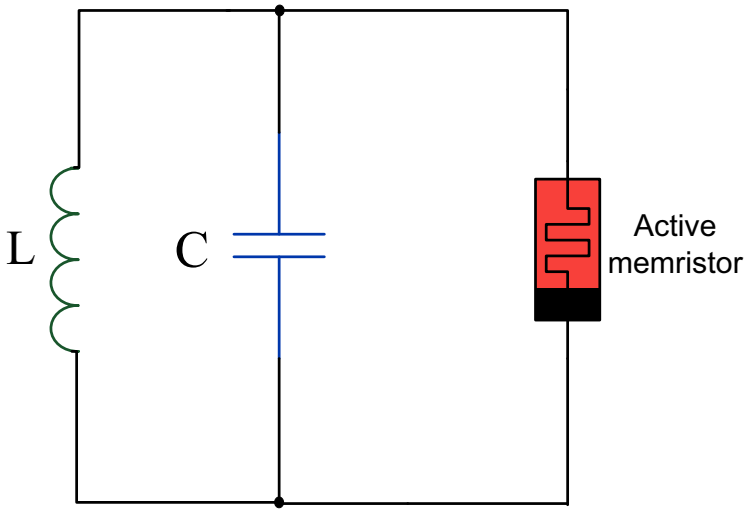


Fig. 4.2: The memristive cell.

Memristive CNN

In the literature, the autowaves were implemented by 1D CNNs, 2D CNNs or CNNs made of Chua's circuits. In [81] travelling waves, a

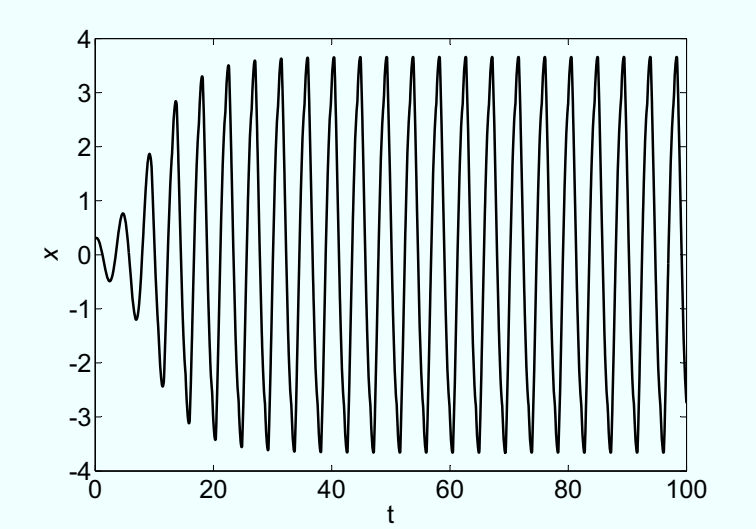


Fig. 4.3: Waveform of signal x when the parameters are chosen as $\alpha = 2$, $\gamma = 0.3$, $\beta = 1$, $a = 0.1$, and $b = 0.5$.

special case of autowaves, were studied in the 1D and 2D CNNs where the cell has two stable equilibrium points. The travelling waves only triggered from one stable equilibrium state to a second one and they remained from then on. Arena et al. [7] utilized the two–layer CNN model with second–order cells to obtain various complex phenomena such as autowaves, and Turing pattern formation, thus demonstrating that state–controlled CNNs give a physical realization of low–cost soft–computing devices.

We construct a MCNN as shown in Fig. 4.5. Each cell is connected with four neighbours by four linear resistors. As the result, a reaction–diffusion process is emulated.

From Fig. 4.5 the dynamic equations of the MCNN are derived

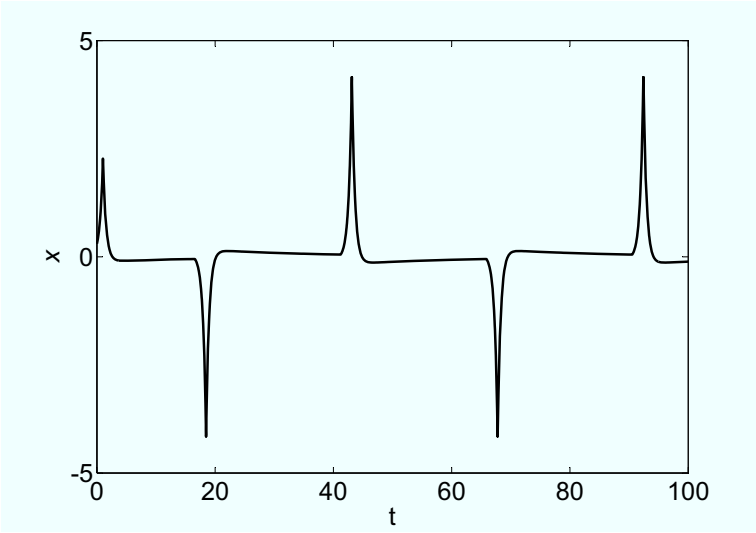


Fig. 4.4: Waveform of signal x when the parameters are chosen as $\alpha = 10$, $\gamma = 0.3$, $\beta = 0.01$, $a = 0.1$, and $b = 0.5$.

$$\begin{cases} \dot{x}_{i,j} = \alpha \left(-y_{i,j} - W\left(z_{i,j}\right) x_{i,j} + \gamma x_{i,j}\right) \\ \qquad\quad +D\left(x_{i-1,j} + x_{i+1,j} + x_{i,j-1} + x_{i,j+1} - 4x_{i,j}\right), \\ \dot{y}_{i,j} = \beta x_{i,j}, \\ \dot{z}_{i,j} = x_{i,j}, \end{cases} \tag{4.8}$$

where the diffusion coefficient D is constant.

4.1.2 Simulation results

The term *autowaves* or *autonomous waves* was introduced by R. V. Khorhlov [49]. Typical examples of autowaves include the waves of combustion, of phase transitions, concentration waves in chemical reactions, and also many biological autowave processes. According to [78], the autowaves have some noticeable properties as follows. The ampli-

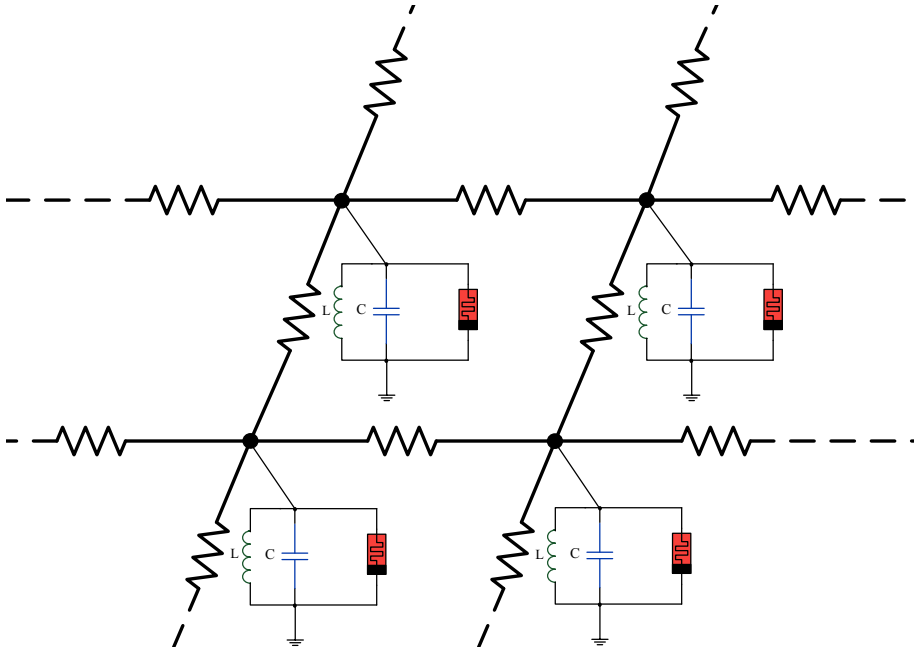


Fig. 4.5: Block diagram of the proposed memristive CNN.

tude and shape of the autowave do not change during propagation. Autowaves do not exhibit reflection or interference, but annihilation occurs when two waves collides. In this Section, the simulation results are reported to illustrate the principal features of autowaves in the MCNN.

A 50×50 MCNN with zero–flux boundary conditions has been simulated. Here, the diffusion coefficient was fixed to $D = 0.51$. We used the following initial conditions: $x_{i,j}(0) = y_{i,j}(0) = z_{i,j}(0) = 0$, where $1 \leq i, j \leq 50$, except $x_{i,2}(0) = 1.5$ in Fig. 4.6, $x_{25,25}(0) = 1.5$ in Fig. 4.7, $x_{i,2}(0) = x_{i,49}(0) = 1.5$ in Fig. 4.8 and $x_{25,2}(0) = x_{25,49}(0) = x_{26,2}(0) = x_{26,49}(0) = 1.5$ in Fig. 4.9.

An autowave has been observed moving with the same shape, from the left–hand side to the righ–hand side of the MCNN, as shown in Fig. 4.6. Unlike classical waves whose amplitude attenuates rapidly with the distance, the shape of autowave remains unchanged. When one point at the centre of the MCNN is active, an autowave propagates and a circle–shape appears as given in Fig. 4.7.

Fig. 4.8 represents two autowaves propagating from the left–hand side and the right–hand side as an effect of the initial conditions considered. The two waves move with the same velocity, finally annihilate each other when the two wavefronts collide with each other. Similarly, two colliding autowaves annihilate rather than penetrating one another, and, therefore, no interference takes place as shown in Fig. 4.9. The major characteristics of autowaves were therefore observed in the MCNN, thus showing the effectiveness of the introduced structure to generate autowaves. It is worth to notice that autowaves are generated with a very simple basic cell consisting of three components and, in view of the considerations in [72], in principle, even a simpler cell (consisting of either a capacitor or an inductor and an active memristor) can be used.

4.2 FPGA–based generation of autowaves in Memristive Cellular Neural Networks

Different hardware implementations of CNN systems emulating autowave propagation have been also proposed: for instance, Gomez-

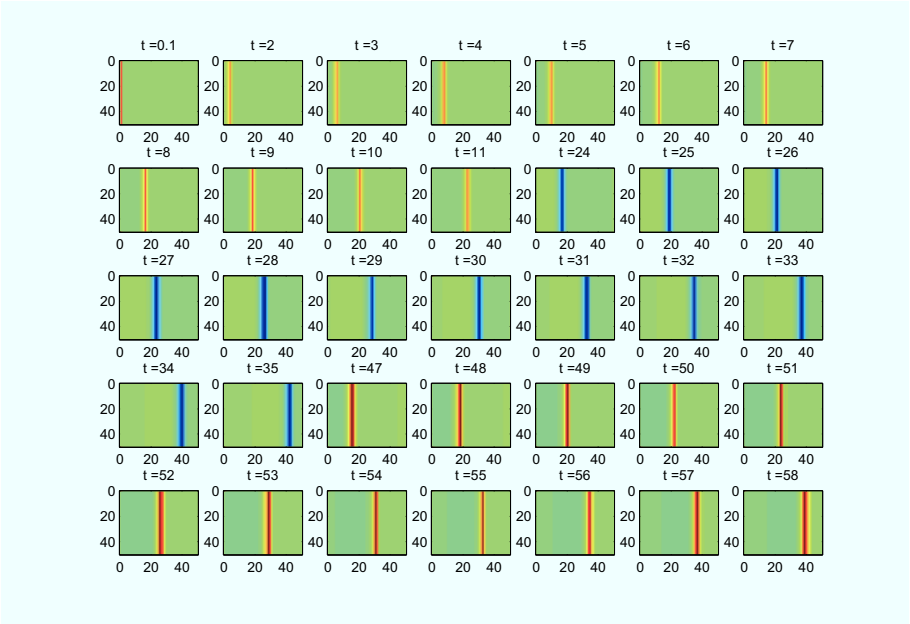


Fig. 4.6: 2D representation of an autowave propagating from the left–hand side to the right–hand side of a MCNN.

Gesteira et al. [47] designed a two–dimensional array (10×8 circuits) where every cell is a Chua's circuit; Yalcin and Suykens [120] demonstrated spatio–temporal pattern formation on the ACE16k CNN chip; and Yeniceri and Yalcin [123] proposed a real–time implementation of autowave generation on FPGA.

In this Section we consider CNNs made of memristor–based cells and implemented through FPGA and show that they can generate autowaves. While single memristor–based nonlinear circuits were focused in [53, 72, 71], as far as complex systems made of more interacting units are concerned we mention here a few examples. In [54], a cellular automaton and a discrete–time CNN (DTCNN) using nonlinear pas-

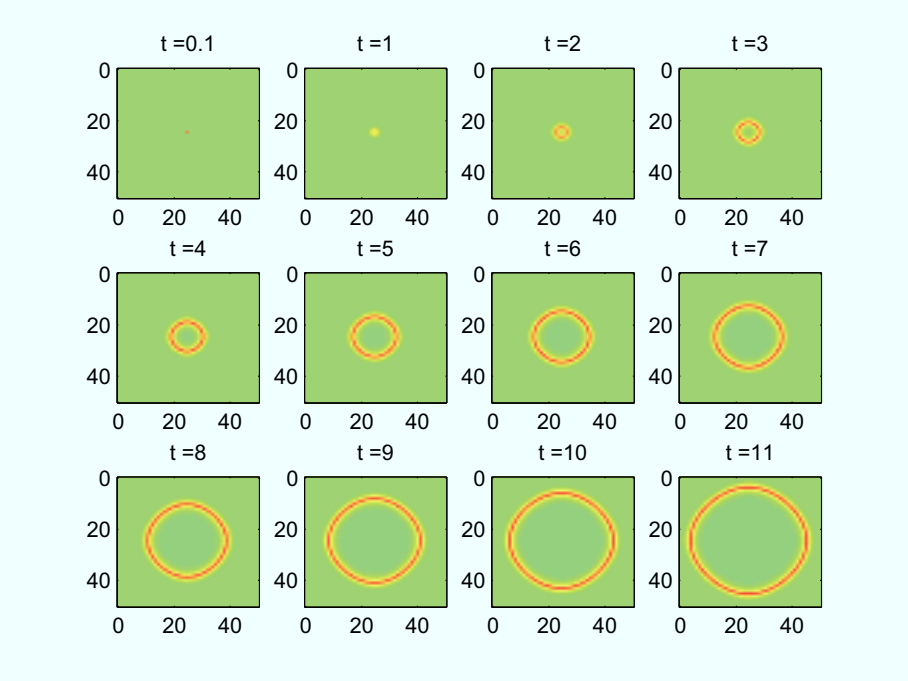


Fig. 4.7: Formation of an autowave from the centre of the MCNN.

sive memristors were designed. One noticeable feature of memristive DTCNN was the multitasking, since memristive DTCNN were shown to be able to perform more than one functions of the memristor cellular automaton at the same time. Another approach was the use of standard CNNs to implement memristive analogue circuits which then can be utilized as basic cells to realize chaotic circuits [20]. In another work [61], the role of memristors in implementing programmable connections of the cells was investigated. In Sect. 4.1 it has been shown that CNNs with memristive cells can generate autowaves. We investigate here an FPGA–based approach for the implementation of this model. The choice of the FPGA as a platform for the hardware imple-

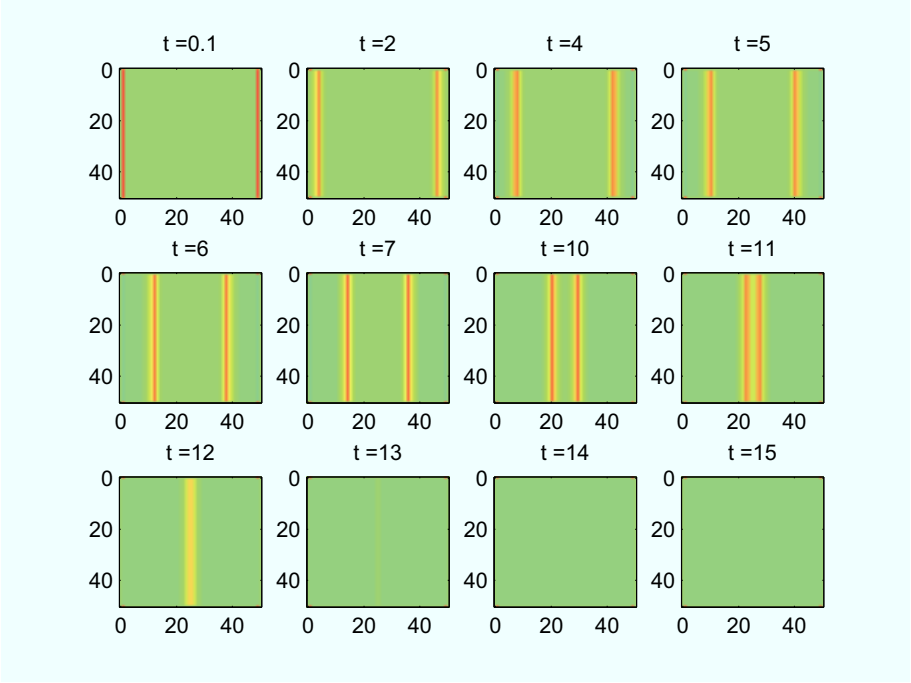


Fig. 4.8: Obliteration of two colliding autowaves, travelling from the left–hand side and the right–hand side.

mentation of the system is motivated by the fact that FPGA combines the advantages of both hardware and software systems by providing reprogrammable hardware.

4.2.1 FPGA–based implementation of Memristive Cellular Neural Networks

FPGA–based implementations concentrating on circuit architectures have been widely used for signal processing [86]. New algorithms can be easily tested with the FPGA development platform that reduces time consumption as well as the price of the process of design [57].

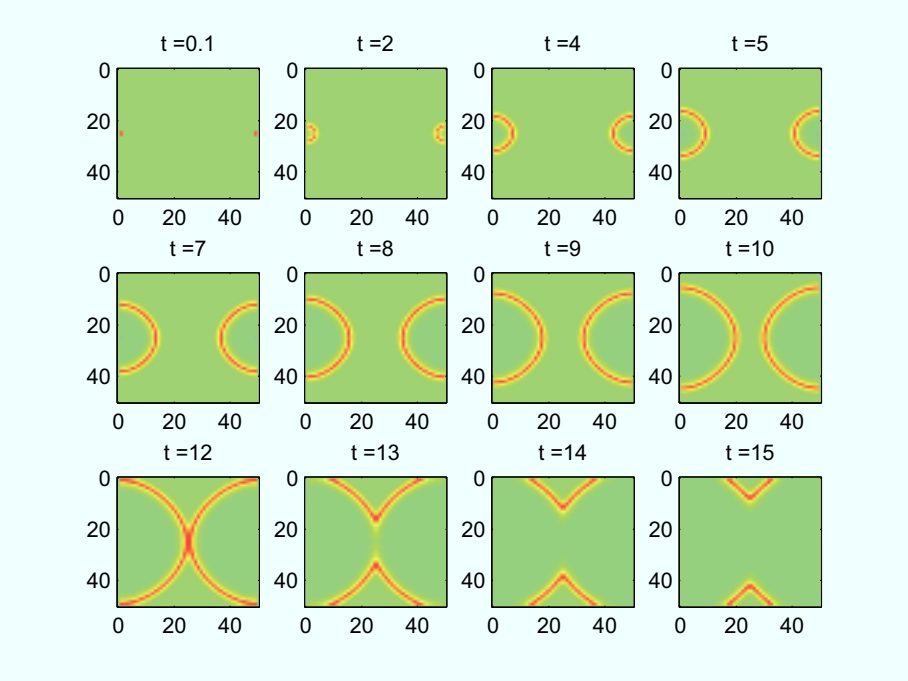


Fig. 4.9: Annihilation of two colliding autowaves.

In order to realize a system based on FPGA, its discrete model is required firstly. There, a discrete-time version of Eqs. (4.8) is obtained by applying the fourth–order Runge–Kutta integration method [82] as follows:

$$\begin{cases} x_{i,j}(k+1) = x_{i,j}(k) \\ \qquad\qquad +\frac{h}{6}\left(k_{1x;i,j} + 2k_{2x;i,j} + 2k_{3x;i,j} + k_{4x;i,j}\right), \\ \\ y_{i,j}(k+1) = y_{i,j}(k) \\ \qquad\qquad +\frac{h}{6}\left(k_{1y;i,j} + 2k_{2y;i,j} + 2k_{3y;i,j} + k_{4y;i,j}\right), \\ \\ z_{i,j}(k+1) = z_{i,j}(k) \\ \qquad\qquad +\frac{h}{6}\left(k_{1z;i,j} + 2k_{2z;i,j} + 2k_{3z;i,j} + k_{4z;i,j}\right), \end{cases} \tag{4.9}$$

where $k_{nx;i,j}$, $k_{ny;i,j}$ and $k_{nz;i,j}$ $(n = 1, 2, 3, 4)$ are evaluated as

$$\begin{cases} k_{1x;i,j} = \alpha\left(-y_{i,j}(k) + \gamma x_{i,j}(k) - W\left(z_{i,j}(k)\right) x_{i,j}(k)\right) \\ \qquad +D\left(x_{i-1,j}(k) + x_{i+1,j}(k) + x_{i,j-1}(k)\right. \\ \qquad \left.+x_{i,j+1}(k) - 4x_{i,j}(k)\right), \\ \\ k_{1y;i,j} = \beta x_{i,j}(k), \\ \\ k_{1z;i,j} = x_{i,j}(k), \end{cases}$$

$$
\begin{cases}
k_{2x;i,j} = \alpha \left[-y_{i,j}(k) - \frac{h}{2} k_{1y;i,j} + \gamma \left(x_{i,j}(k) + \frac{h}{2} k_{1x;i,j} \right) \right. \\
\qquad -W \left(z_{i,j}(k) + \frac{h}{2} k_{1z;i,j} \right) \left(x_{i,j}(k) + \frac{h}{2} k_{1x;i,j} \right) \Big] \\
\qquad +D \left[x_{i-1,j}(k) + \frac{h}{2} k_{1x;i-1,j} + x_{i+1,j}(k) \right. \\
\qquad +\frac{h}{2} k_{1x;i+1,j} + x_{i,j-1}(k) + \frac{h}{2} k_{1x;i,j-1} + x_{i,j+1}(k) \\
\qquad \left. +\frac{h}{2} k_{1x;i,j+1} - 4 \left(x_{i,j}(k) + \frac{h}{2} k_{1x;i,j} \right) \right], \\
\\
k_{2y;i,j} = \beta \left(x_{i,j}(k) + \frac{h}{2} k_{1x;i,j} \right), \\
\\
k_{2z;i,j} = x_{i,j}(k) + \frac{h}{2} k_{1x;i,j},
\end{cases}
$$

$$
\begin{cases}
k_{3x;i,j} = \alpha \left[-y_{i,j}(k) - \frac{h}{2} k_{2y;i,j} + \gamma \left(x_{i,j}(k) + \frac{h}{2} k_{2x;i,j} \right) \right. \\
\qquad -W \left(z_{i,j}(k) + \frac{h}{2} k_{2z;i,j} \right) \left(x_{i,j}(k) + \frac{h}{2} k_{2x;i,j} \right) \Big] \\
\qquad +D \left[x_{i-1,j}(k) + \frac{h}{2} k_{2x;i-1,j} + x_{i+1,j}(k) \right. \\
\qquad +\frac{h}{2} k_{2x;i+1,j} + x_{i,j-1}(k) + \frac{h}{2} k_{2x;i,j-1} + x_{i,j+1}(k) \\
\qquad \left. +\frac{h}{2} k_{2x;i,j+1} - 4 \left(x_{i,j}(k) + \frac{h}{2} k_{2x;i,j} \right) \right], \\
\\
k_{3y;i,j} = \beta \left(x_{i,j}(k) + \frac{h}{2} k_{2x;i,j} \right), \\
\\
k_{3z;i,j} = x_{i,j}(k) + \frac{h}{2} k_{2x;i,j},
\end{cases}
$$

$$\begin{cases} k_{4x;i,j} = \alpha\left[-y_{i,j}\left(k\right) - hk_{3y;i,j} + \gamma\left(x_{i,j}\left(k\right) + hk_{3x;i,j}\right)\right. \\ \qquad -W\left(z_{i,j}\left(k\right) + hk_{3z;i,j}\right)\left(x_{i,j}\left(k\right) + hk_{3x;i,j}\right)] \\ \qquad +D\left[x_{i-1,j}\left(k\right) + hk_{3x;i-1,j} + x_{i+1,j}\left(k\right)\right. \\ \qquad +hk_{3x;i+1,j} + x_{i,j-1}\left(k\right) + hk_{3x;i,j-1} + x_{i,j+1}\left(k\right) \\ \qquad \left.+hk_{3x;i,j+1} - 4\left(x_{i,j}\left(k\right) + hk_{3x;i,j}\right)\right], \\ \\ k_{4y;i,j} = \beta\left(x_{i,j}\left(k\right) + hk_{3x;i,j}\right), \\ \\ k_{4z;i,j} = x_{i,j}\left(k\right) + hk_{3x;i,j}, \end{cases}$$

and $h = 0.004$ is an interval.

Using the model (4.9), MCNNs can be implemented in an FPGA–based system with the block diagram shown in Fig. 4.10. The core of system is the "calculation and control block" which computes the integration of Eqs. (4.9), displays the results on a monitor and controls the interface with the RAM memory.

Here, RAM memory is an IS42S16400 high–speed synchronous dynamic RAM. It is the external memory provided on the development board and is used to store the state variables data of the memristive cells.

4.2.2 Experimental results

The whole system consists of an Altera DE2 development board and a monitor. DE2 board includes a Cyclone II EP2C35F672C6 FPGA chip in a 672–pin package and provides other hardware resources to connect to other external devices. The DE2 board is connected to the monitor

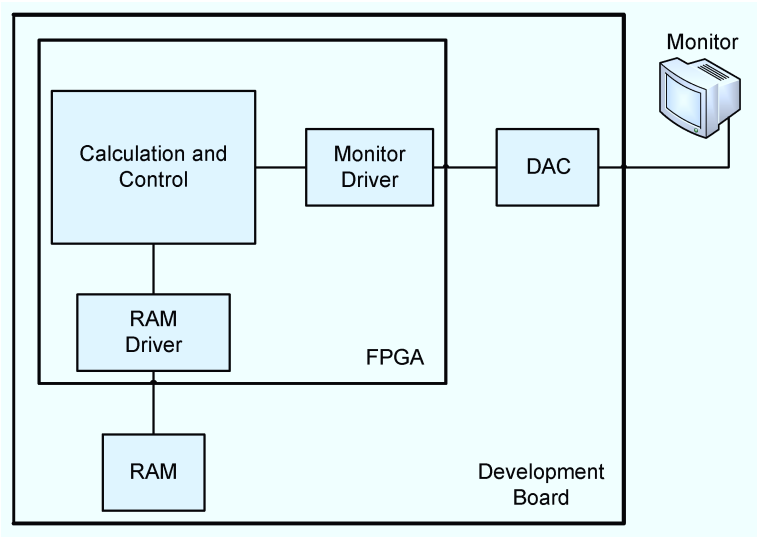
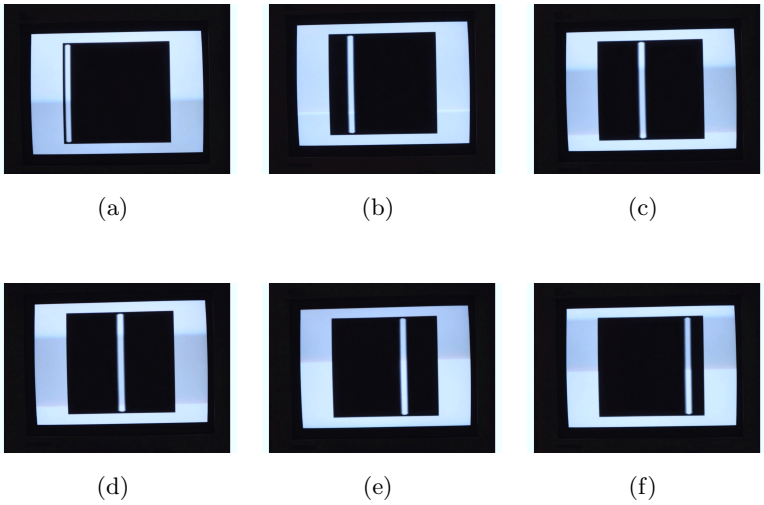


Fig. 4.10: Block diagram of the whole system based on an FPGA chip.

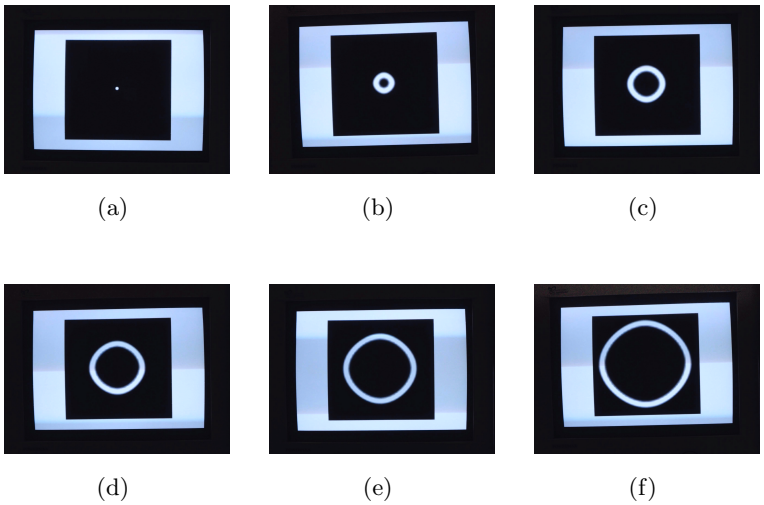
through a VGA DAC (10–bit high–speed triple DACs) with VGA–out connector. The calculation and control block, RAM driver block, and monitor driver block are programmed on FPGA chip with hardware description language VHDL. Firstly, the calculation and control block based on Altera's Nios II processor emulates MCNN by integrating Eqs. (4.9) for a time step. After that, the result is stored in the RAM and displayed on the monitor. The outputs of the FPGA system as shown in the monitor are then captured by a camera as reported in Figs. 4.11, 4.12, 4.13 and 4.14. These figures are intended to demonstrate that the generated waves have the main properties of the autowaves, namely an amplitude and shape of the wave that do not change during propagation, the absence of reflection or interference, and annihilation when two waves collides.



(a) (b) (c)

(d) (e) (f)

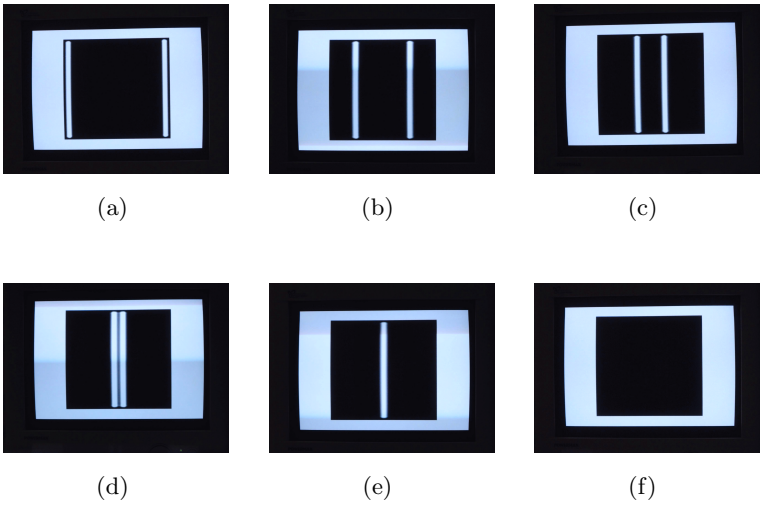
Fig. 4.11: Autowave propagation from the left–hand side to the right–hand side of the MCNN.



(a) (b) (c)

(d) (e) (f)

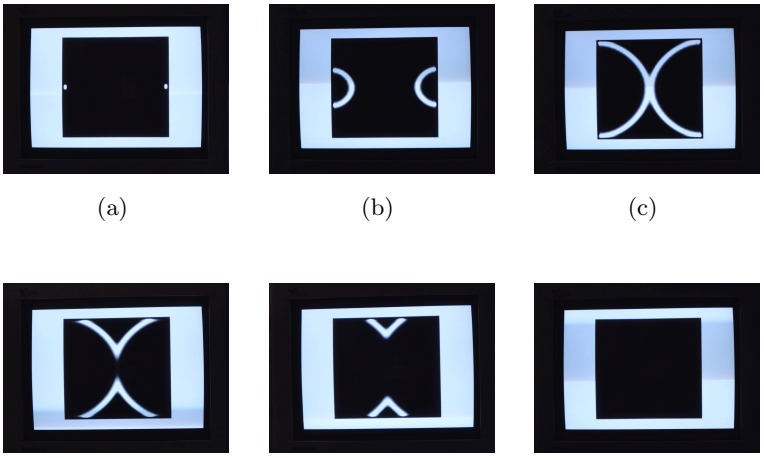
Fig. 4.12: Autowave formation starts from the centre of MCNN.



(a) (b) (c)

(d) (e) (f)

Fig. 4.13: Annihilation of two colliding autowaves.



(a) (b) (c)

(d) (e) (f)

Fig. 4.14: Annihilation of two colliding autowaves.

5

Simple memristive time–delay chaotic systems

Memristive systems have appeared in various application fields from non–volatile memory devices and biological structures to chaotic circuits. In this chapter, we propose nonlinear circuits based on memristive systems with the presence of delay, i.e., memristive systems in which the state of the memristor depends on the time–delay. These systems can exhibit chaotic behaviour and, notably, in the second model, only a capacitor and a memristor are required to obtain chaos.

Chaotic circuits have been designed to confirm theoretical models [43] as well as to be used in diverse applications such as secure chaotic communications [34], robotics [8] or random generator implementation [121]. Chaotic circuits can be either autonomous or non–autonomous, and an actual issue on the research on chaotic is the design of chaotic circuit with the minimum number of elements. On one hand, some simple non–autonomous chaotic circuits were proposed. Linsay built an anharmonic oscillator with a resistor, an inductor, and a varactor diode [63]. Dean [36] presented a circuit with a capacitor, a linear resistor, and

a resistor including ohmic losses in the inductor winding. Chaos could occur in a sinusoidally driven second–order circuit made of three linear elements and a Chua's diode [58]. A non–autonomous chaotic circuit based on one transistor, two capacitors, and two resistors was described by Lindberg [62, 42]. On the other hand, in the realm of autonomous circuits, Chua's circuit has received a significant amount of attention [43]. The four–element Chua's circuit introduced in [10] can be considered as the simplest circuit of this kind. In addition, by using a nonlinear active memristor, a three–elements autonomous circuit has been realized [72]. Piper [80] introduced some simple autonomous chaotic circuits using only op–amps and linear time–invariant passive components. More recently, the autonomous Hartley's oscillator based on a junction field effect transistor (JFET) and a tapped coil has been implemented [104]. Hence, the authors have named it the simplest chaotic two–component circuit. However, it is notable that, when the Tchitnga's circuit is analyzed in terms of the concept of mathematical simplicity given [80], it is not really simple because of its four state equations.

From another point of view, if the prospective study is to create a simple chaotic circuit described by few dynamical equations, it is clear that time–delayed systems are good candidates. Due to time–delay these systems are infinite–dimensional dynamical systems [118] and thus a system described by just one delay differential equation can be chaotic.

In this chapter, we investigate the possibility of designing memristive time–delay systems (MTDS) exhibiting chaotic behaviour. In par-

ticular, we introduce three different autonomous MTDS models and propose implementations of the second model which consists of just two components: a time–delay memristive element and a capacitor as well as the third model consisting of just one memristive element.

5.1 Models of memristive time–delay systems

This Section is devoted to the introduction of the mathematical models of two MTDS showing chaotic behavior. These models are built starting from circuit configurations who are good candidates for the generation of chaos. This is in view of the final aim of our research, which is the real implementation of the mathematical model introduced.

5.1.1 The 6–element memristive time–delay system

The first model introduced in this chapter is the MTDS shown in Fig. 5.1. The MTDS consists of an integrator, a nonlinear active memristor, and a single time–delay block. Analogously to the approach presented in [71], a nonlinear active memristive system is considered. In particular, it is governed by the following equations:

$$\begin{cases} \dot{y} = f\left(y, v_M, t\right) = l v_M + m y + n v_M y \\ i_M = G\left(y, v_M, t\right) v_M = \alpha v_M + \beta v_M y^2, \end{cases} \tag{5.1}$$

where v_M, i_M, y are the voltage across the terminals of the memristive system, the current through it and its state variable, respectively, and l, m, n, α, β are constants. By applying the Kirchhoff's circuit laws to the MTDS in Fig. 5.1, the following circuit equations are obtained:

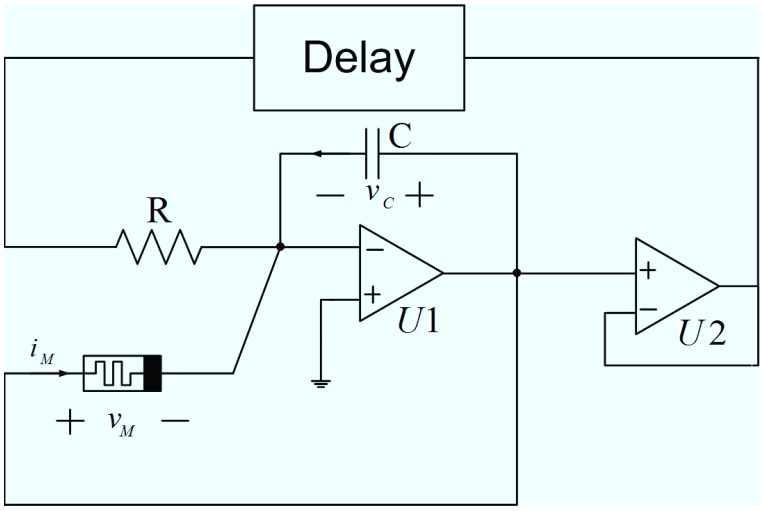


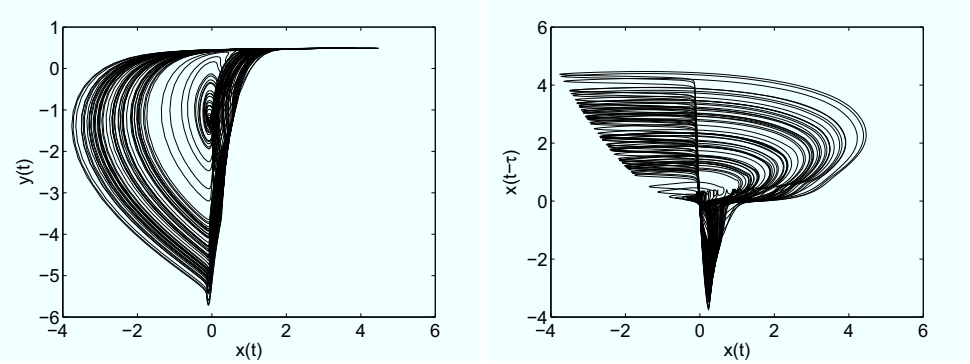
Fig. 5.1: Circuital model of the 6–element MTDS based on a nonlinear active memristor and a delay unit.

$$\begin{cases} \frac{dv_C(t)}{dt} = -\frac{\alpha}{C} v_C(t) - \frac{1}{RC} v_C(t-\tau) - \frac{\beta}{C} v_C(t)\, y^2(t) \\ \frac{dy(t)}{dt} = l v_C(t) + m y(t) + n v_C(t)\, y(t), \end{cases} \tag{5.2}$$

where τ is the time–delay. The dimensionless equations of the 6–element MTDS are derived as follows

$$\begin{cases} \dot{x} = ax + bx_\tau + cxy^2 \\ \dot{y} = lx + my + nxy, \end{cases} \tag{5.3}$$

where $x = v_C(t)$, $x_\tau = x(t-\tau)$, $a = -\frac{\alpha}{C}$, $b = \frac{-1}{RC}$, and $c = -\frac{\beta}{C}$. If we set $a = 1.5$, $b = -2$, $c = -2$, $l = 2.5$, $m = -0.5$, and $n = -5$, Eqs. (5.3) become



(a) (b)

Fig. 5.2: Projection of the chaotic attractors exhibited by the 6–element MTDS (5.4).

$$\begin{cases} \dot{x} = 1.5x - 2x_\tau - 2xy^2 \\ \dot{y} = 2.5x - 0.5y - 5xy. \end{cases} \tag{5.4}$$

Once fixed the values of the parameters, Eqs. (5.4) have been numerically integrated for different values of τ and chaos has been obtained through a sequence of period–doubling bifurcations induced by increasing values of this bifurcation parameter. The chaotic attractor obtained for $\tau = 1.3$ is shown in Fig. 5.2, while the bifurcation diagram of Eqs. (5.4) when τ is varied from 0.3 to 1.6 is illustrated in Fig. 5.3.

5.1.2 The 2–element memristive time–delay system

The most simple chaotic circuit based on memristor is the so–called 3–element circuit introduced in [72]: this consists of only three circuit elements (an inductor, a capacitor and a memristive system), since, according to the Poincarè-Bendixson theorem [13], three is the min-

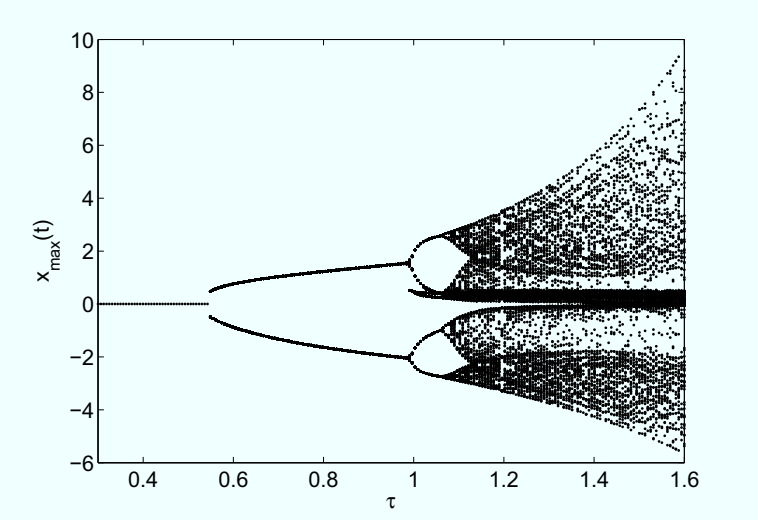


Fig. 5.3: Numerical bifurcation diagram for the 6–element MTDS, $\tau \in [0.3, 1.6]$.

imum number of state variables for an autonomous continuous–time system to be able to generate chaotic behavior. However, when time–delay systems are dealt with, since they can be considered as infinitive–dimensional dynamical systems [66], even one delay differential equation is enough to generate chaos [40, 52]. For this reason, we investigated maybe the simplest configuration with one time–delay memristive system: the parallel of a memristive system and a second circuit element. One can either consider the parallel with a resistor or with a memory element (inductor or capacitor). The first case was discarded, because the presence of a resistor in parallel with the meristive system has the only effect of redefining the $i - v$ characteristics of the memristive system. In the second case it is equivalent to consider an

inductor or a capacitor. We focused on the parallel of a capacitor with a nonlinear active memristive system. The circuit is shown in Fig. 5.4.

The memristive system in Fig. 5.4 is a voltage–controlled one–port system described by the following equations

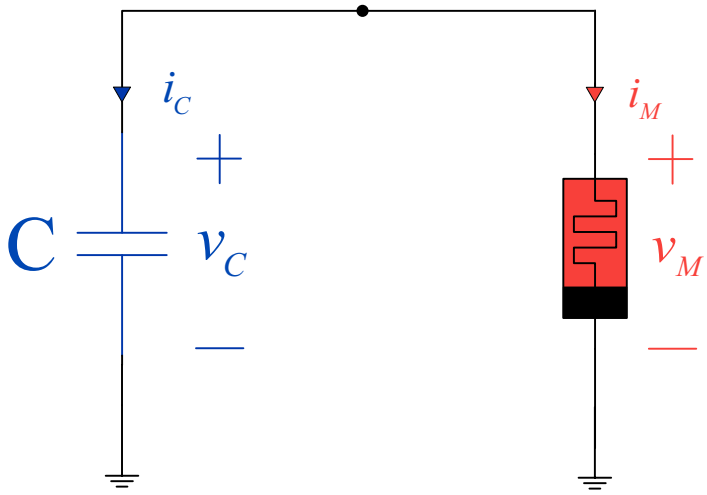


Fig. 5.4: The 2–element MTDS.

$$\begin{cases} \dot{x} = f\left(x_\tau, v_M, t\right) = a x_\tau + b\left|x_\tau\right| + c v_M \\ i_M = G\left(x, v_M, t\right) v_M = \left(\alpha + \beta x\right) v_M, \end{cases} \tag{5.5}$$

where x is the state variable of the memristor, τ is time–delay and a, b, c, α, β are constants. By applying the Kirchhoff's circuit laws and the constitutive relationship of the memristive system (5.5), the equations governing the circuit are obtained:

$$\begin{cases} \frac{dx(t)}{dt} = a x\left(t-\tau\right) + b\left|x\left(t-\tau\right)\right| + c v_C\left(t\right) \\ \frac{dv_C(t)}{dt} = -\frac{\alpha}{C} v_C\left(t\right) - \frac{\beta}{C} x\left(t\right) v_C\left(t\right). \end{cases} \tag{5.6}$$

By defining $y = v_C(t)$, $x_\tau = x(t-\tau)$, $m = -\frac{\alpha}{C}$, and $n = -\frac{\beta}{C}$, the following dimensionless equations are derived for the 2–element MTDS:

$$\begin{cases} \dot{x} = a x_\tau + b\,|x_\tau| + c y \\ \dot{y} = m y + n x y. \end{cases} \tag{5.7}$$

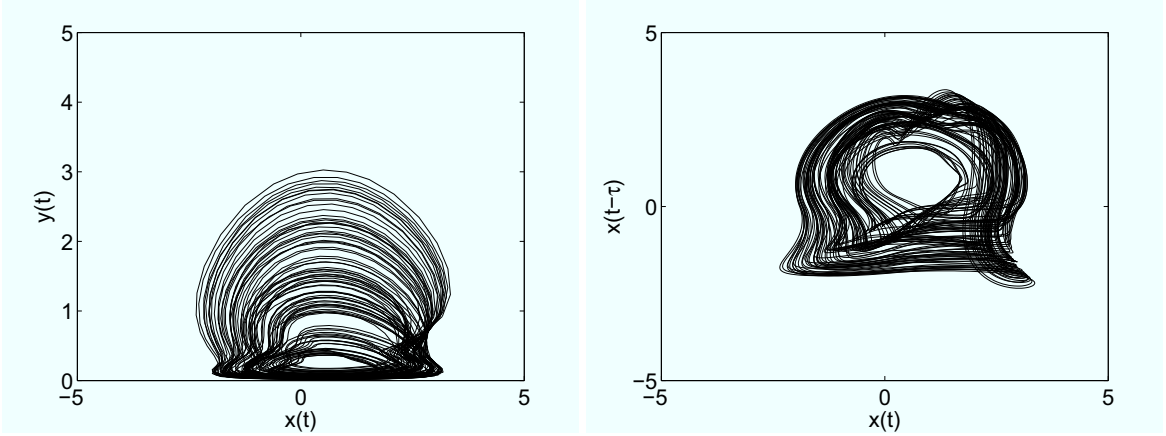
In the following we set $a = 1$, $b = -2$, $c = 5$, $m = 0.5$, and $n = -0.9$ and consider τ as a bifurcation parameter. Eqs. (5.7) become:

$$\begin{cases} \dot{x} = x_\tau - 2\,|x_\tau| + 5y \\ \dot{y} = 0.5y - 0.9xy. \end{cases} \tag{5.8}$$

Eqs. (5.8) have been numerically integrated and chaotic behavior has been obtained for $\tau \gtrsim 1.25$. An example of the chaotic behavior obtained with the 2–element MTDS is shown in Fig. (5.5), while the bifurcation diagram with respect to τ is shown in Fig. (5.6). Another interesting bifurcation parameter is b. When this parameter is varied, chaos is preserved for a quite large interval, beyond which periodic behavior or stable equilibrium point is obtained. The bifurcation diagram with respect to b is shown in Fig. (5.7).

5.1.3 One–element memristive time–delay system

In the previous Section a memristive time–delayed system with two elements (only a capacitor and a memristor) has been presented. This simple configuration can be an autonomous chaotic oscillator, however the second circuit element is a memory element. In order to construct a simpler configuration, we consider the novel memristive system which consists of only a memristor and a DC voltage source.



(a) (b)

Fig. 5.5: Chaotic attractor of two–element MTDS (5.8) obtained for $\tau = 1.3$.

For the sake of simplicity, we start from the circuit configuration shown in Fig. 5.8 where the memristive time–delay system (MTDS) includes a DC voltage source and a nonlinear active memristor.

The memristive system in Fig. 5.8 is a voltage–controlled one–port system described by the following equations

$$\begin{cases} \dot{x} = f\left(x_\tau, v_M, t\right) = a x_\tau + b\mathrm{sgn}\left(x_\tau\right) + c v_M \\ i_M = G\left(x, v_M, t\right) v_M = \left(\alpha + \beta x\right) v_M, \end{cases} \tag{5.9}$$

where x is the state of the memristor. There the signum function $\mathrm{sgn}(x)$ which can be implemented easily by an operational amplifier [119, 80, 96] is defined as

$$\mathrm{sgn}\left(x\right) = \begin{cases} -1, & \text{for } x < 0 \\ 0, & \text{for } x = 0 \\ 1, & \text{for } x > 0. \end{cases} \tag{5.10}$$

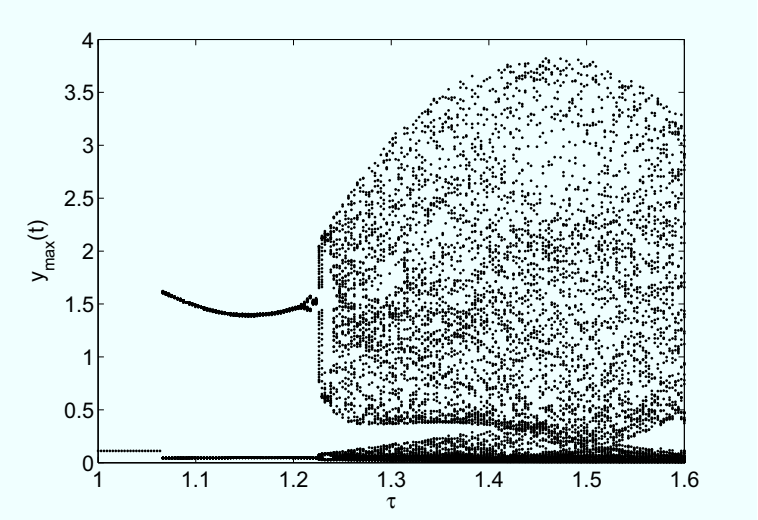


Fig. 5.6: Numerical bifurcation diagram for the 2–element MTDS (5.8), $\tau \in [1, 1.6]$.

By applying the Kirchhoff's circuit laws to the MTDS, the circuit equations are obtained as

$$\begin{cases} \frac{dx(t)}{dt} = ax\left(t-\tau\right) + b\mathrm{sgn}\left(x\left(t-\tau\right)\right) + cV_e \\ i_M\left(t\right) = \left(\alpha + \beta x\left(t\right)\right)V_e. \end{cases} \tag{5.11}$$

where τ is the delay. By defining $y = i_M(t)$, the dimensionless equation of MTDS is derived as

$$\begin{cases} \dot{x} = ax_\tau + b\mathrm{sgn}\left(x_\tau\right) + cV_e \\ y = \alpha V_e + \beta V_e x. \end{cases} \tag{5.12}$$

The parameters are chosen as $a = -1$, $b = 1$, $c = -0.5$, $V_E = 1$, $\alpha = -0.5 \times 10^{-3}$, and $\beta = 0.625 \times 10^{-3}$, hence Eqs. (5.12) become

$$\begin{cases} \dot{x} = -x_\tau + \mathrm{sgn}\left(x_\tau\right) - 0.5 \\ y = -0.5 \times 10^{-3} + 0.625 \times 10^{-3} x. \end{cases} \tag{5.13}$$

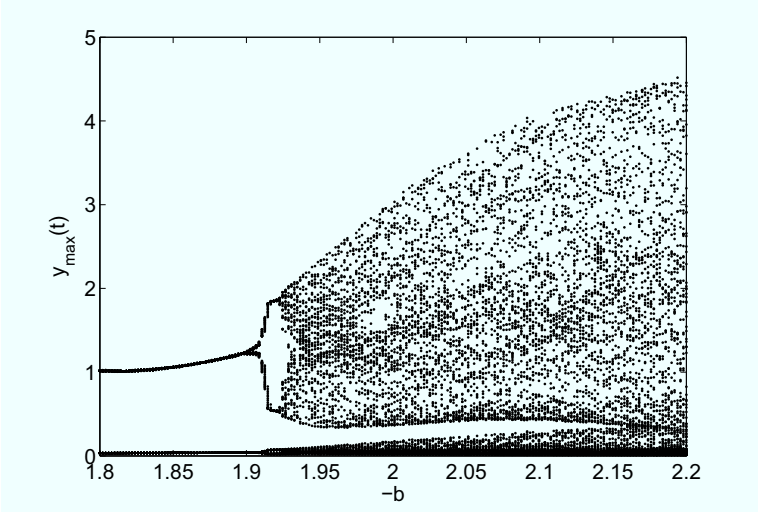


Fig. 5.7: Numerical bifurcation diagram for the 2–element MTDS (5.8), $b \in [-2.2, -1.8]$.

When τ equals 2, the maximum Lyapunov exponent of the MTDS (5.13) calculated by the discussed method in [40, 117, 95] is $\lambda_{max} \simeq 0.1633$ therefore the system (5.13) is a chaotic system. The chaotic attractor and the state variable y obtained for $\tau = 2$ are illustrated in Fig. 5.9 and Fig. 5.10, respectively.

5.2 Implementation of the 2–element memristive time–delay system

In this Section we discuss the implementation of the 2–element memristive time–delay circuit and related experimental results. The memristive system in Eqs. (5.5) is implemented through a multiplier and a series of operational–based blocks devoted to realize the different

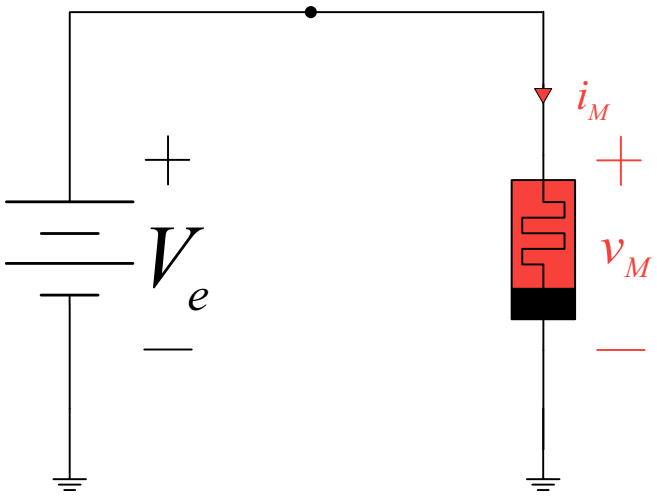


Fig. 5.8: Model of MTDS based on a single nonlinear active memristor.

terms appearing in Eqs. (5.5), so that the whole circuit is implemented as in Fig. 5.11. It consists of a capacitor C_1 in parallel with the circuitry needed to implement the memristive system. The state variables of the mathematical model x, y are implemented as voltages across the two capacitors C_2, C_1 respectively, following for the design of the operational–based blocks and for the choice of the parameters of the circuit components the design guidelines detailed in [4, 5, 43, 71, 72, 96]. The time–delay block has been also implemented with an operational–based approach. In particular, the approach discussed in [21] and based on a cascade of Bessels filters has been used. This approach is suitable to implement time–delays in the order of magnitude of milliseconds as required in our study.

The circuit equations have the following form

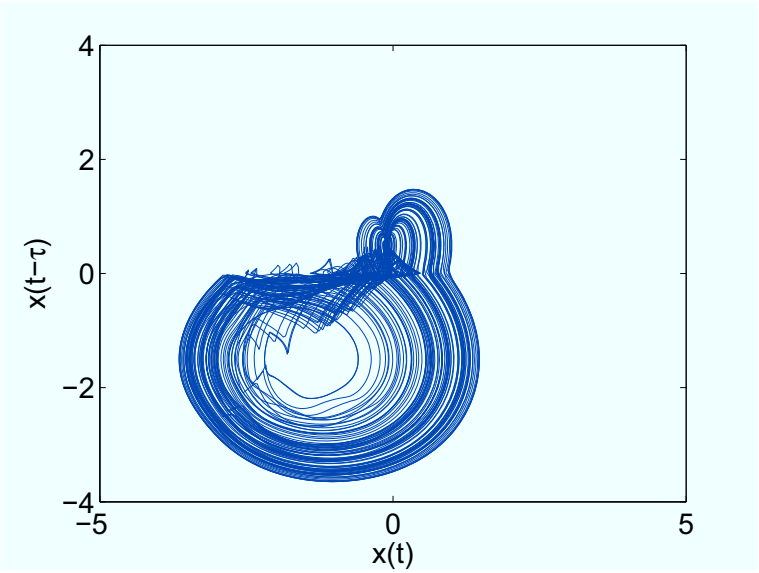


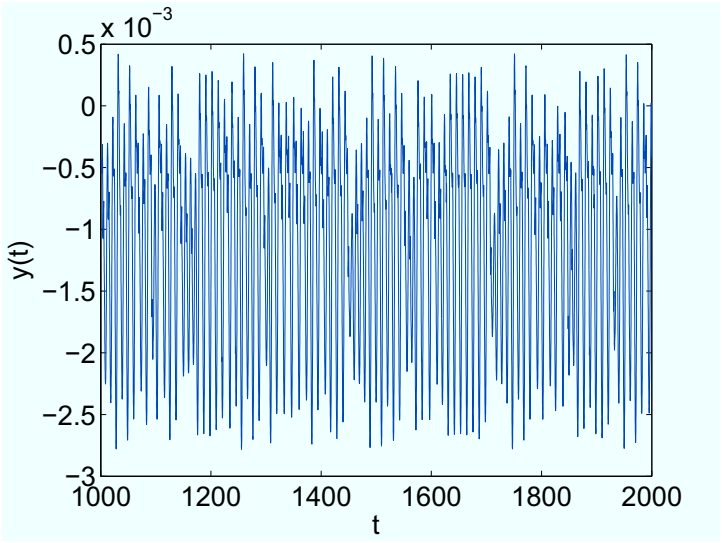
Fig. 5.9: Chaotic attractor exhibited by the MTDS (5.13) in the phase plane $x - x_\tau$.

$$\begin{cases} \frac{dx}{dt} = \frac{1}{R_{13}C_2}\left(-x + \frac{R_{11}}{R_8}x + \frac{R_{11}}{R_{10}}x_\tau - \frac{R_{17}}{R_{16}}\frac{R_{11}}{R_7}|x_\tau| + \frac{R_{11}}{R_9}y\right) \\ \frac{dy}{dt} = \frac{1}{R_3C_1}y - \frac{R_4+R_5}{10R_3C_1R_4}xy. \end{cases} \tag{5.14}$$

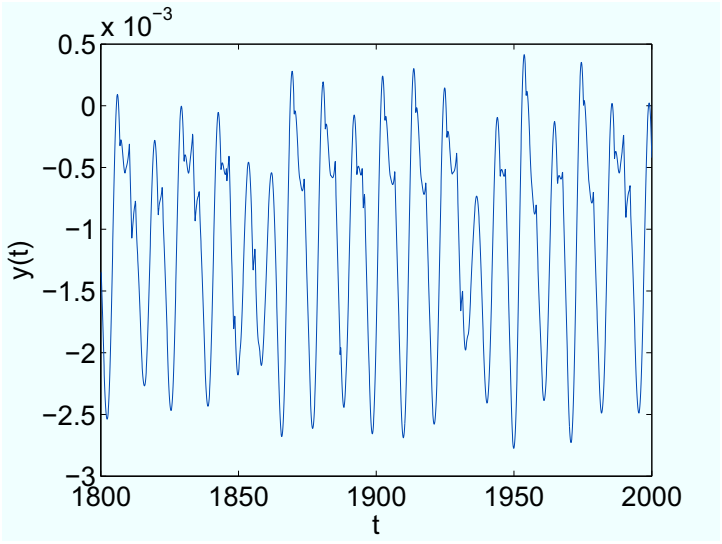
TL084 opamps and an Analog Devices AD633 multiplier have been used. Values of resistors and capacitors are reported in Fig. 5.11. The schematic of the time–delay block is shown in Fig. 5.12. The value of the time–delay implemented in this block is $T_{delay} = 1.3ms$, so that the dimensionless delay τ is:

$$\tau = \frac{T_{delay}}{R_{13}C_2} = 1.3. \tag{5.15}$$

The 2–element memristive time–delay circuit has been implemented on a breadboard with discrete off–the–shelf components. Signal wave-



(a)



(b)

Fig. 5.10: Waveform of the state variable y of the MTDS (5.13) for $\tau = 2$.

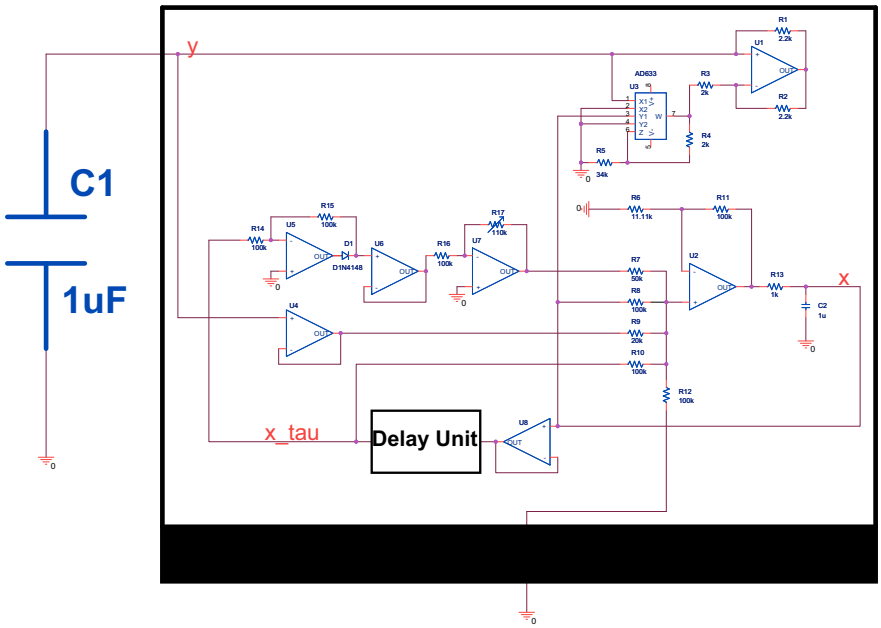


Fig. 5.11: Schematic of the 2–element memristive time–delay circuit. R_{17} is a variable resistor which implements the bifurcation parameter b. The values of components are as follows $R_1 = R_2 = 2.2$ kΩ, $R_3 = R_4 = 2$ kΩ, $R_5 = 34$ kΩ, $R_6 = 11.11$ kΩ, $R_7 = 50$ kΩ, $R_9 = 20$ kΩ, $R_{13} = 1$ kΩ, $R_{17} = 110$ kΩ, $R_8 = R_{10} = R_{11} = R_{12} = R_{14} = R_{15} = R_{16} = 100$ kΩ, and $C_1 = C_2 = 1\ \mu F$.

forms have been recorded by using a data acquisition board National Instruments SCB–68 with a sampling frequency of $f_s = 10kHz$ for $T = 5s$. The chaotic attractor obtained for $R_{17} = 100k\Omega$, corresponding to $b = -2$, is shown in Fig. 5.13. A good agreement between the theoretical and experimental attractor can be observed. We have then investigated the behaviour of the circuit with respect to b, by varying

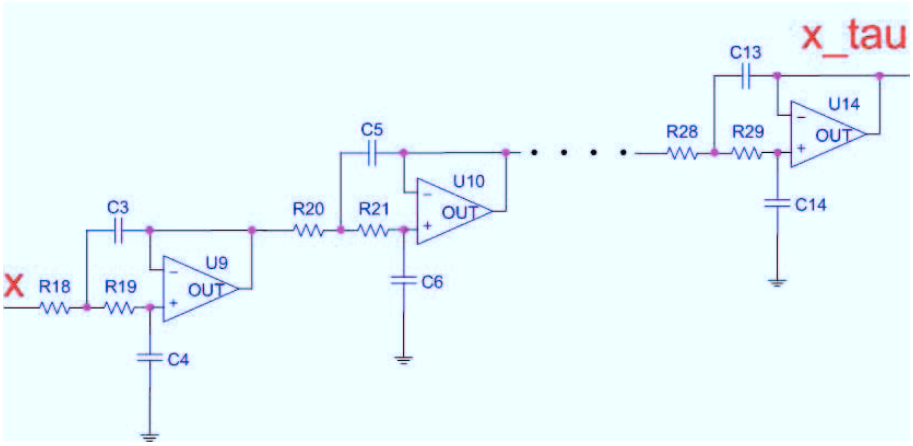
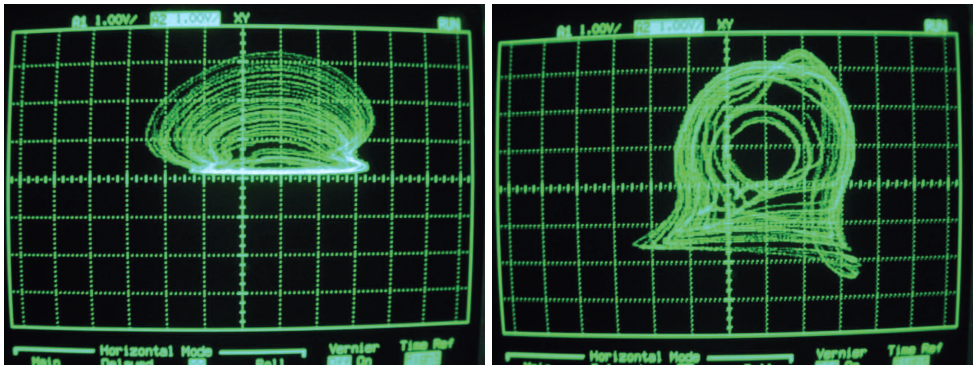


Fig. 5.12: Schematic of the delay unit including 6 Bessel filters in series where $R_{18} = R_{19} = R_{20} = R_{21} = \ldots = R_{28} = R_{29} = 10$ kΩ, $C_3 = C_5 = \ldots = C_{2i+1} = \ldots = C_{13} = 22\ nF$, and $C_4 = C_6 = \ldots = C_{2i} = \ldots = C_{14} = 10\ nF$.

the value of the variable resistor R_{17}. There R_{17} consists of a $90k\Omega$ resistor in series with a $20k\Omega$ trimmer.

The experimental bifurcation diagram showing the local maxima of the output signal y at different values of R_{17} is shown in Fig. 5.14. Chaotic regions of the system behaviour and windows of periodic behaviors are observed in the bifurcation diagram. The experimental bifurcation diagram confirms the numerical analysis carried out in previous Section.



(a) (b)

Fig. 5.13: Experimental chaotic attractor of the 2–element memristive time–delay circuit in (a) $x(t)$ – $y(t)$ plane, and (b) $x(t)$ – $x(t-\tau)$ plane when $R_{17} = 100$ kΩ, and X axis = Y axis = 1 V/div.

5.3 Implementation of the single–memristor–based chaotic circuit

In order to confirm the theoretical memristive system in Eqs. (5.13), the single–memristor–based chaotic circuit is implemented by using cost–effective electronic components. The whole circuit (see Fig. 5.15) consists of a DC voltage source (V_e) in parallel with a memristive system. It has been designed following an approach based on operational amplifiers [43, 71, 96]. The state variables of the mathematical model x, y are the voltage across the capacitor C_1 and the current through the memristive system respectively. The time–delay block has been also implemented with an operational–based approach [21] and its detailed schematic is illustrated in Fig. 5.16. The circuit equations are derived

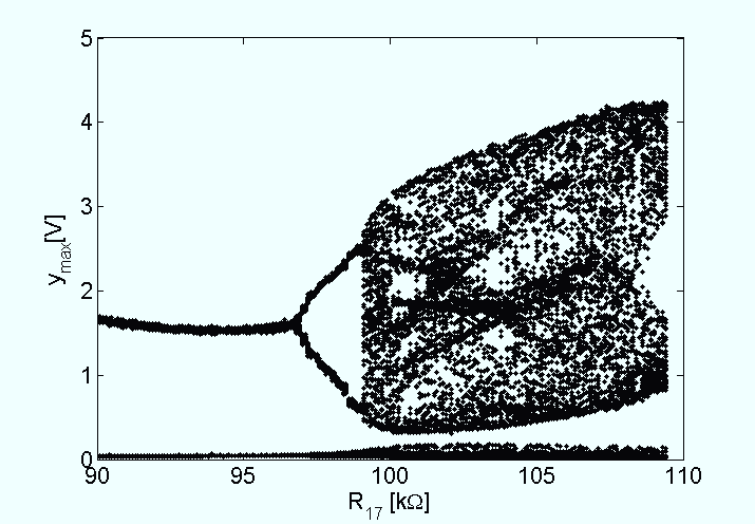


Fig. 5.14: Experimental bifurcation diagram for the 2–element memristive time–delay circuit with respect to the parameter b.

from Fig. 5.15 as follows

$$\begin{cases} \frac{dx}{dt} = \frac{1}{C_2}\left(-\frac{1}{R_6}x_\tau + \frac{V_{sat}}{R_7}\mathrm{sgn}\,(x_\tau) - \frac{1}{R_8}V_e\right) \\ y = -\frac{1}{R_3}V_e + \frac{R_4+R_5}{10R_3R_4}V_e x, \end{cases} \tag{5.16}$$

where V_{sat} is the saturation voltage of the operational amplifier $U5$. Waveforms of the chaotic attractor displayed on the oscilloscope have been captured as in Fig. 5.17. The good agreement between the theoretical and experimental attractor confirms that chaos can be obtained by our single–memristor–based chaotic circuit

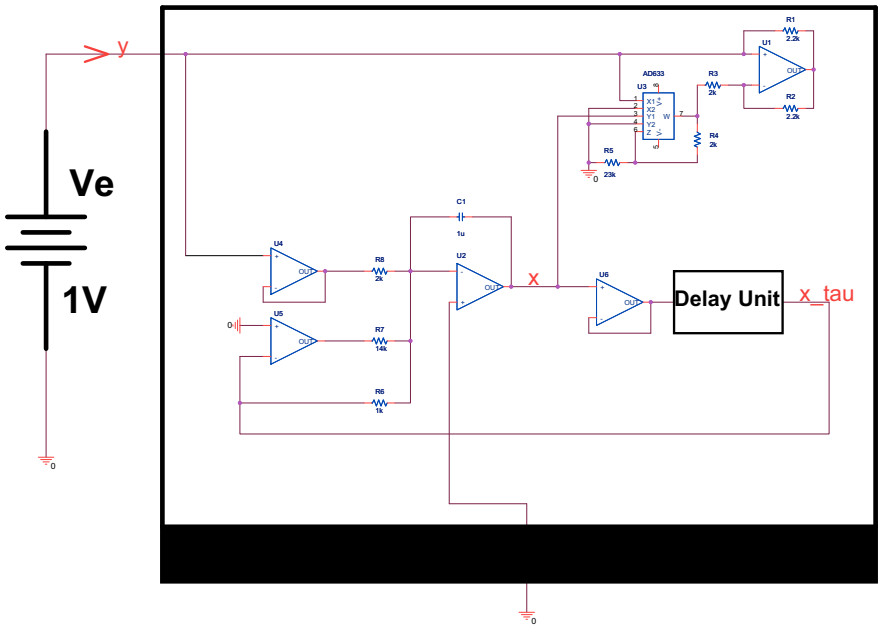


Fig. 5.15: Schematic of the single–memristor–based chaotic circuit in which the values of components are selected as follows: $R_1 = R_2 = 2.2$ kΩ, $R_3 = R_4 = 2$ kΩ, $R_5 = 23$ kΩ, $R_6 = 1$ kΩ, $R_7 = 14$ kΩ, $R_8 = 2$ kΩ, and $C_1 = 1$ μF.

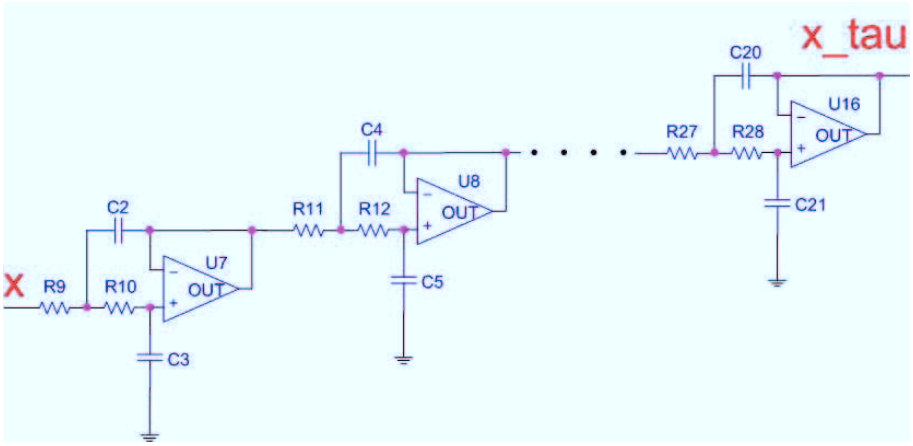


Fig. 5.16: Schematic of the delay unit based on the proposed approach in [21] where $R_9 = R_{10} = R_{11} = R_{12} = \ldots = R_{27} = R_{28} = 10$ kΩ, $C_3 = C_5 = \ldots = C_{2i+1} = \ldots = C_{21} = 10\ nF$, and $C_2 = C_4 = \ldots = C_{2i} = \ldots = C_{20} = 22\ nF$. The value of the time–delay of this unit is $T_{delay} = 2ms$ corresponding the dimensionless delay $\tau = 2$.

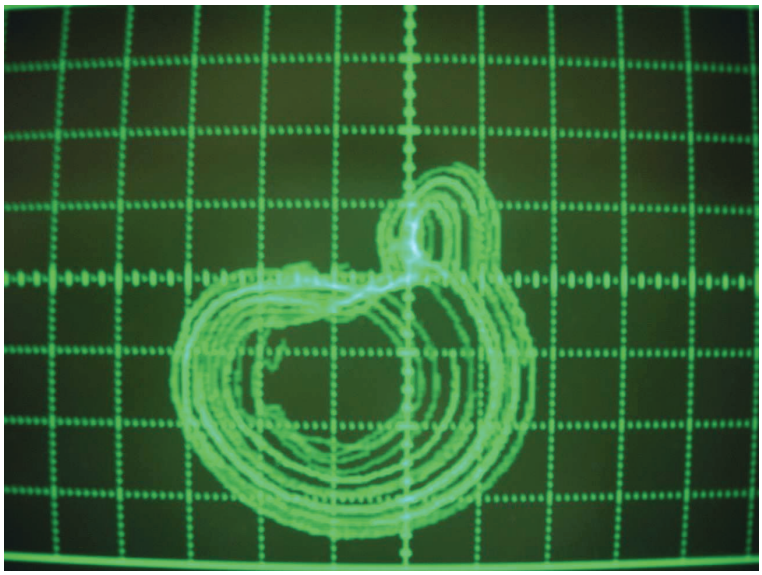


Fig. 5.17: Experimental chaotic attractors displayed in $x(t) - x(t-\tau)$ phase plane (horizontal axis: 1 V/div; vertical axis: 1 V/div).

6

Noise in chaotic circuits

The presence of noise is considered in two systems: a single chaotic system with only one stable equilibrium and a network of Chua's circuits. In the first system, we develop a novel control strategy for it both in the ideal case of absence of noise and in the presence of noise based on the exploitation of its peculiarities. While in the second one, we experimentally investigate the robustness to noise of synchronization in all the four–nodes network motifs.

6.1 Chaos control in a system with only one stable equilibrium and in the presence of noise

Three-dimensional (3-D) autonomous system has received an enormous amount of attention because it can be considered as the typical model of chaotic systems. Although numerous 3-D autonomous chaotic systems were found [96], there has been an ongoing discovery. It is worth noting that the novel chaotic systems with only stable equilibria have

reported recently. Chen and Wang proposed chaotic systems with only one stable equilibrium [112, 113] which cannot be proven by applying Silnikov criterion. The 3-D chaotic system with two stable node-foci including only two quadratic terms in a form very similar to the Lorenz system was introduced by Yang [122]. By generalizing a Sprott C system, Wei [116] investigated a new 3-D chaotic system with only two stable equilibria. Another 3-D autonomous chaotic system, which is topologically non-equivalent to the original Lorenz and all Lorenz-like systems, was studied by theoretical analysis in [115]. Moreover the approach constructing a chaotic system with any number of equilibria was presented in [114]. These discoveries indicate a relationship between the local stability of an equilibrium and the global complex dynamical behaviors of a chaotic system.

Controlling chaos has significant applications in various areas including physics, biology, communications or engineering [15, 87]. Starting from the first works of Ott, Grebogi, and York [75], many control techniques have been reported such as time-continuous delayed feedback control [88], notch filter feedback control [23, 1] or linear feedback control [26]. Depending on the characteristics of systems, appropriate techniques were chosen to control chaos in these ones. The discoveries of chaotic systems with the presence of stable equilibria motivate us to develop an effective control method for such systems.

6.1.1 Chaotic system with only one stable equilibrium

By adding a tiny perturbation to the Sprott E system [94], Wang and Chen [113] obtained a new system

$$\begin{cases} \dot{x} = yz + a \\ \dot{y} = x^2 - y \\ \dot{z} = 1 - 4x, \end{cases} \tag{6.1}$$

where a is a constant. The only one equilibrium of the system (6.1) is

$$P\left(x_E, y_E, z_E\right) = \left(\frac{1}{4}, \frac{1}{16}, -16a\right) \tag{6.2}$$

Because the tiny perturbation could change the stability of equilibrium, system (6.1) with only one stable equilibrium could exhibit chaotic behaviors, i.e., when $a = 0.006$ (see Fig. 6.1).

In order to observe the presence of multistability regions of system (6.1), forward and backward continuation for the bifurcation parameter a in the range of $a \in [-0.01, 0.02]$ have calculated and presented in Fig. 6.2. It is clear that chaotic attractor and stable equilibrium coexist, demonstrating the relation between the local stability of equilibria and the global complex dynamic of a system [112, 113], which cannot be proven by applying Silnikov criterion [91]. On one hand due to the existence of the stable equilibrium, chaos can be suppress by driving the system trajectory close to the equilibrium. But on another hand, the addition of an external unexpected force, especially noise may lead the system from the equilibrium to chaotic attractor.

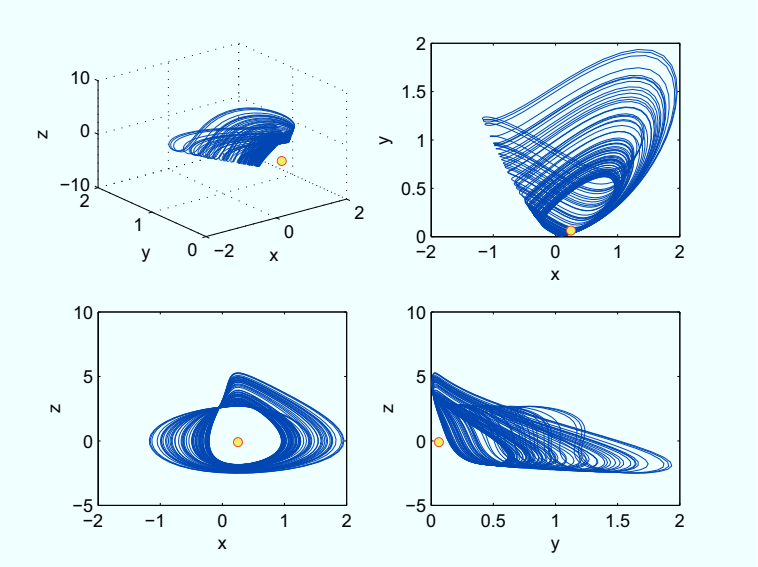


Fig. 6.1: The chaotic attractor of system (6.1) with only one stable equilibrium (yellow point), when $a = 0.006$.

6.1.2 Feedback control

Because the influence of noise on nonlinear dynamical systems has recorded in literature such as noise–induced synchronization [102, 101], noise-induced transitions [59, 12] or noise-induced chaos [44, 39], it is interesting to consider the effect of noise on system (6.1). In fact, Chen's system with the presence of noise is given as

$$\begin{cases} \dot{x} = yz + a \\ \dot{y} = x^2 - y + \varepsilon \dot{w} \\ \dot{z} = 1 - 4x, \end{cases} \tag{6.3}$$

where $\dot{w}$ is independent standard Wiener process with Gaussian increment, and the parameter ε is the noise intensity.

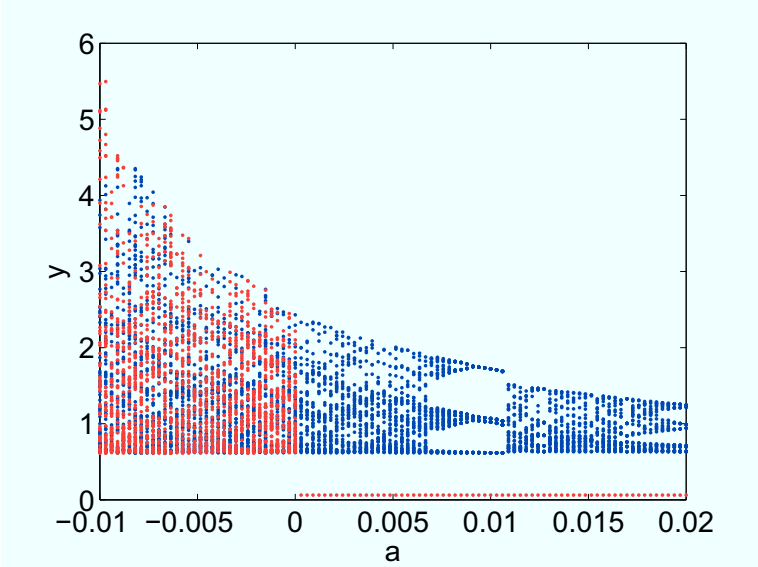


Fig. 6.2: Forward (blue line) and backward (red line) continuation of system (6.1) for a.

Linear feedback is one of the most effective feedback methods for chaos control due to the fact that this technique requires a simple structure controller and does not need the access to the system parameters. Linear feedback control applied successfully to control chaos in literature [25, 26]. Here, a new feedback control law will be introduced to suppress chaos and lead the system to $P\left(x_E, y_E, z_E\right)$. The controlled system is defined as

$$\begin{cases} \dot{x} = yz + a \\ \dot{y} = x^2 - y + \varepsilon \dot{w} \\ \dot{z} = 1 - 4x + u_z, \end{cases} \tag{6.4}$$

where the controller u_z is

$$u_z = \begin{cases} 0 & \text{if } |z - z_E| \leq c \\ k\,(z - z_E) & \text{if } |z - z_E| > c, \end{cases} \tag{6.5}$$

and k, c are constants.

In the absence of noise, after driving the system into the basin of attraction of the equilibrium point, the control u_z can be switched off. In contrast, in the presence of noise we have thus to consider that the noise can push the trajectory away from the equilibrium point and thus the control has to be reactivated. In order to estimate the threshold c for which the control has to be activated, we defined as δ the value of the constant that has to be initially added to go from the equilibrium point to the chaotic attractor. This value is presented as a function of the parameter a in Fig. 6.3 and is used to define the threshold for the control $c = \delta$.

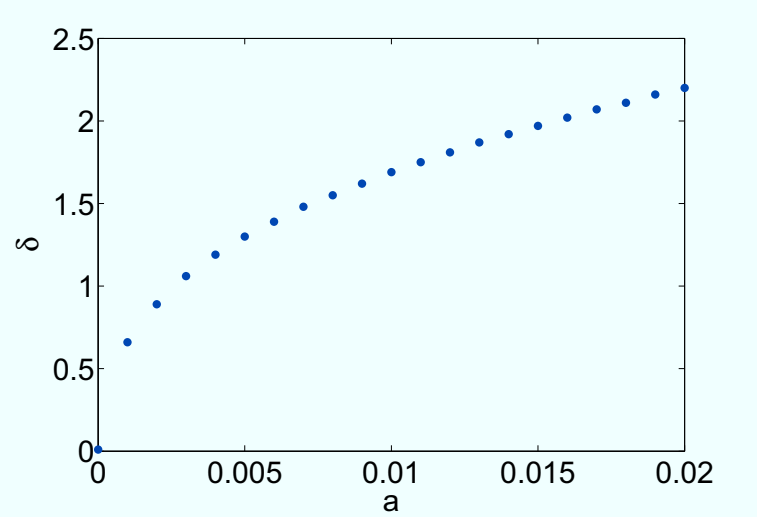


Fig. 6.3: The threshold δ as the function of the parameter a.

6.1.3 Experimental results

Experimental implementations have been carried out to confirm the proposed control strategy.

Emulating chaotic system with only one stable equilibrium by using CNN-based approach

Arena et al. introduced State Controlled-Cellular Neural Networks (SC-CNNs) for the analysis and design of complex system [4, 5]. Because of its simplicity such as the use of only common off-the-shelf electronic components, not requiring inductor, SC-CNN based method has been applied to implement many chaotic circuits [43, 50, 20]. A SC-CNN based circuit is realized to emulate Chen's system (6.1). The schematic of the circuit and values of components are shown in Fig. 6.4.

By applying the Kirchhoff's laws, the following equations are obtained for the Chen's system

$$\begin{cases} x = \frac{1}{R_7 C_1}\left(-x + \frac{R_1}{R_5}x + \frac{R_1}{R_4}yz + \frac{R_1}{R_3}a\right) \\ y = \frac{1}{R_{12} C_2}\left(-y + \frac{R_8}{R_{10}}x^2\right) \\ z = \frac{1}{R_{18} C_3}\left(-z + \frac{R_{13}}{R_{15}}z + \frac{R_{13}}{R_{16}}V_1 - \frac{R_{13}}{R_{17}}x\right). \end{cases} \tag{6.6}$$

The captured chaotic attractors (see Fig. 6.5) and experimental continuations (see Fig. 6.6) are in good agreement with simulations.

Experimental setup and measurements

The experimental setup for chaos control of the system in the presence of noise is constructed as shown in Fig. 6.7. It consists of Chen's circuit, controller, and noise generator.

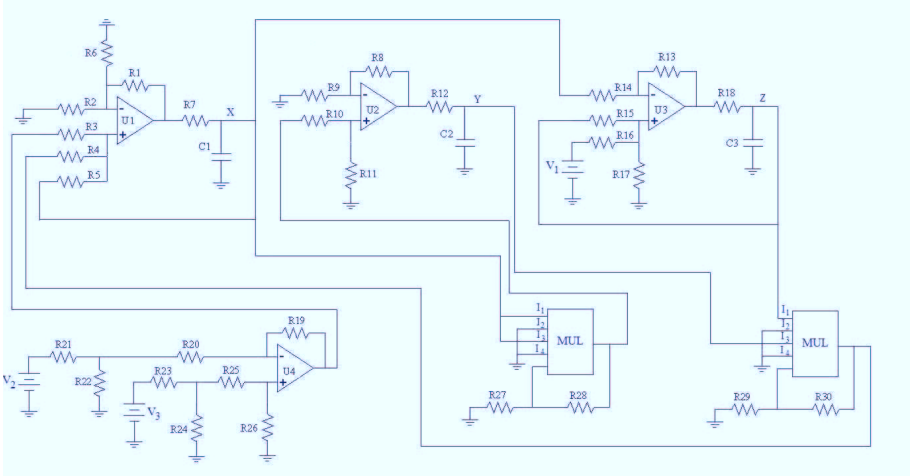
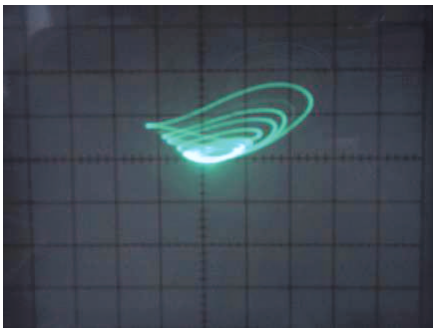
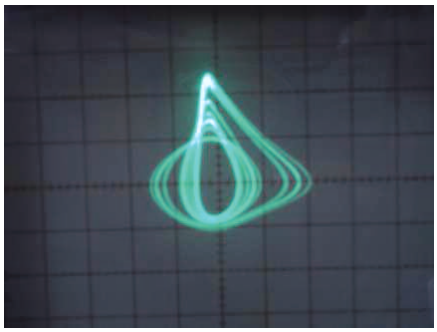


Fig. 6.4: Schematic of the circuit emulating Chen's system (6.1). Here, the values of the components are chosen as follows: $R_1 = R_2 = R_3 = R_4 = R_5 = R_6 = R_8 = R_9 = R_{10} = R_{11} = R_{13} = R_{15} = R_{16} = R_{20} = R_{25} = 100$ kΩ, $R_7 = R_{12} = R_{18} = 100\ \Omega$, $R_{14} = 25$ kΩ, $R_{17} = 33.33$ kΩ , $R_{21} = R_{23} = 10$ kΩ, $R_{22} = 690\ \Omega$, $R_{19} = R_{26} = R_{28} = R_{30} = 1$ kΩ, $R_{27} = R_{29} = 9$ kΩ, and R_{24} which implements the bifurcation parameter a is a variable resistor, $C_1 = C_2 = C_3 = 4.4\ \mu F$. The DC sources are $V1 = 1V$, $V2 = V3 = 15V$.

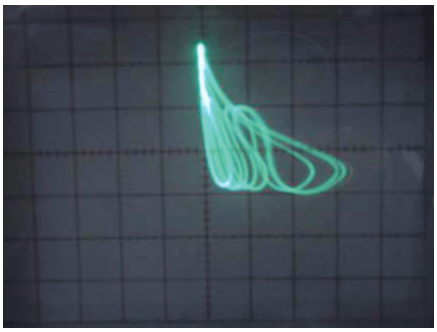
In our experiments, a Chua's circuit is employed to generate noise because of its simplicity and convenience [2, 3]. The frequency of this circuit is about 20 times greater than that of the Chen's circuit in order to let the Chua's circuit act as a noise generator. Experimental data have been acquired by using a National Instruments (NI–SB68) data acquisition board with a period of time $T = 10s$. Fig. 6.8 shows the trend of the state variable $z(t)$ and the control law $u_z(t)$ in the absence



(a) x-y phase plane



(b) x-z phase plane



(c) y-z phase plane

Fig. 6.5: Experimental chaotic attractors displayed on the oscilloscope when $a = 6mV$.

of noise. The system starts in the chaotic attractor. Then the control u_z is activated and it drives the system into the basin of attraction of the equilibrium point. After that although the control u_z is switched off, the system still stays in this state. In contrast, the same does not happen in the presence of noise as illustrated in Fig. 6.9. After driving the system to the equilibrium point, if the control u_z is turned off, the system goes to the chaotic attractor since the noise can lead the trajectory of the

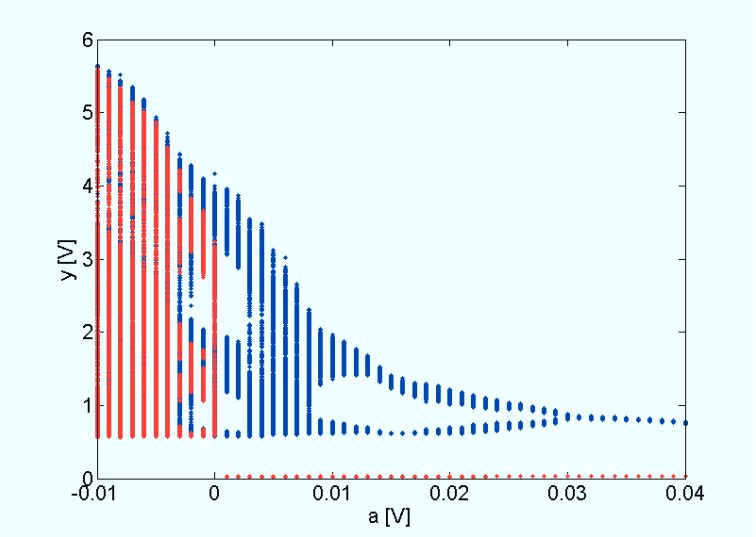


Fig. 6.6: Experimental continuations for the parameter a: forward continuation – blue line and backward continuation – red line.

system going far from the equilibrium point. Experimental results show the effectiveness and the precision of the designed controller.

6.2 Robustness to noise in synchronization of network motifs

Network motifs are defined as *recurring, significant patterns of interconnections* that can be found in complex networks [68]. Such structures of interconnections occur in a number, which is higher than in surrogate networks, obtained by randomly connecting the same number of nodes with the same number of links of the original networks. Therefore, they represent a clear topological feature distinguishing real

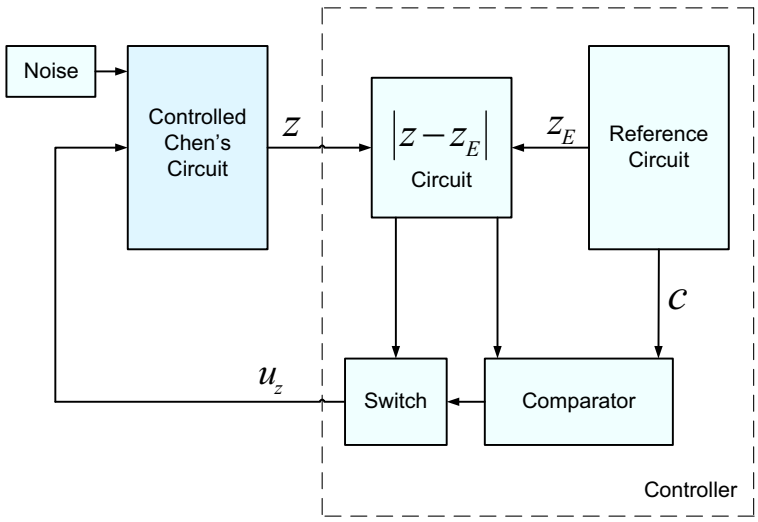


Fig. 6.7: Experimental setup including: Chen's circuit, noise, and controller.

networks from randomized ones. In particular, motifs are found in a variety of complex networks from different fields: ecology (for instance, food webs), biochemistry (genetic networks, for example), neurobiology (for instance, neural systems), and engineering (connectivity networks of logic electronic circuits) [68, 17]. Motifs abundant in networks from a specific field are distinct from those appearing in other types of networks, thus emphasizing that they have a significant role in the structure of that particular class of networks.

For this reason, beyond the occurrence of motifs in real networks, their relationship with the dynamical behavior exhibited by the network has been also studied. In particular, understanding how the topology of these motifs influence synchronization [16, 108] has been the sub-

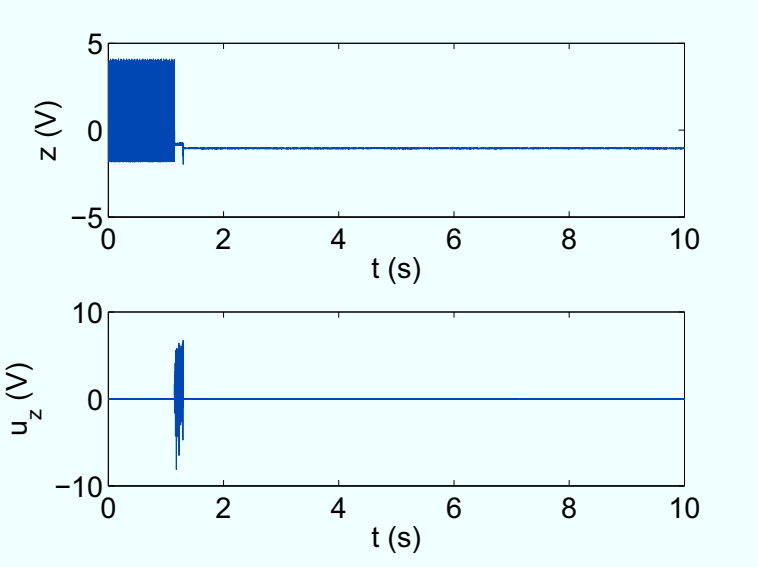


Fig. 6.8: Trend of the state variable $z(t)$ and the control law $u_z(t)$ in the absence of noise.

ject of several works [64, 109]. More specifically, in [109], the emergence of collective phase synchronization in triangular, square, and pentagonal network motifs, made of Kuramoto phase coupled oscillators, has been investigated and it has been found that lower synchronization thresholds occur in motifs with high interconnectedness, while in [64], a measure of the stability of the synchronous state in motifs has been introduced and found to be correlated (in motifs of undirected links) with their relative abundance in real networks.

Another aspect of interest in the synchronization of network motifs is the study of heterogeneous networks, i.e., networks of nodes with different dynamical behavior. In star–like motifs in which the central node (i.e., the hub) oscillates at different frequencies with respect to

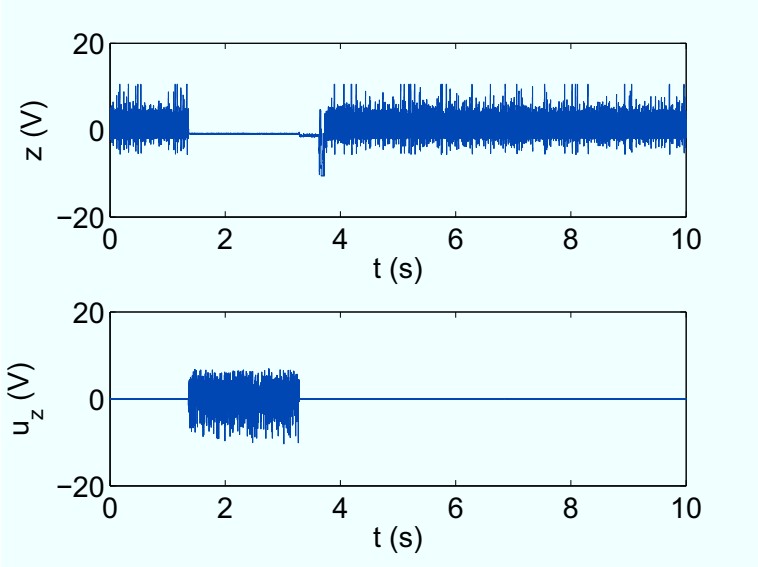


Fig. 6.9: Trend of the state variable $z(t)$ and the control law $u_z(t)$ in the presence of noise.

the peripheral ones (the leaves), phenomena such as remote synchronization [14] or explosive synchronization [46] may arise. These studies are in turn applied to understand what happens in other less simple topologies of complex networks of heterogeneous nodes.

In this Section, we experimentally investigate the role of noise in all the four–nodes network motifs, and in particular how it affects synchronization.

6.2.1 Experimental setup

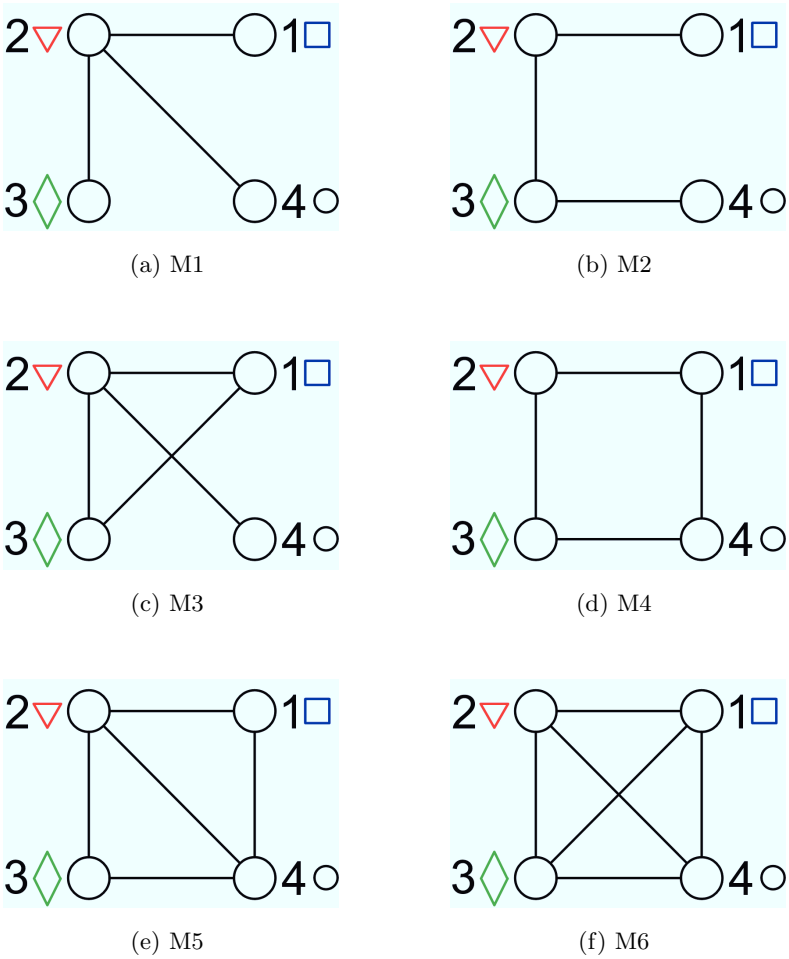
The experimental setup used in this work consists of a network of chaotic oscillators (Chua's circuits [67, 43]), a waveform generator Agilent 33220A used to generate a Gaussian white noise and an acquisition

board NI USB 6255. Synchronization in the presence of a noisy oscillator has been investigated, i.e., noise was applied only to a node of the network. In particular, four nodes have been taken into account and all the four–nodes possible undirected motifs have been investigated. There are six distinct four–nodes undirected network motifs as shown in Fig. 6.10. For each of these motifs, noise was applied to one node and a series of acquisitions of all the network signals with respect to increasing values of the noise level has been performed. The analysis has been then repeated for the other nodes of the network and then for the other motifs. So, for each motif four distinct series of acquisitions have been performed.

Fig. 6.11 shows the Chua's circuit used as a node of the network. The configuration used is the so–called state–controlled cellular nonlinear network one, discussed in detail in [43]. The parameters are chosen so that the circuit exhibits the double scroll chaotic attractor: $R_1 = 4$ kΩ, $R_2 = 13.3$ kΩ, $R_3 = 5.6$ kΩ, $R_4 = 20$ kΩ, $R_5 = 20$ kΩ, $R_6 = 380$ Ω (potentiometer), $R_7 = 112$ kΩ, $R_8 = 112$ kΩ, $R_9 = 1$ MΩ, $R_{10} = 1$ MΩ, $R_{11} = 8$ kΩ, $R_{12} = 1$ kΩ, $R_{13} = 51.1$ kΩ, $R_{14} = 100$ kΩ, $R_{15} = 100$ kΩ, $R_{16} = 100$ kΩ, $R_{17} = 100$ kΩ, $R_{18} = 1$ kΩ, $R_{19} = 8.2$ kΩ, $R_{20} = 100$ kΩ, $R_{21} = 100$ kΩ, $R_{22} = 7.8$ kΩ, $R_{23} = 1$ kΩ, $R_{24} = 2$ kΩ, $R_{25} = 2$ kΩ, $C_1 = 100$ nF, $C_2 = 100$ nF, and $C_3 = 100$ nF. The power supply of the circuit is $\pm 9V$.

Four of these circuits are used in the experimental setup, which is schematically illustrated in Fig. 6.12a. The circuits are coupled through resistors connected between the corresponding capacitors. In this way,

bidirectional, diffusive couplings are realized. The coupling network reflects the structure of the motif after substitution of each edge with a resistor of value R_c. An example of coupling network (the one corresponding to motif M2) is reported in Fig. 6.12b.



(a) M1 (b) M2

(c) M3 (d) M4

(e) M5 (f) M6

Fig. 6.10: Four–nodes undirected network motifs. Symbols are also associated to node labels to refer to the nodes in the plot of experimental results.

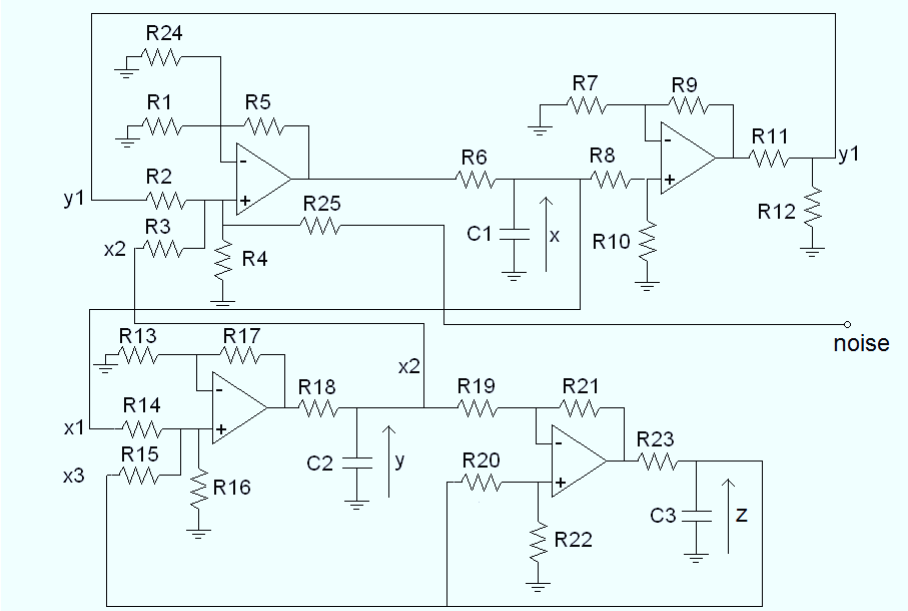


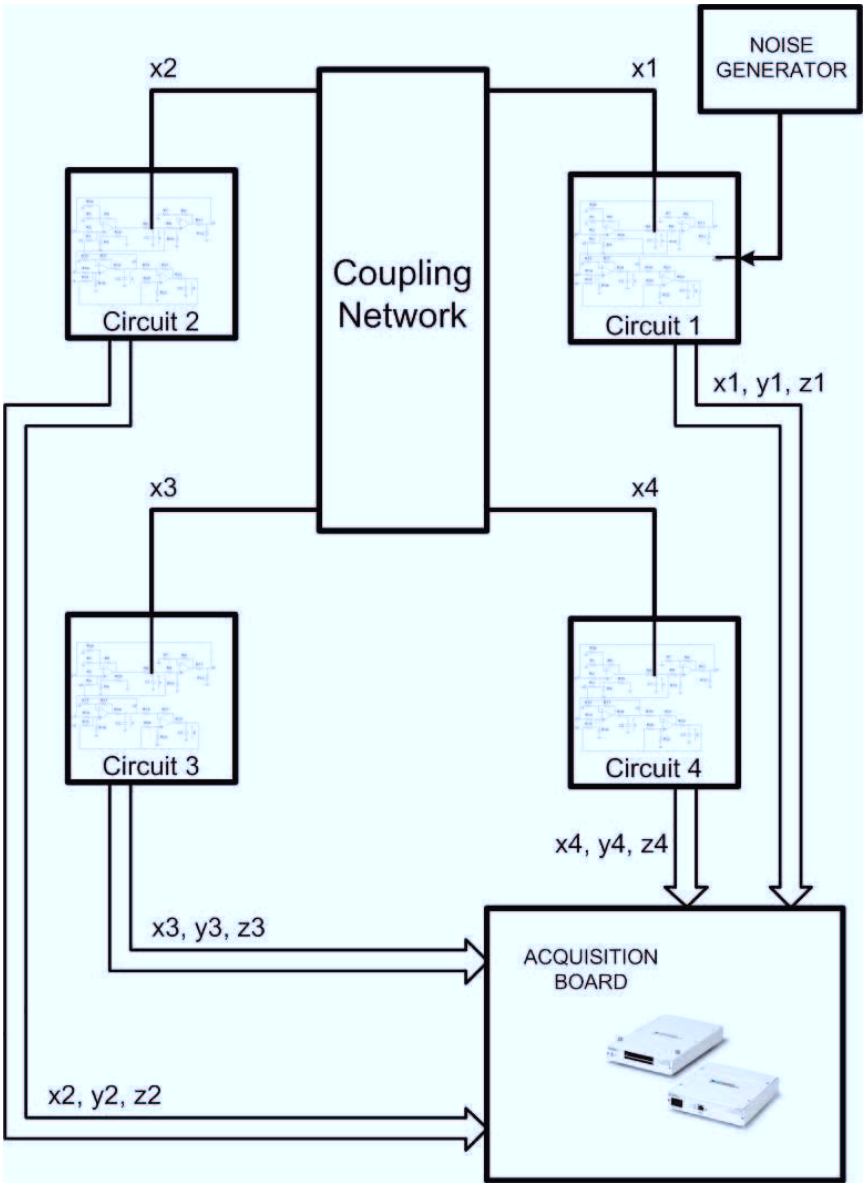
Fig. 6.11: Circuitry for a single node: the Chua's circuit.

We now describe the mathematical model of the experimental setup used. Each single isolated node obeys to the following dimensionless equations:

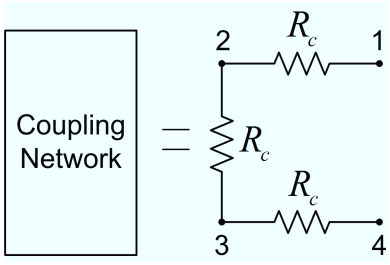
$$\begin{cases} \dot{x}_i = \alpha \left(y_i - h\left(x_i \right) \right) \\ \dot{y}_i = x_i - y_i + z_i \\ \dot{z}_i = -\beta y_i, \end{cases} \tag{6.7}$$

with $h\left(x\right) = m_1 x + 0.5\left(m_0 - m_1\right)\left(|x+1| - |x-1|\right)$, $i = 1, ..., 4$, and where x_i, y_i, and z_i represent the voltage across capacitors C_{3i-2}, C_{3i-1}, and C_{3i}.

When coupling between the network nodes is considered, one has to take into account that the coupling is bidirectional and diffusive and takes place between the variables associated to capacitors C_{3i-2} and



(a)



(b)

Fig. 6.12: (a) Schematic illustration of the experimental setup. (b) An example of coupling network: the coupling network to implement motif M2.

that noise is added to a node of the network, say circuit $\bar{i}$, so that the dimensionless equations describing the whole network are

$$\begin{cases} \dot{x}_i = \alpha\left(y_i - h\left(x_i\right)\right) + \sigma \sum_{j=1}^{4} g_{ij} x_j + \xi_i \eta \\ \dot{y}_i = x_i - y_i + z_i \\ \dot{z}_i = -\beta y_i, \end{cases} \tag{6.8}$$

where g_{ij} are the coefficients of the Laplacian matrix of the graph, i.e., $g_{ij} = 1$ if there is an edge between node i and j, $g_{ij} = 0$ if not, and $g_{ij} = -k_i$ where k_i is the degree of node i. σ represents the strength of the coupling (namely, the coupling coefficient) and in our experimental setup is $\sigma = \frac{R_{18}}{R_c}$. Finally, η represents the noise applied to circuit $\bar{i}$ ($\xi_i = 1$ if $i = \bar{i}$ and $\xi_i = 0$ if $i \neq \bar{i}$).

6.2.2 Experimental results

As mentioned above, for each of the 6 four–nodes motifs, we performed a series of four acquisition campaigns with respect to increasing values of the noise level. In each of these campaign, a zero-mean Gaussian noise was applied to a single node of the network and all the state variables of the networks were acquired with a sampling rate of f_s= 70 kHz. The noise was generated by a waveform generator Agilent 33220A and the noise level was varied by changing, at steps of 200 mV, the peak–to–peak amplitude parameter in the instrument, which is equivalent to change the variance of the signal. After acquisition of the whole set of signals, synchronization is evaluated by introducing the synchronization error defined as follows. We first consider the average

(with respect to time) Euclidean distance between the state vector of circuits i and j,

$$\delta_{ij} = \left\langle \sqrt{\frac{\left(x_i\left(t\right) - x_j\left(t\right)\right)^2 + \left(y_i\left(t\right) - y_j\left(t\right)\right)^2 + \left(z_i\left(t\right) - z_j\left(t\right)\right)^2}{3}} \right\rangle, \tag{6.9}$$

where $\langle . \rangle$ denotes average with respect to time. We then define the synchronization error as follows:

$$\delta = \frac{1}{N^2} \sum_{i,j} \delta_{ij}, \tag{6.10}$$

where N is the number of network nodes. Since for each motif we consider the application of noise in each of the network nodes, we introduce the subscript h in the parameter, namely δ_h, to indicate that it has been calculated in the case in which noise was applied to node h.

Furthermore, in order to compare the six motifs among them, the average synchronization error with respect to all the possible injection nodes is also defined,

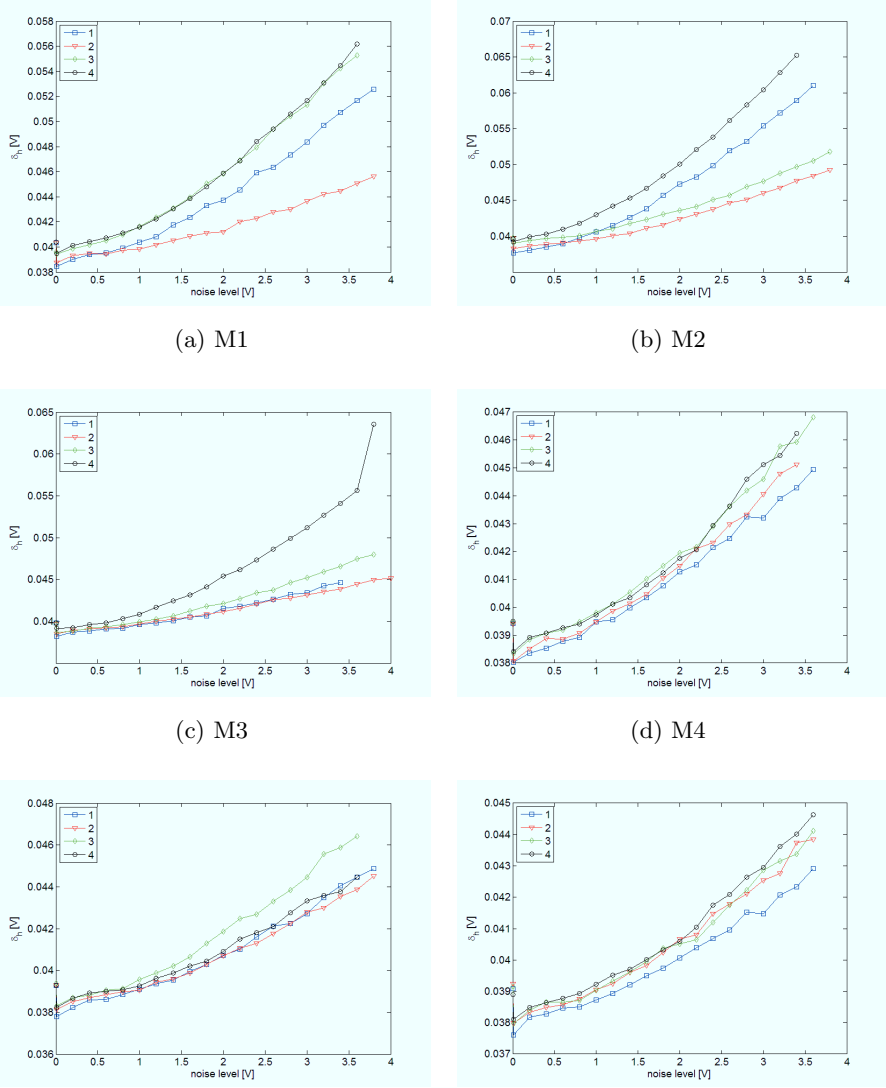
$$\Delta = \frac{1}{N} \sum_{i} \delta_h. \tag{6.11}$$

The coupling resistor was set equal for all the six network motifs in such a way that, in the absence of noise, all the networks are synchronized. In particular, a value of $R_c = 70\ \Omega$ has been considered. Fig. 6.13 reports the average synchronization error δ_h with respect to the noise level for the four–nodes network motifs. Each of the curves shown represents the average synchronization error when noise is applied to one of the four nodes of the network. As general trend, the synchronization

error grows as the noise level is increased. However, it is very interesting to note that in all the cases examined, when noise is applied to a high degree node, it deteriorates synchronization less than when it is applied to a low degree node. In fact, for motif M1 (Fig. 6.13a) node 2 has degree equal to three, while the other nodes have degree equal to one, and the best synchronization curve with respect to the noise level corresponds to the case in which noise is applied to node 2. Similarly, in motif M2 (Fig. 6.13b) synchronization is more robust to noise when this is applied to nodes 2 and 3 (having degree two), instead of nodes 1 and 4 (having degree equal to one). In motif M3 (Fig. 6.13c), node 4 has the lower performance (it has degree one, while the other have two or three). In motifs M4 and M6 (Figs. 6.13d and 6.13f), all the nodes have the same degree and exhibit very similar performance. And, finally, in motif M5 (Fig. 6.13e) nodes 2 and 4 have degree three and nodes 1 and 3 have degree two, and, indeed, all the curves have similar behavior except for node 3 which has poorer performance.

We now compare the six motifs by considering the average behavior with respect to the four different series of acquisitions performed for each of the network nodes, taking into account the parameter Δ. This parameter with respect to the noise level is shown in Fig. 6.14 for the 6 four–nodes motifs. It is worth to note that for high values of the noise level (greater than 1 V), the network motifs show different robustness to noise, with the best performance shown by motif M6 (the all-to-all network) and the worst by motif M2 (the array). For low values of the

noise level (less than 1 V), motif M2 perform slightly better than motif M1.



(a) M1 (b) M2

(c) M3 (d) M4

(e) M5 (f) M6

Fig. 6.13: Average synchronization error δ_h as a function of the noise level for the four-nodes network motifs: (a) M1; (b) M2; (c) M3; (d) M4; (e) M5; (f) M6. Each of the four curves represents the average synchronization error when noise is applied to one of the four nodes of the network.

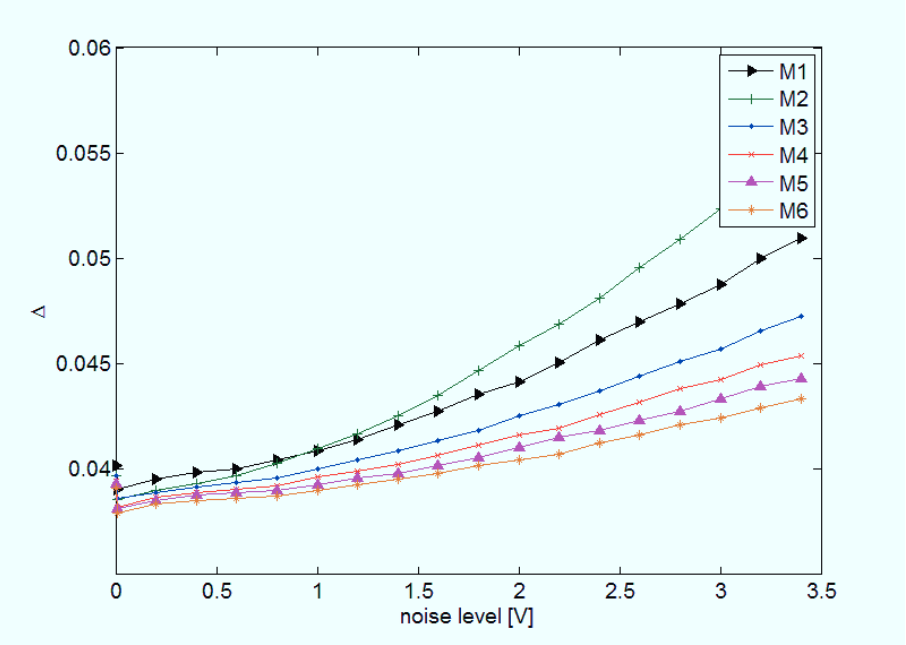


Fig. 6.14: Δ as a function of the noise level for the 6 four–nodes network motifs

7

Concluding remarks

This thesis focused on the design, implementation and investigation of new nonlinear circuits and complex phenomena in three reference frameworks (time–delay systems, CNNs and memristive systems). The idea was to exploit the peculiarities of such systems and combine them together to design new complex circuits. Noticeably, there are comparative few researches in the literature relative to the abilities of combining these systems. Starting from the problem statements and proposed objectives in Chapter 1, various novel nonlinear systems were represented sequentially through mathematical models and experimental circuits.

A general procedure for designing time–delay chaotic electronic circuit has been reviewed in Chapter 2. In order to overcome the difficulties when realizing time–delay block, the approximation of an ideal delay through a cascade of multiple second–order filters allows to design an efficient and simple circuitry. It has been shown that the use of piece–wise linear nonlinearities simplifies the implementation stage and makes possible the implementation of the whole circuits by only using simple off–the–shelf circuital components like resistors, capacitors and

operational amplifiers. In addition, the implementation of time–delay chaotic circuits with digital time–delay block has been also investigated in Chapter 3. It has been demonstrated the ease and flexibility of this approach for realizing chaotic circuits. Furthermore, the behavior of the circuits with respect to the precision and the sampling rate of the conversion process is considered through synchronization error. For future work, the procedure will be enlarged to promoting related studies on multiple time–delay chaotic systems.

Cellular Neural/Nonlinear Network based on memristive cells, the basic cell consists of an oscillating circuit made up of a capacitor, an inductor and an active memristor, are discussed in Chapter 4. Based on this system, autowaves are able to be obtained. We anticipate that our proposed system may be used as the basis to investigate other complex phenomena like spiral waves or Turing pattern formation. Moreover, it would be useful to discover new memristive CNNs by modifying the basic cell, i.e. using simple 2–element cell.

Chapter 5 has dealt with the possibility of designing memristive time-delay systems exhibiting chaotic behavior. We propose three nonlinear circuits based on memristive systems with the presence of delay. It is worth noting that chaotic behaviors are observed in very simple configurations, i.e. 2–element circuit.

Moreover, effects of noise on the chaos control of Chen's system with only one stable equilibrium and the synchronization of network motifs have studied in Chapter 6. In particular, obtained results can be generalized to apply for other similar systems.

References

1. Ahlborn, A., Parlitz, U.: Chaos control using notch filter feedback. Phys. Rev. Lett. **96**, 034,102–1–4 (2006)
2. Ando, B., Baglio, S., Graziani, S., Pitrone, N.: CNNS for noise generation in dithered transducers. In: The 17th Instrumentation and Measurement Technology Conference, IMTC 2000, pp. 1071–1076. Baltimor, MD (2000)
3. Ando, B., Graziani, S.: Stochastic resonance: Theory and applications. Kluwer, Norwel, MA (2000)
4. Arena, P., Baglio, S., Fortuna, L., Manganaro, G.: Chua's circuit can be generated by CNN cells. IEEE Trans. Circuits Syst. I Fundam. Theory Appl. **42**, 123–125 (1995)
5. Arena, P., Baglio, S., Fortuna, L., Manganaro, G.: How state controlled CNNs generate the dynamics of the Colpitts–like oscillator. IEEE Trans. Circuits Syst. I Fundam. Theory Appl. **43**, 602–605 (1996)
6. Arena, P., Baglio, S., Fortuna, L., Manganaro, G.: Self–organization in a two–layer CNN. IEEE Trans. Circuits Syst. I Fundam. Theory Appl. **45**, 157–162 (1998)
7. Arena, P., Caponetto, R., Fortuna, L., Manganaro, G.: Cellular neural networks to explore complexity. Soft Computing **1**, 120–136 (1997)
8. Arena, P., Fortuna, L., Frasca, M.: Attitude control in walking hexapod robots: An analogic spatio–temporal approach. Int. J. Circ. Theor. Appl. **30**, 349–362 (2002)
9. Atherton, D.P.: Nonlinear control engineering. Van Nostrand Reinhold Co. Ltd., Wokingham, UK (1982)
10. Barboza, R., Chua, L.O.: The four–element Chua's circuit. Int. J. Bif. Chaos **18**, 943–955 (2008)

11. Barnerjee, T., Biswas, D., Sarkar, B.C.: Design and analysis of a first order time–delayed chaotic system. Nonlinear Dyn. **70**, 721–734 (2012)
12. Bashkirtseva, I., Chen, G., Ryashko, L.: Analysis of noise–induced transitions from regular to chaotic oscillations in the Chen system. Chaos **22**, 033,104–1–9 (2012)
13. Bendixson, I.: Sur les courbes definies par des equations differentielles. Acta Mathematica **24**, 1–88 (1901)
14. Bergnera, A., Frasca, M., Sciuto, G., Buscarino, A., Ngamga, E., Fortuna, L., Kurths, J.: Remote synchronization in network motifs. Phys. Rev. E **85**, 026,208–1–7 (2012)
15. Boccaletti, S., Grebogi, C., Lai, Y.C., Mancini, H., Maza, D.: The control of chaos: theory and applications. Phys. Reports **329**, 103–197 (2000)
16. Boccaletti, S., Kurths, J., Osipov, G., Valladares, D.L., Zhou, C.S.: The synchronization of chaotic systems. Phys. Rep. **366**, 1–101 (2002)
17. Boccaletti, S., Latora, V., Moreno, Y., Chavez, M., Hwang, D.U.: Complex networks: Structure and dynamics. Phys. Rep. **424**, 175–308 (2006)
18. Bocci, V., Chiodi, G., Iacoangeli, F., Nobrege, R., Pinci, D., Rinaldi, W.: Time–multiplexing of signal using highly integrated digital delay: an FPGA implementation. In: IEEE Nuclear science symposium conference record, pp. 398–402. Rome, Italy (2005)
19. Buscarino, A., Fortuna, L., Frasca, M.: Experimental robust synchronization of hyperchaotic circuits. Physica D **238**, 1917–1922 (2009)
20. Buscarino, A., Fortuna, L., Frasca, M., Gambuzza, L.V., Sciuto, G.: Memristive chaotic circuits based on cellular nonlinear networks. Int. J. Bif. Chaos **22**, 1250,070–1–13 (2012)
21. Buscarino, A., Fortuna, L., Frasca, M., Sciuto, G.: Design of time–delay chaotic electronic circuits. IEEE Trans. Circuits Syst. I Regular Papers **58**, 1888–1896 (2011)
22. Buscarino, A., Fortuna, L., Frasca, M., Sciuto, G.: Harmonic balance method for time–delay chaotic systems design. In: The 18th World Congress of the International Federation of Automatic Control, IFAC 2011, pp. 5112–5117. Milan, Italy (2011)
23. Cai, C., Xu, Z., Bo, W., Feng, B.: Notch filter feedback control in a class of chaotic systems. Automatica **38**, 695–701 (2002)
24. Chang, H.H., Liu, S.I.: A wide–range and fast–locking all–digital cycle–controlled delay–locked loop. IEEE J. Solid–State Circuits **40**, 661–670 (2005)

25. Chen, G.: Controlling Chua's global unfolding circuit family. IEEE Trans. Circuit Syst. I **40**, 829–832 (1993)
26. Chen, G., Dong, X.: From chaos to order: methodologies, perspectives and applications. World Scientific, Singapore (1998)
27. Chua, L.O.: Memristor–The missing circuit element. IEEE Trans. Circuit Theory **18**, 507–519 (1971)
28. Chua, L.O.: Resistance switching memories are memristors. Appl. Phys. A **102**(4), 765–783 (2011)
29. Chua, L.O., Hasler, M., Moschytz, G.S., Neirynck, J.: Autonomous cellular neural networks: A unified paradigm for pattern formation and active wave propagation. IEEE Trans. Circuits Syst. I Fundam. Theory Appl. **42**, 559–577 (1995)
30. Chua, L.O., Kang, S.M.: Memristive devices and system. Proc. IEEE **64**, 209–223 (1976)
31. Chua, L.O., Roska, T.: Cellular neural networks and visual computing. Cambridge University Press, Cambridge (2002)
32. Chua, L.O., Yang, L.: Cellular neural networks: Application. IEEE Trans. Circuits Syst. I Fundam. Theory Appl. **35**, 1273–1290 (1988)
33. Chua, L.O., Yang, L.: Cellular neural networks: Theory. IEEE Trans. Circuits Syst. I Fundam. Theory Appl. **35**, 1257–1272 (1988)
34. Cuomo, K.M., Oppenheim, A.V.: Circuit implementation of synchronized chaos with applications to communications. Phys. Rev. Lett. **71**, 65–68 (1993)
35. Daryanani, G.: Principles of active network synthesis and design. John Wiley & Sons, Singapore (1976)
36. Dean, J.H.B.: Modeling the dynamics of nonlinear inductor circuits. IEEE Trans. Magn. **30**, 2795–2801 (1994)
37. Driscoll, T., Quinn, J., Klien, S., Kim, H.T., Kim, B.J., Pershin, Y.V., Ventra, M.D., Basov, D.N.: Memristive adaptive filters. Appl. Phys. Lett **97**, 093,502 (2010)
38. Duan, S., Wang, L.: A novel delayed chaotic neural model and its circuitry implementation. Computers Maths. Appl. **57**, 1736–1742 (2009)
39. Ellner, S.P., Turchin, P.: When can noise induce chaos and why does it matter: a critique. Oikos **111**, 620–631 (2005)
40. Farmer, J.D.: Chaotic attractor of an infinite–dimensional dynamical system. Physica D **4**, 366–393 (1982)

41. Fortuna, L., Arena, P., Balya, D., Zarandy, A.: Cellular neural networks: a paradigm for nonlinear spatio-temporal processing. Circuits Syst. Magazine **1**, 6–21 (2001)
42. Fortuna, L., Frasca, M.: Experimental synchronization of single–transistor–based chaotic circuits. Chaos **17**, 043,118–1–5 (2007)
43. Fortuna, L., Frasca, M., Xibilia, M.G.: Chua's circuit implementation: Yesterday, today and tomorrow. World Scientific, Singapore (2009)
44. Gao, J.B., Hwang, S.K., Liu, J.M.: When can noise induce chaos? Phys. Rev. Lett. **82**, 1132–1135 (1999)
45. Genesio, R., Tesi, A.: Harmonic balance methods for the analysis of chaotic dynamics in nonlinear systems. Automatica **28**, 531–548 (1992)
46. Gomez-Gardenes, J., Gomez, S., Arenas, A., Moreno, Y.: Explosive synchronization transitions in scale–free networks. Phys. Rev. Lett. **106**, 128,701 (2011)
47. Gomez-Gesteira, M., deCastro, M., Perez-Villar, V., , Chua, L.O.: Experimental Chua's circuit arrays as an autowave simulator. IEEE Trans. Circuits Syst. I Fundam. Theory Appl. **46**, 495–499 (1999)
48. Goras, L., Chua, L.O., Leenaerts, D.M.W.: Turing patterns in CNNs–Part I: Once over lightly. IEEE Trans. Circuits Syst. I Fundam. Theory Appl. **42**, 602–611 (1995)
49. Grekhova, M.T.: Autowave processes in systems with diffusion. Gorky, Acad. Sci. USSR (1981)
50. Gunay, E.: A new autonomous chaos generator form state controlled–cellular neural networks. Int. J. Bif. Chaos **22**, 1250,069–1–10 (2012)
51. Horbelt, W., Timmer, J., Voss, H.U.: Parameter estimation in nolinear delayed feedback systems from noisy data. Phys. Lett. A **299**, 513–521 (2002)
52. Ikeda, K., Matsumoto, K.: High–dimensional chaotic behaviour in system with time–delayed feedback. Physica D **29**, 223–235 (1987)
53. Itoh, M., Chua, L.O.: Memristor oscillators. Int. J. Bif. Chaos **18**, 3183–3206 (2008)
54. Itoh, M., Chua, L.O.: Memristor cellular automata and memristor discrete time cellular neural networks. Int. J. Bif. Chaos **19**, 3605–3656 (2009)
55. Kapitaniak, T.: Synchronization of chaos using continuous control. Phys. Rev. E **50**, 1642–1644 (1994)
56. Kilinc, S., Yalcin, M., Ozoguz, S.: Multiscroll chaotic attractors from a hysteresis based time–delay differential equation. Int. J. Bif. Chaos **20**, 3275–3281 (2010)

57. Kilts, S.: Advanced FPGA Design: Architecture, Implementation, and Optimization, 1nd edn. Wiley–IEEE Press, New Jersey (2007)
58. Lakshmanan, M., Murali, K.: Experimental chaos from nonautonomous electronic circuits. Philosophical Transactions: Physical Sciences Philosophical Transactions: Physical Sciences **353**, 33–46 (1995)
59. Landa, P.S., McClintock, P.V.E.: Changes in the dynamical behavior of nonlinear systems induced by noise. Phys. Rep. **323**, 1–80 (2000)
60. Le, L.B., Konishi, K., Hara, N.: Design and experimental verification of multiple delay feedback control for time–delay nonlinear oscillator. Nonlinear Dyn. **67**, 1407–1418 (2012)
61. Lehtonen, E., Laiho, M.: CNN using memristors for neighborhood connections. In: Proc IEEE Int. Workshop Cellular Nanoscale Networks Appl. (CNNA'2010), pp. 1–4 (2010)
62. Lindberg, E., Murali, K., Tamasevicius, A.: The smallest transistor–based nonautonomous chaotic circuit. IEEE Trans. Circuits Syst. II, Exp. Briefs **52**, 661–664 (2005)
63. Linsay, P.S.: Period doubling and chaotic behavior in a driven anharmonic oscillator. Phys. Rev. Lett. **47**, 1349–1352 (1981)
64. Lodato, I., Boccaletti, S., Latora, V.: Synchronization properties of network motifs. Europhys. Lett. **78**, 28,001–p1–p2 (2007)
65. Lu, H., He, Z.: Chaotic behavior in first–order autonomous continuous–time systems with delay. IEEE Trans. Circuits Sys. I: Fundam. Theory Appl. **43**, 700–702 (1996)
66. Mackey, M.C., Glass, L.: Oscillation and chaos in physiological control system. Science **197**, 287–289 (1977)
67. Madan, R.: Chua's Circuit: A Pradigm for Chaos. World Scientific, Singapore (1993)
68. Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D., Alon, U.: Network motifs: simple building blocks of complex networks. Science **298**, 824–827 (2002)
69. Mishra, D., Ucar, A.: Pspice realization of the Ucar prototype delay model. Int. J. Non. Sci. Num. Sim. **7**, 209–214 (2006)
70. Monmansson, E., Cirstea, M.N.: FPGA design methodology for industrial control system–A review. IEEE Trans. Ind. Electron. **54**, 1824–1842 (2007)
71. Muthuswamy, B.: Implementing memristor based chaotic circuits. Int. J. Bif. Chaos **20**, 1335–1350 (2010)

72. Muthuswamy, B., Chua, L.O.: Simplest chaotic circuits. Int. J. Bif. Chaos **20**, 1567–1580 (2010)
73. Namajunas, A., Pyragas, K., Tamasevicius, A.: An electronic analog of the Mackey–Glass system. Phys. Lett. A **201**, 42–46 (1995)
74. Niculescu, S.I., Verriest, E.I., Dugard, L., Dion, J.M.: Stability and control of time–delay systems, *Lecture notes in control and information sciences*, vol. 228, chap. Stability and robust stability of time–delay systems: A guided tour, pp. 1–71. Springer (1998)
75. Ott, E., Grebogi, C., Yorke, J.A.: Controlling chaos. Phys. Rev. Lett. **64**, 1196–1199 (1990)
76. Parayandeh, A., Prodic, A.: Programmable analog–to–digital converter for low–power DC–DC SMPS. IEEE Trans. Power Electron. **23**, 500–505 (2008)
77. Perez-Munuzuri, V., Perez-Munuzuri, V., Perez-Villar, V., Chua, L.O.: Spiral waves on a 2–D array of nonlinear circuits. IEEE Trans. Circuits Syst. I Fundam. Theory Appl. **40**, 872–877 (1993)
78. Perez-Munuzuri, V., Perez-Villar, V., Chua, L.O.: Autowaves for image processing on a two–dimensional CNN array of excitable nonlinear circuits: Flat and wrinkled labyrinths. IEEE Trans. Circuits Syst. I Fundam. Theory Appl. **40**, 174–181 (1993)
79. Pershin, Y.V., Fontaine, S.L., Ventra, M.D.: Memristive model of amoeba learning. Phys. Rev. E **80**, 021,926 (2009)
80. Piper, J.R., Sprott, J.C.: Simple autonomous chaotic circuits. IEEE Trans. Circuits Syst. II, Exp. Briefs **57**, 730–734 (2010)
81. Pivka, L.: Autowaves and spatio–temporal chaos in CNNs–Part I: A tutorial. IEEE Trans. Circuits Syst. I Fundam. Theory Appl. **42**, 638–649 (1995)
82. Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P.: Numerical Recipes: The Art of Scientific Computing, 3nd edn. Cambridge University Press, New York (2007)
83. Pyragas, K.: Continuous control of chaos by self–controlling feedback. Phys. Lett. A **170**, 421–428 (1992)
84. Pyragas, K., Tamasevicius, A.: Experimental control of chaos by delayed self–controlling feedback. Phys. Lett. A **180**, 99–102 (1993)
85. Richert, M., Whitmer, D.: Chaotic dynamics of RLD oscillator. In: Report of Biophysical Measurements Lab, pp. 1–28. Univ. Cali., San Diego (2003)

86. Roger Woods John MacAllister, G.L., Li, Y.: FPGA–based implementation of complex signal processing systems, 1nd edn. Wiley, Wiltshire (2008)
87. Sanjuan, M.A.F., Grebogi, C.: Recent Progress in Controlling Chaos. World Scientific, Singapore (2010)
88. Scholl, E., Schuster, H.G.: Handbook of Chaos Control. Wiley, New York (2007)
89. Sedra, A.S., Smith, K.C.: Microelectronic circuits. Oxford University Press, Oxford, U. K. (2003)
90. Seferiadis, G., Pouchet, M., Gough, M.P.: FPGA implementation of a delay–line readout system for a particle detector. Measurement **39**, 90–99 (2006)
91. Shilnikov, L.P., Shilnikov, A.L., Turaev, D.V., Chua, L.O.: Methods of Qualitative Theory in Nonlinear Dynamics. World Scientific, Singapore (1998)
92. Shin, S., Kim, K., Kang, S.M.: Memristor applications for programmable analog ICs. IEEE Trans. Nanotechnology **410**, 266–274 (2011)
93. Slotine, J.J.E., Li, W.: Applied nonlinear control. Prentice-Hall International Editions, London, U.K. (2002)
94. Sprott, J.C.: Some simple chaotic flows. Phys. Rev. E **50**, R647–650 (1994)
95. Sprott, J.C.: A simple chaotic delay differential equation. Phys. Lett. A **366**, 397–402 (2007)
96. Sprott, J.C.: Elegant chaos: algebraically simple chaotic flows. World Scientific, Singapore (2010)
97. Srinivasan, K., Mohamed, I.R., Murali, K., Lakshmanan, M., Sinha, S.: Design of time delayed chaotic circuit with threshold controller. Int. J. Bif. Chaos **20**, 2185–2191 (2010)
98. Srinivasan, K., Senthilkumar, D.V., Murali, K., Lakshmanan, M., Kurths, J.: Synchronization transitions in coupled time-delay electronic circuits with a threshold nonlinearity. Chaos **21**, 023,119 (2011)
99. Strukov, D.B., Snider, G.S., Stewart, D.R., Williams, R.S.: The missing memristor found. Nature **453**, 80–83 (2008)
100. Sun, C., Lin, Y., Han, M.: Stability and Hopf bifurcation for an epidemic disease model with delay. Chaos, Solitons and Fractals **30**, 204–216 (2006)
101. Sun, Y., Zhao, D.: Effects of noise on the outer synchronization of two unidirectionally coupled complex dynamical networks. Chaos **22**, 023,131–1–10 (2012)

102. Sun, Z., Yang, X.: Generating and enhancing lag synchronization of chaotic systems by white noise. Chaos **21**, 033,114–1–10 (2011)
103. Tamasevicius, A., Mykolaitis, G., Bumeliene, S.: Delayed feedback chaotic oscillator with improved spectral characteristic. Electron. Lett. **42**, 736–737 (2006)
104. Tchitnga, R., Fotsin, H.B., Nana, B., Fotso, P.H.L., Woafo, P.: Hartlay's oscillator: The simplest chaotic two–component circuit. Chaos Solit. Fract. **45**, 306–313 (2012)
105. Tour, J.M., He, T.: The fourth element. Nature **453**, 42–43 (2008)
106. Ucar, A.: A prototype model for chaos studies. Int. J. Eng. Sci. **40**, 251–258 (2002)
107. Ucar, A.: On the chaotic behaviour of a prototype delayed dynamical system. Chaos Solitions Fractals **16**, 187–194 (2003)
108. Ucar, A.: Synchronization in complex networks. Phys. Rep. **469**, 93–153 (2008)
109. Vega, Y.M., Vazquez-Prada, M., Pacheco, A.F.: Fitness for synchronization of network motifs. Physica A **343**, 279–287 (2004)
110. Voss, H.U.: Real–time anticipation of chaotic states of an electronic circuit. Int. J. Bif. Chaos **12**, 1619–1625 (2002)
111. Wang, L., Yang, X.: Genration of multi–scroll delayed chaotic oscillator. Electron. Lett. **42**, 1439–1441 (2006)
112. Wang, X., Chen, G.: Symmetrical multi–petal chaotic attractors in a 3D autonomous system with only one stable equilibrium. In: The 4th International Workshop on Chaos-Fractals Theories and Applications, IWCFTA 2011, pp. 82–85. Hangzhou, China (2011)
113. Wang, X., Chen, G.: A chaotic system with only one stable equilibrium. Commun. Nonlinear Sci. Numer. Simulat. **17**, 1264–1271 (2012)
114. Wang, X., Chen, G.: Constructing a chaotic system with any number of equilibria (2012). http://arxiv.org/abs/1201.5751.
115. Wei, Z., Yang, Q.: Dynamical analysis of a new autonomous 3–D chaotic system only with stable equilibria. Nonlinear Analysis: Real World Applications **12**, 106–118 (2011)
116. Wei, Z., Yang, Q.: Dynamical analysis of the generalized Sprott C system with only two stable equilibria. Nonlinear Dyn. **68**, 543–554 (2012)
117. Wolf, A., Swift, J.B., Swinney, H.L., Vastano, J.A.: Determining Lyapunov exponents from a time series. Physica D **16**, 285–317 (1985)

118. Xia, Y., Fu, M., Shi, P.: Analysis and synthesis of dynamical systems with time–delays. Springer, New York (2009)
119. Yalcin, M.E., Ozoguz, S.: n–scroll chaotic attractors from a first–order time–delay differential equation. Chaos **17**, 033,112–1–8 (2007)
120. Yalcin, M.E., Suykens, J.A.K.: Spatiotemporal pattern formation on the ACE16k CNN chip. Int. J. Bif. Chaos **16**, 1537–1546 (2006)
121. Yalcin, M.E., Suykens, J.A.K., Vandewalle, J.: True random bit generation from a double–scroll attractor. IEEE Trans. Circuits Syst. I, Regular Papers **51**, 1395–1404 (2004)
122. Yang, Q.G., Wei, Z., Chen, G.R.: A unusual 3D autonomous quadratic chaotic system with two stable node–foci. Int. J. Bif. Chaos **20**, 1061–1083 (2010)
123. Yeniceri, R., Yalcin, M.E.: An implementation of 2D locally coupled relaxation oscillators on an FPGA for real–time autowave generation. In: Proc IEEE Int. Workshop Cellular Nanoscale Networks Appl. (CNNA'2008), pp. 29–33 (2008)
124. Zhang, R., de S. Cavalcante, H.L.D., Gao, Z., Gauthier, D.J., Socolar, J.E.S., Adams, M.M., Lathrop, D.P.: Boolean chaos. Phys. Rev. E **80**, 045,202 (2009)
125. Zhang, X., Cui, Z., Zhu, Y.: Synchronization and circuit experiment simulation of chaotic time–delay systems. In: Proc. Pacific–Asia Conf. Circuits, Commun. Syst., pp. 781–784. Chengdu, China (2009)

A

Personal publication

Buscarino, A., Fortuna, L., Frasca, M., Iachello, M., Pham, V.–T.: Robustness to noise in synchronization of network motifs: Experimental results. Chaos **22**, 043106–1–9 (2012)

Buscarino, A., Fortuna, L., Frasca, M., Gambuzza, L.V., Pham, V.–T.: Chaos control in a chaotic system with only one stable equilibrium point. In: ICNAAM2012, pp. 691–694. Kos, Greece (2012)

Pham, V.–T., Buscarino, A., Fortuna, L., Frasca, M.: Simple memristive time–delay chaotic systems. Int. J. Bif. Chaos, submitted (2012)

Pham, V.–T., Buscarino, A., Fortuna, L., Frasca, M., Hoang, T.M.: FPGA–based generation of autowaves in memristive cellular neural networks. In: CNNA2012, pp. 1–6. Turin, Italy (2012)

Pham, V.–T., Buscarino, A., Fortuna, L., Frasca, M.: Autowaves in memristive cellular neural networks. Int. J. Bif. Chaos **22**, 1230027–1–9 (2012)

Pham, V.–T., Fortuna, L., Frasca, M.: Implementation of chaotic circuits with a digital time–delay block. Nonlinear Dyn. **68**, 345–355 (2012)

Pham, V.–T., Fortuna, L., Frasca, M., Anh, T.T., Hoang, T.M.: Realization of synchronization of coupled multiple delay systems on FPGA platform. In: INDS2011, pp. 52–56. Klagenfurt, Austria (2011)

Fortuna, L., Frasca, Pham, V.–T.: An implementation of FPGA–based delay circuit. In: Dynamics Days Europe 2010, Bristol, United Kingdom (2010)

List of Figures