Hardware Implementation of a Chaotic Circuit Based on a Memristor

Juan Polo, Hans López and Cesar Hernández

Abstract — **The utilization of a Memristor in a chaotic circuit paves the way for random number generation. Across other research, authors detected, through simulation tools, the presence of monofractal/multifractal behavior of a memristor-based random number generation. Despite the novelty of the finding, the logical step to follow requires the hardware implementation of the memristor and the chaotic circuit that uses it to investigate the details, limitations, and difficulties not contemplated in the advanced simulations. In light of this objective, this article systematically reviews five memristor emulator circuits, comparing them in terms of cost and frequency of operation. This study proposes a memristor emulator circuit that uses a wave rectifier based on operational amplifiers as a nonlinear element. This memristor and a corresponding reference unit were utilized in a hardware implementation of a chaotic Chua-based circuit. The modifications that are made in conjunction with the measurements made in the laboratory provide evidence that the chaotic system obtained satisfies the essential conditions to be used in a future generator otherwise implemented in hardware that exhibits monofractal/multifractal behavior.**

Index Terms— **Attractor, Chaotic Circuit, Emulation, Entropy, Hardware, Memristor.**

I. INTRODUCTION

IN electric circuits, four basic variables are normally used: current (i), voltage (v), magnetic flow (ϕ), and electric charge (q). These variables are used in the definition of the three passive elements (resistance, capacitance, and inductance) and the expressions of current in terms of charge or voltage in terms of magnetic flow. Nonetheless, no expression relates flow and charge through passive elements [1]. In 1971, Chua analyzed the lack of an expression that establishes a relationship between these two variables. As a result, the author proposed a fourth passive element called the memristor, whose behavior marks this relationship [2].

The name of the memristor is derived from its being resistance to memory, and the relationship between the magnetic flow and the electric charge is characteristically nonlinear, which sets it apart from other passive elements.

This can be observed on the $v - i$ plane. The variable controlling the memristor's behavior, there are two possible classifications. Equation (1) describes when the charge controls the voltage, and (2) expresses when the flow controls the current.

$$v(t) = M\big(q(t)\big)i(t)$$
$$M(q) = \frac{d\phi(q)}{dq} \tag{1}$$

$$i(t) = W\big(\phi(t)\big)v(t)$$
$$W(\phi) = \frac{dq(\phi)}{d\phi} \tag{2}$$

$M(q)$ and $W(\phi)$ are respectively measured in Ohm (Ω) and Siemens (S), with positive values (over zero), and are known as incremental memristance and incremental conductance. Chua proposed three intrinsic properties of the memristor that serve as identification criteria, evidenced in the $v - i$ plane with a sinusoidal input signal to determine whether a device is a memristor.

1. A pinched hysteresis loop must be set in the $v - i$ plane that corresponds to an incremental memristance that goes through the origin. The current and the voltage are only equal at the origin.
2. A reversely proportional relationship must be established between the input signal frequency and the area of the pinched hysteresis loop lobes.
3. The increase in the input signal frequency should lead to a progressive loss of the non-linearity of the device, making it behave like a regular resistor.

Despite establishing certain features of the memristor, the research made by Chua served as a conceptual tool, given that a device with these characteristics had not been discovered. The idea consisted of emulating its behavior through mutator circuits [2]. It was not until 2008 that the first passive device with memristor characteristics was presented by HP Labs, based on a structure comprised of nanoscopic TiO2 sheets with platinum electrodes. The proposed structure is described in Fig. 1.
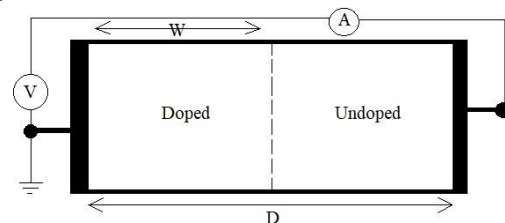

Fig. 1. Memristor structure from HP Labs (Redrawing based on [3])

Equation (3) describes the voltage over the memristor in terms of the doped region width $w(t)$ determined by the charge.

$$v(t) = (R_{ON}\frac{w(t)}{D} + R_{OFF}(1 - \frac{w(t)}{D}))i(t) \qquad (3)$$
$$w(t) = \mu_V \frac{R_{ON}}{D}q(t)$$

The expression of memristance can be simplified, particularly considering that $R_{ON} \ll R_{OFF}$, resulting in (4).

$$M(q) = R_{OFF}(1 - \mu_V \frac{R_{ON}}{D^2}q(t)) \qquad (4)$$

Although the proposition of HP Labs is a new passive element, it cannot be used in multiple applications given its high manufacturing costs, which makes it a non-commercially viable product; this is the same case for memristors based on hafnium oxide $HfO2$ [3] or graphene-based memristors. Consequently, emulator circuits need to be created using low-cost commercial electronics.

Conversely, [4] designed and implemented a chaotic circuit using the memristor, evidencing the application of this as an RNG. While the utilization of the memristor for random number generation is not novel, the possibility of using it as a source of truly random numbers (because it is a nonlinear device sensitive to initial conditions) in conjunction with a versatile structure would allow adjusting the Hurst parameter, choosing mono-fractality/multifractality, or adjusting the width of the multifractal spectrum, such evidence had not been encountered within the studied literature. we uncovered an article that developed a founding for that theory [4].

Specifically, in [4] the monofractal/multifractal, behavior of a memristor-based RNG was ascertained, and it was evidenced that it is possible to modify its characteristics and shown what could be done to achieve it. Nevertheless, in this research solely the equations that describe the behavior of the memristor were used in a simulation scenario, leaving open the expectation of the details, limitations, and difficulties, which brings an implementation in hardware.

The goal of this work is to implement a hardware-oriented memristor circuit that meets the properties of memristance exposed in the literature and assess the performance of the memristor in a chaotic circuit that serves as an entropy source for the future implementation of a memristance-based random number generator.

This work is organized into six sections: Section 2 describes the emulation process of the memristor. Section 3 details the proposed emulator circuit for the memristor. Section 4 describes the memristor-based chaotic circuits. Section 5 discusses the implementation of the memristor-based chaotic circuit. Section 6 presents applications of the chaotic circuits. Lastly, Section 7 presents a set of conclusions on the overall work.

II. MEMRISTOR EMULATION

There are various techniques to emulate a memristor. Typically, however, emulation models are based on two basic structures. The first emulation model computes the integral of the input signal (voltage or current), passes it through a function that introduces the nonlinear nature of the memristor, and then computes the derivative, which is the output (voltage or current). The block diagram can be seen in Fig. 2.
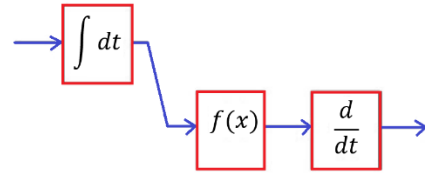

Fig. 2. First emulation model of the memristor (Redrawing based on [1])

The second model carries out the same procedure as the first two blocks. Although, instead of computing the derivative of the resulting signal, it multiplies it with the input value to deliver the voltage or current output. The block diagram is presented in Fig. 3.
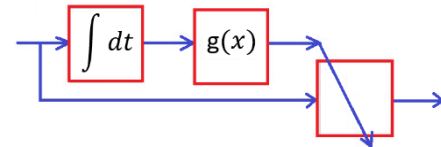

Fig. 3. Second emulation model of the memristor (Redrawing based on [1])

To identify the potential factors necessary for implementing memristor emulators using the models described earlier and in other contexts, we examined several references [5]–[9], focusing on the components utilized, operational parameters, and associated implementation expenses in US dollars. The ALTIUM schematic editing software is used, including its connection with vendor databases, to determine the availability and cost of the components needed in the circuits proposed by [5]–[9].

In [5], a grounded memristor is proposed that uses the model shown in Fig. 2 as the emulation structure. It implements a hyperbolic sinus function based on exponential amplifiers setting transistors as diodes and general-purpose amplifiers. This circuit can be seen in Fig. 4.
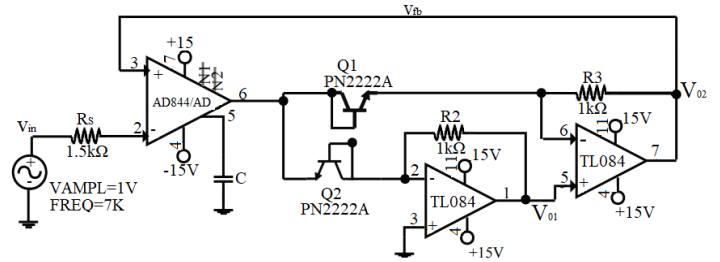

Fig. 4. Memristor proposed in [5] (Redrawing based on [5])

Defining a range for the amplitude of the input signal is not feasible as the authors of [5] exclusively discuss tests conducted with a 1V input signal. However, they present a frequency sweep revealing the memristance behavior ranging from 3 kHz to 16 kHz. Equation (5) describes the expression of the resulting memristance.

$$R_m = R_s - \frac{2I_{ES}R_D}{i_{in}}sinh(\frac{q_{in}}{CV_T}) \qquad (5)$$

Where:
I_{ES}: Reverse-saturation current
V_T: Voltage Threshold

The simplification process of the memristance equation involves a hyperbolic sine function derived from its Taylor series, which results in the linear form of (6).

$$R_m = R_s - \frac{2I_{ES}R_D}{CV_T}\left(\frac{q_{in}}{i_{in}}\right) \tag{6}$$

The authors [5] performed a simulation of the proposed strategy in PSPICE and implemented the memristor emulator in hardware form. This successfully proving the compliance of the characteristics of the memristor that were described. The total implementation cost is 19.29 USD.

Similarly, [6] conceived an emulator based on the structure in Fig. 2 that uses an additional Current Feedback Operational Amplifier (CFOA) and involves a hyperbolic cosine function. The corresponding circuit is shown in Fig. 5.
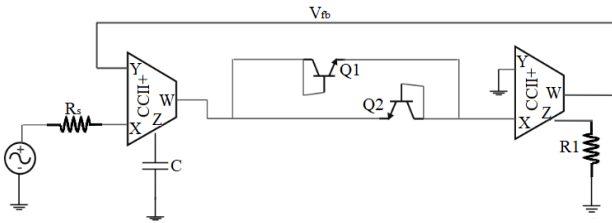

Fig. 5. Memristor proposed in [6] (Redrawing based on [6])

The findings presented by the authors lack discussion on testing with input signals of amplitudes other than 1 V, thus precluding the definition of a specific range for the input amplitude. Concerning the frequency of the input signal, the authors assert that the emulator operates effectively within the range of 700 Hz to 3 kHz.

Equation (7) represents the expression for the memristance utilizing the hyperbolic cosine function. Similar to the previous instance, this equation can be simplified utilizing its Taylor series to yield a non-linear quadratic function, demonstrated in (8).

$$R_m = R_s + \frac{2I_S R_1}{i_{in}}\left(cosh\left(\frac{q}{CV_{TH}}\right) - 1\right) \tag{7}$$

$$R_m = R_s + \frac{I_S R_1}{C^2 V_{TH}^2}\left(\frac{q(t)^2}{i_{in}}\right) \tag{8}$$

Where:
I_S: Reverse-saturation current
V_{TH}: Threshold voltage

The scheme proposed in [5] was simulated in PSPICE and implemented in hardware by the authors. The testing shows successful compliance with the memristor characteristics. The total implementation cost is 30.83 USD.

The approach suggested in [7] employs the model depicted in Fig. 3, where the nonlinearity is depicted by a varactor diode with adjustable capacitance. A prior characterization of this component is necessary. Fig. 6 illustrates the circuit layout.

The approximation used to characterize the varactor diode is the quadratic function, resulting in (9).

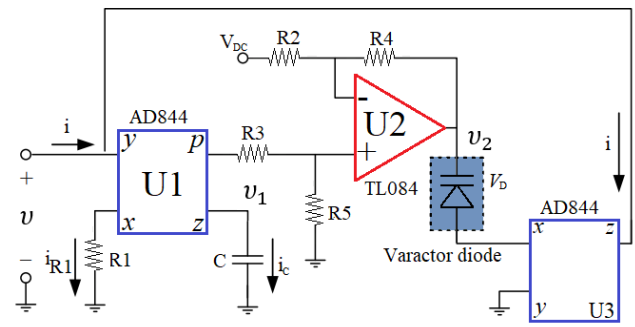$$W(\phi) = 3\lambda\phi^2 + 2k\phi + \zeta \tag{9}$$

Fig. 6. Memristor proposed in [7] (Redrawing based on [7])

The authors use a 0.3 V input signal throughout all the tests, with frequency variations between 4 and 9 kHz. The authors of [7] perform a simulation of the circuit in Multisim and implement it in hardware, confirming compliance with the memristor characteristics.

It is impossible to determine the implementation cost given that the reference for the varactor diode used by the authors is unavailable from any vendor used in ALTIUM. This strategy can help determine how certain elements with non-linear behavior can be used.

In [8], the proposed memristor does not match any of the described models and is the simplest alternative since it only uses general-purpose operational amplifiers and a light-dependent resistance (LDR) in an optocoupler serving as a non-linear function. The circuit is represented in Fig. 7.
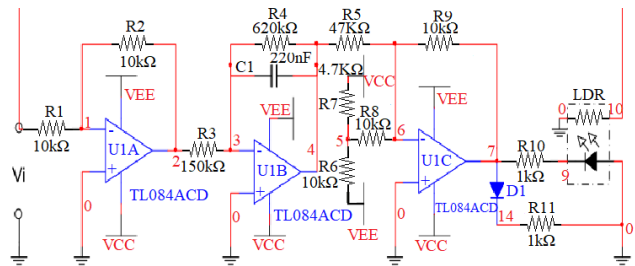

Fig. 7. Memristor proposed in [8] (Redrawing based on [8])

The LDR is restricted to a range in which the behavior can be labeled as linear, thus representing a linear inductance. Equation (10) presents the corresponding expression.

$$W(t) = -0.3\left(\frac{R_9 R_2 V cos(\omega t)}{\omega R_5 R_3 R_1 C_1} + \frac{R_9 V_5}{R_8}\right) + 0.11 \tag{10}$$

The input signal range for the memristor emulator spans from 0.4 to 0.8 V, accompanied by frequencies ranging between 0.1 and 280 Hz. However, similar to the previous scenario, implementation is hindered due to the unavailability of the specified optocoupler as used by [8]. The authors conduct a simulation of the circuit in Multisim and subsequently proceed to its hardware implementation, demonstrating the successful operation of the memristor emulator.

The work in [9] uses the structure of the model shown in Fig. 3 to effectively deliver a variable memductance based on analog multipliers and a quadratic function.
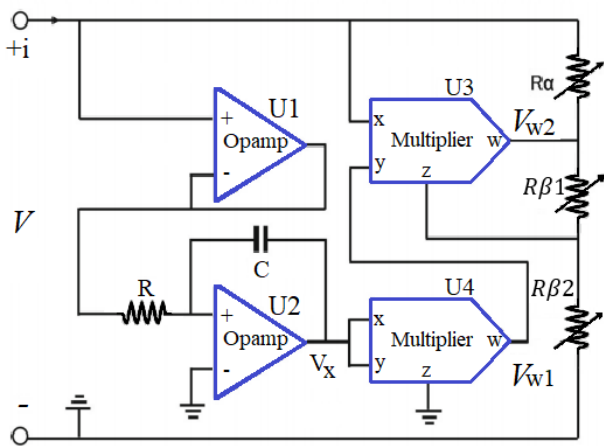
The circuit is presented in Fig. 8.

Fig. 8. Memristor proposed in [9] (Redrawing based on [9])

The authors [9] use an input signal with an amplitude ranging from 1 to 4 V and a frequency between 100 Hz and 2.2 kHz. The expression for the output memductance is presented in (11).

$$W(\phi) = \frac{1}{R_\alpha} - \frac{1}{R^2 C^2} \frac{R_{\beta 1} + R_{\beta 2}}{100 R_\alpha R_{\beta 1}} \phi^2(t) \qquad (11)$$

The authors [9] simulated with PSICE and a hardware implementation, confirming the correct behavior of the memristor emulator. The total implementation cost is 42.13 USD. The comparison of the proposed alternatives is summarized in Table I; this includes the costs and operation ranges in terms of amplitude and frequency.

TABLE I
COMPARATIVE SUMMARY OF THE MEMRISTOR ALTERNATIVES.

Proposal	Cost (USD)	Input amplitude (V)	Frequency range (Hz)
[5]	19.29	1	3000 to 16000
[6]	30.83	1	700 to 3000
[7]	-	0.3	4000 to 9000
[8]	-	0.4 a 0.8	0.1 to 280
[9]	42.13	1 a 4	100 to 2200

While the approach presented in [9] may not be the most economical option, it provides a broad spectrum in both frequency and amplitude and facilitates the alteration of the memristor equation through a resistance-based array. Consequently, this strategy is selected as a benchmark for the development of the memristor emulation circuit.

III. PROPOSED EMULATOR CIRCUIT OF THE MEMRISTOR

Before constructing the memristor emulator circuit, a reviewed circuit was selected to form the basis of the proposal, opting for a known structure previously employed for such purposes. The reference circuit from [9] was chosen and is detailed in Fig. 8.

Equation (11) corresponds to the resulting memductance expression, defined by the values of potentiometers, R_α, $R_{\beta 1}$, and $R_{\beta 2}$, whose purpose is to deliver a voltage level suitable for the OP AMP and multipliers. The value of the components used by the authors of [9] can be seen in Table II.

The strategy discussed in [9] features blocks corresponding to the emulation model of Fig. 3. The non-linear function used by the authors is a quadratic function based on analog

multipliers. To build the emulator, the non-linear function block is required to be changed.

TABLE II
CIRCUIT COMPONENTS USED IN [9].

Component	Value
R	$1.5\ k\Omega$
C	$47\ nF$
R_α	$1.5\ k\Omega$
$R_{\beta 1}$	$3\ k\Omega$
$R_{\beta 2}$	$36\ k\Omega$

This paper suggests employing a full-wave rectifier as a nonlinear function, a choice not explored in the referenced literature. This solution can be realized using a general-purpose operational amplifier (OP AMP) and low-cost high-speed commutation diodes. Fig. 9 illustrates the resulting circuit.
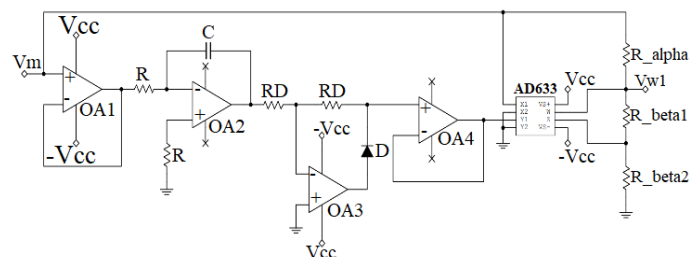

Fig. 9. Full wave rectifier (non-linear function).

The expression for the resulting memductance can be obtained by establishing a relationship between the emulator voltage and current in the form of (12).

$$i_m(t) = \frac{dq}{dt} = \frac{dq}{d\phi}\frac{d\phi}{dt} = W(\phi)v_m(t) \qquad (12)$$

This is made possible with the resistance R_α, as seen in (13).

$$i_m = \frac{v_m - v_{\omega 1}}{R_\alpha} \qquad (13)$$

The integrator output is given in terms of the emulator voltage, resulting in (14). This voltage represents the flow in the form of (15).

$$v_x = -\frac{1}{RC}\int v_m(t)dt \qquad (14)$$

$$v_x = -\frac{1}{RC}\phi(t) \qquad (15)$$

Equation (15) can be modified to represent the full-wave rectifier output in terms of the flow, resulting in (16).

$$v_f = \frac{1}{RC}|\phi(t)| \qquad (16)$$

The multiplier output has a 0.1 gain intrinsic to the device. This equation is given regarding the emulator voltage, the rectifier output, and the voltage on port z. The resulting expression is presented in (17).

$$v_{\omega 1} = \frac{v_m v_f}{10} + v_{\omega 2} \qquad (17)$$

The voltage $v_{\omega 2}$ is a fraction of the multiplier output, given in terms of the resistances $R_{\beta 1}$ and $R_{\beta 2}$, as seen in (18).

$$v_{\omega 2} = \frac{R_{\beta 2}}{R_{\beta 1} + R_{\beta 2}} v_{\omega 1} \qquad (18)$$

Replacing (18) in (17) results in the multiplier output in (19).

$$v_{\omega 1} = \frac{v_m (R_{\beta 1} + R_{\beta 2})}{10 R_{\beta 1} RC} |\phi(t)| \qquad (19)$$

This can be used in (13), leading up to an expression of the memductance stemming from the relationship between voltage and current, presented in (20). The memorability of the emulator can be expressed in the form of (21).

$$i_m = v_m \left(\frac{1}{R_\alpha} - \frac{R_{\beta 1} + R_{\beta 2}}{10 R_\alpha R_{\beta 1} RC} |\phi(t)| \right) \qquad (20)$$

$$W(\phi) = \frac{1}{R_\alpha} - \frac{R_{\beta 1} + R_{\beta 2}}{10 R_\alpha R_{\beta 1} RC} |\phi(t)| \qquad (21)$$

In order to verify that the proposed emulator aligns with the characteristics of the memristor, the circuit undergoes simulation in Multisim with a sinusoidal input of 4V. The frequency of the input signal is varied, allowing observation of the v-I curv''s behavior. Fig. 10 encapsulates the results of these tests.

Fig. 10 concludes that there is compliance with the first characteristic of the memristor given the appearance of a pinched hysteresis loop in the $v - i$ plane. The increase in frequency causes a reduction of the lobe area within the pinched hysteresis loop, showing a more linear behavior and complying with the remaining characteristics. The proposed memristor meets the previously described conditions for an emulator memristor.
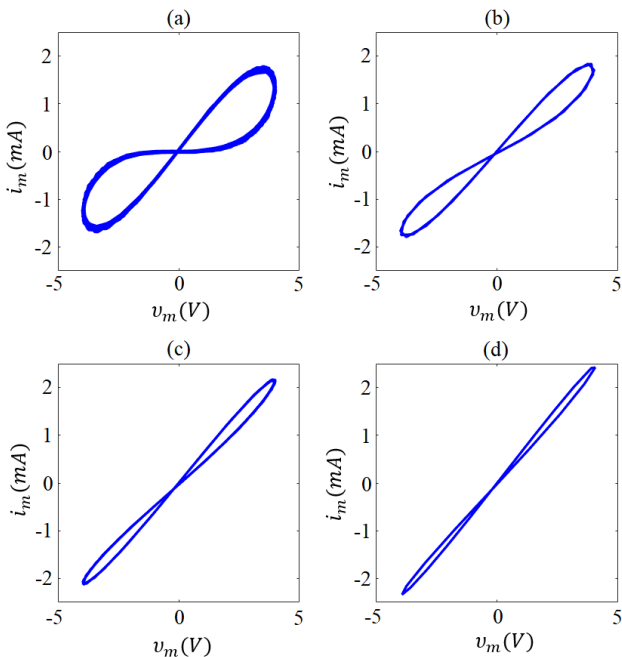

Fig. 10. v-i curves in the memristor with an absolute value function, with a frequency of (a) 4 kHz, (b) 8 kHz, (c) 15 kHz, and (d) 25 kHz.

IV. CHAOTIC CIRCUITS BASED ON MEMRISTORS

Utilizing the memristor implementation, one can construct a chaotic circuit for generating an entropy source. This entails examining the five circuits outlined in [4] to select an appropriate option for implementation. The evaluation of these five alternatives considers factors such as the type of memristor employed, the properties of the components utilized, and the component count [10].

A. Chaotic Circuit 1

The circuit proposed by [11] is the simplest chaotic circuit given that it is comprised of three elements connected in series, as seen in Fig. 11. This circuit is the third-order system represented in (22), where the variables are the capacitor voltage (v_C), the current through the inductor (i_L), and the inner variable of the memristor (x_s).
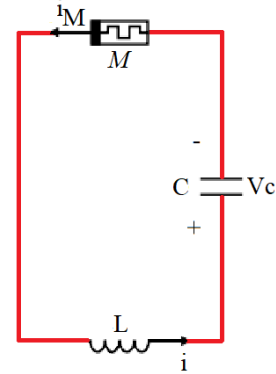

Fig. 11. Chaotic circuit from [11] (Redrawing based on [11])

$$\begin{cases} C \dfrac{dV_{C_1}}{dt} = i \\[2mm] -L \dfrac{di}{dt} = V_{C_1} + \beta (x_s^2 - 1) i \\[2mm] \dfrac{dx_s}{dt} = i + \alpha x_s - i x_s \end{cases} \qquad (22)$$

The memristance used is a charge-controlled memristor that uses a quadratic function with an inner variable to ensure the nonlinearity of the element. Due to this, the circuit cannot be implemented with the reference and proposed memristor since these are current-controlled.

B. Chaotic Circuit 2

The circuit proposed in [13] modifies the alternative proposed by [11] by adding resistance in parallel, as seen in Fig. 12. This change does not alter the order of the system and it generates a current divider in the system, as seen in (23).
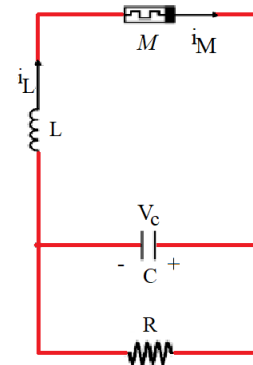

Fig. 12. Chaotic circuit proposed in [12] (Redrawing based on [12])

$$\begin{cases} C\dfrac{V_C}{dt} = i_L - \dfrac{V_C}{R} \\[2mm] L\dfrac{i_L}{dt} = -V_C - \beta(x_s^2 - 1)i_L \\[2mm] \dfrac{x_s}{dt} = -i_L - \alpha i_L + i_L x_s \end{cases} \qquad (23)$$

As seen in the case of the solution from [11], this circuit uses a current-controlled memristor, so the reference memristor and proposed memristor cannot be implemented.

C. Chaotic Circuit 3

In contrast to the previous circuits, the work of [13] describes the fourth-order system in (24). This circuit is comprised of five elements, where the only element different from the ones used in the previous cases is the negative resistance, which can be observed in Fig. 13.
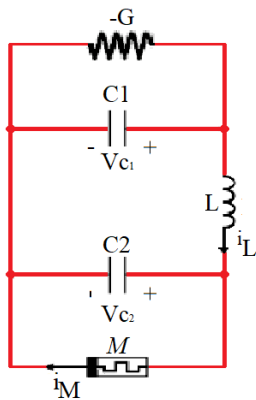

Fig. 13. Chaotic circuit from [13] (Redrawing based on [13])

$$\begin{cases} C_2\dfrac{dV_{C_2}}{dt} = i_L - W(\varphi)V_{C_2} \\[2mm] L\dfrac{i_L}{dt} = V_{C_1} - V_{C_2} \\[2mm] C_1\dfrac{dV_{C_1}}{dt} = GV_{C_1} - i_L \\[2mm] \dfrac{\varphi}{dt} = V_{C_2} \end{cases} \qquad (24)$$

The memristor used in this circuit is controlled by current, so the circuits of the reference memristor and proposed memristor can be used. In [13], the memristor used in this scenario uses a quadratic function to represent its nonlinear nature, resulting in a memristor similar to the reference memristor.

D. Chaotic Circuit 4

The circuit proposed in [14] can be seen in Fig. 14. It is comprised of seven elements that represent a fourth-order system, as seen in (25). This alternative uses the same elements as in the previous cases.

$$\begin{cases} C_2\dfrac{dV_{C_2}}{dt} = \dfrac{V_{C_1} - V_{C_2}}{R} + GV_{C_2} - W(\varphi)V_{C_2} \\[2mm] C_1\dfrac{dV_{C_1}}{dt} = \dfrac{V_{C_2} - V_{C_1}}{R} - i_L \\[2mm] L\dfrac{i_L}{dt} = V_{C_1} + ri_L \\[2mm] \dfrac{\varphi}{dt} = V_{C_2} \end{cases} \qquad (25)$$
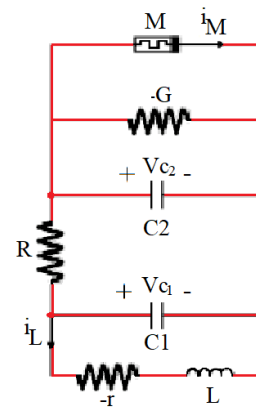

Fig. 14. Chaotic circuit described in [14] (Redrawing based on [14])

As seen in the previous scenario, the memristor used corresponds to a current-controlled memristor and can be used along with the reference and proposed memristors. Similar to the work in [14], the memristor has a quadratic function, and its structure is similar to the one proposed.

E. Chaotic Circuit 5

The option presented in [15], also referred to as the Chua circuit consists of five elements illustrated in Fig. 15. It defines a fourth-order system delineated by (26). Perhaps most notably, the elements employed in this configuration have not been utilized in other circuits.
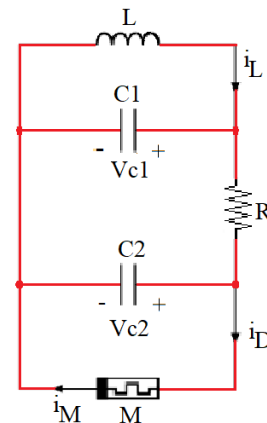

Fig. 15. Chaotic circuit of [15] (Redrawing based on [15])

$$\begin{cases} C_2\dfrac{dV_{C_2}}{dt} = \dfrac{V_{C_1}}{R} - \dfrac{V_{C_2}}{R} - i_D(V_{C_2}) \\[2mm] C_1\dfrac{dV_{C_1}}{dt} = \dfrac{V_{C_2}}{R} - \dfrac{V_{C_1}}{R} + i_L \\[2mm] L\dfrac{di_L}{dt} = -V_{C_1} \end{cases} \qquad (26)$$

In this circuit, a current-controlled memristor is used; hence, it is possible to use this circuit with the reference memristor and the proposed memristor. The alternative stated in [15] uses a memristor with a nonlinear quadratic function, offering a similar result.

V. Implementation of the Chaotic Circuit Based on Memristors

A. Selection of the chaotic circuit

Based on the analysis of the discussed chaotic circuits, one of them is selected to be implemented and used as an entropy

source. This requires discarding the alternatives that are not viable from an implementation standpoint regarding the values of the circuit elements and additional considerations. The analyzed strategies are comprised of passive elements such as resistors, inductors, and capacitors. The values used in the circuit variants can be detailed in Table III.

TABLE III
VALUES OF THE ELEMENTS USED IN DIFFERENT CHAOTIC CIRCUITS.

Element	Circuit				
	1	2	3	4	5
L	330 mH	4.45 H	22 mH	18 mH	18 mH
R	-	2 Ω	8.2 $k\Omega$	3.1 $k\Omega$ 0.5 $k\Omega$ 1 $k\Omega$	2 $k\Omega$
C	1 nF	0.5 F	6.8 nF 68 nF	10 nF 100 nF	68 nF 6.8 nF

Drawing from the details concerning the elements employed in chaotic circuits and the predefined criteria, the process of selecting the most appropriate circuit for implementation commences. Table IV aims to provide a summarized overview of the circuit characteristics.

TABLE IV
ASSESSMENT CRITERIA OF THE CHAOTIC CIRCUITS CONSIDERED.

Criterion	Circuit				
	1	2	3	4	5
Memristor type	Charge-controlled	Charge-controlled	Current-controlled	Current-controlled	Current-controlled
Number of elements	3	4	5	8	5
Type of inductor	Grounded	Floating	Floating	Grounded	Grounded

Based on the previous statements, Circuit 1 and Circuit 2 cannot be implemented given that they do not meet the characteristics expected for the type of memristor since they cannot be used with the reference and proposed counterparts. Despite using a memristor compatible with the reference and proposed memristor Circuit 4 is discarded since it requires more elements than other circuits. Circuits 3 and 5 meet the criteria of the memristor and use the same number of elements. However, the type of inductor required is different causing Circuit 3 to use more elements for the implementation. Circuit 5 is chosen as the chaotic circuit to be implemented, serving as an entropy source.

B. Implementation of the selected chaotic circuit

Various authors have successfully utilized and implemented Circuit 5 as a chaotic circuit, as evidenced in [16] – [18]. Therefore, these references serve as benchmarks for implementing the referenced and proposed memristors. Table V lists the values of the elements included in Circuit 5 as reported in each of the cited references.

TABLE V
VALUES USED FOR THE ELEMENTS OF CIRCUIT 5 IN DIFFERENT IMPLEMENTATIONS

Element	Reference		
	[16]	[17]	[18]
L	11.8 mH	20 mH	18 mH
C1	6.8 nF	10 nF	6.8 nF
C2	68 nF	1 nF	68 nF
R	2.1 $k\Omega$	-4 $k\Omega$	2 $k\Omega$

As none of the references mention the utilization of emulators as substitutes for the inductor, it is necessary to evaluate the feasibility of constructing the chaotic circuit

using an inductance emulator. The Chua circuit (Fig. 16) is purposefully employed, as it represents a classic chaotic circuit that incorporates Circuit 5 and has previously been effectively implemented [19].
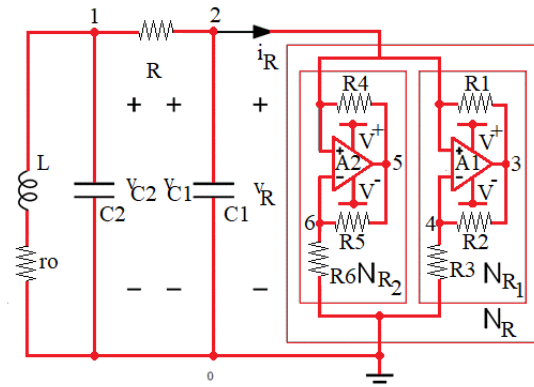

Fig. 16. Chua's circuit (Redrawing based on [19])

The circuit resulting from using the inductance emulator is presented in Fig. 17. The values used in the components of this implementation scenario are registered in Table VI.
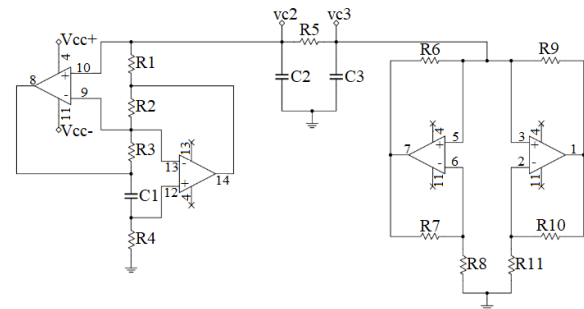

Fig. 17. Chua's circuit with inductance emulator.

TABLE VI
ELEMENT VALUES USED IN THE CHAOTIC CIRCUIT OF CHUA WITH INDUCTANCE EMULATOR.

Element	Value
R1, R2, R3	1.2 $k\Omega$
R4	1.5 $k\Omega$
C1	10 nF
C2	68 nF
C3	6.8 nF
R5	2 $k\Omega$
R6, R7	22 $k\Omega$
R8	3.3 $k\Omega$
R9, R10	220 Ω
R11	2.2 $k\Omega$

Based on the V_{C_2} vs V_{C_3} chart, the characteristic attractor of Chua's circuit can be obtained. This result serves as a criterion to determine whether it is possible to apply the inductance emulator. The result obtained is shown in Fig. 18.

Reference memristor.

The work of [11] is initially chosen as a reference for implementation development, given that it is the alternative closest to the reference memristor.

Nonetheless, since there is no consistency between the simulation, the implementation, or the results stated by the authors in [11], it is decided to use the work of [16] as a basis for development. In [16], although the values of the elements

that comprise the chaotic circuit are different from those used in [11], the most significant change can be seen in the memristor. Therefore, the changes proposed by [16] regarding the reference memristor are implemented. The resulting circuit can be seen in Fig. 19.
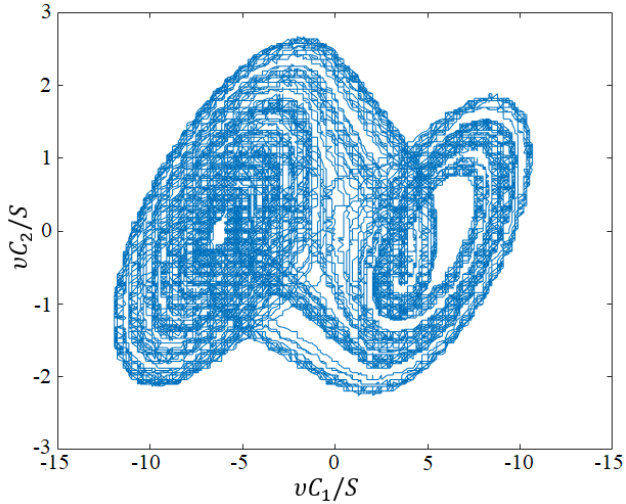

Fig. 18. Attractor from Chua's circuit with inductance emulator.

The changes mentioned influence the equations that describe the behavior of the chaotic circuit. This can be observed in (27).
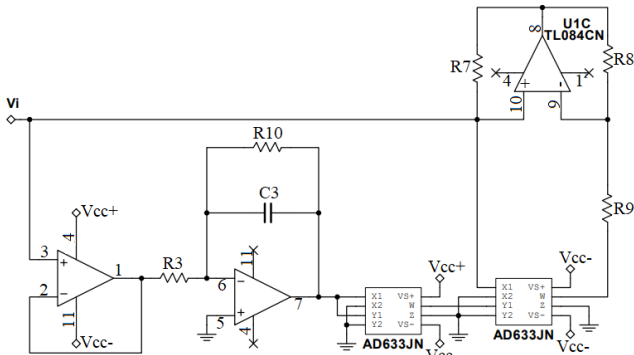

Fig. 19. Reference memristor with the changes discussed in [16]

$$\frac{dv_1}{dt} = \frac{v_2 - v_1}{RC_1} + \frac{(G_a - G_b v_0^2)v_1}{C_1}$$

$$\frac{dv_2}{dt} = \frac{v_1 - v_2}{RC_2} - \frac{i_3}{C_2}$$

$$\frac{di_3}{dt} = \frac{v_2}{L} \qquad (27)$$

$$\frac{dv_0}{dt} = -\frac{v_1}{R_1 C_0} - \frac{v_0}{R_2 C_0}$$

A simulation of this method is conducted to establish a reference point for comparison with the anticipated implementation outcome. The signals acquired in the simulation and the attractors generated for the chaotic system are depicted in Fig. 20 and Fig. 21, respectively.

Fig. 20 reveals that all signals remain within the operation range of the OP AMP. The effect of changing the memristor integrator can be seen. Fig. 21 shows that the attractors are also altered by this change. The attractor used is the one generated by V (C2) vs Vϕ, which is characteristic of this circuit. The implementation of the modified circuit encompasses the values registered in Table VII.
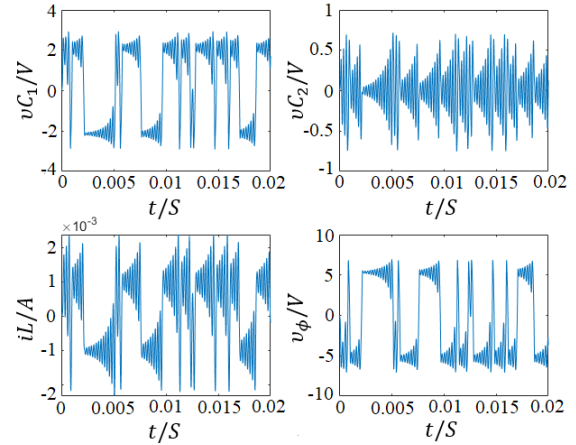

Fig. 20. Signals obtained from simulation of chaotic circuit 5 with the reference memristor.
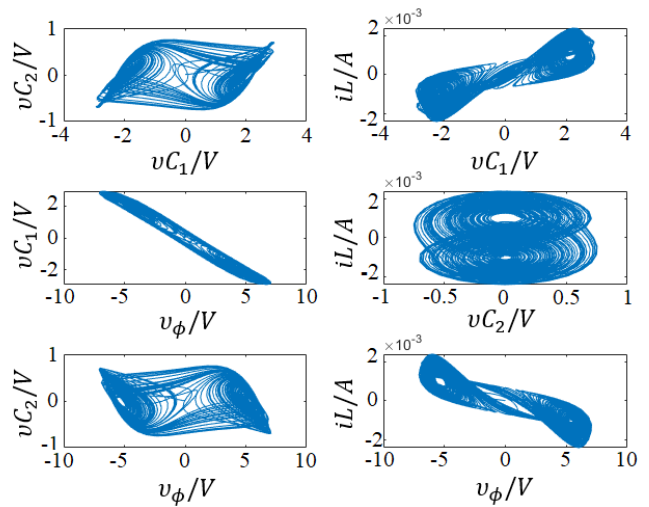

Fig. 21. Attractors obtained from simulation of chaotic circuit 5 with the reference memristor.

TABLE VII
VALUES OF THE ELEMENTS USED IN THE IMPLEMENTATION OF THE CHAOTIC CIRCUIT, BASED ON THE MODIFIED REFERENCE MEMRISTOR.

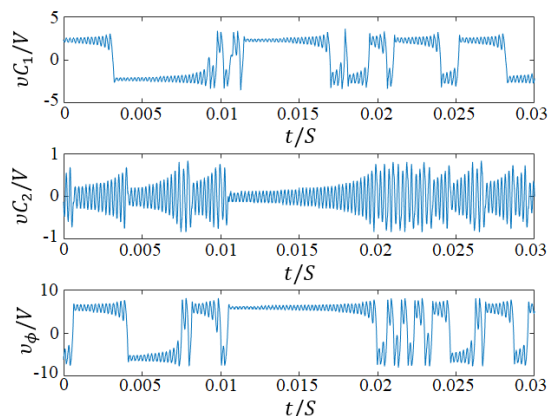| Element	Value
R1, R2, R3	1.2 $k\Omega$
R4	1.5 $k\Omega$
C1	10 nF
C2	68 nF
C3	6.8 nF
R5	2 $k\Omega$
R6	4 $k\Omega$
C4	1 nF
R7, R8	2 $k\Omega$
R9	1.5 $k\Omega$

Fig. 22. Signals obtained in the implementation of chaotic circuit 5 with the modified reference memristor.

The implementation yields signals and attractors exhibiting behaviors akin to those described in the simulation, affirming the success of the implementation in functioning as an entropy source for the random number generator [20],[21]. The circuit signals and attractors obtained are depicted in Fig. 22 and Fig. 23, respectively.
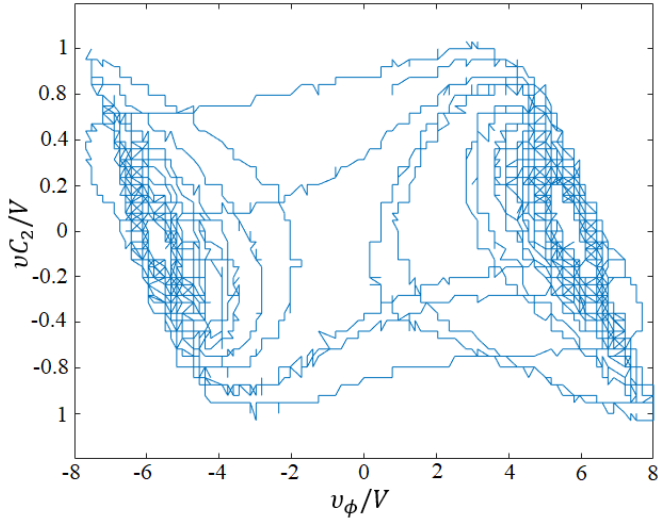

Fig. 23. Attractors obtained in the implementation of chaotic circuit 5 with the modified reference memristor.

Proposed memristor

The changes applied to the reference memristor over the proposed memristor result in the circuit shown in Fig. 24. Equation (28) presents the equation system that describes the behavior of the chaotic circuit.
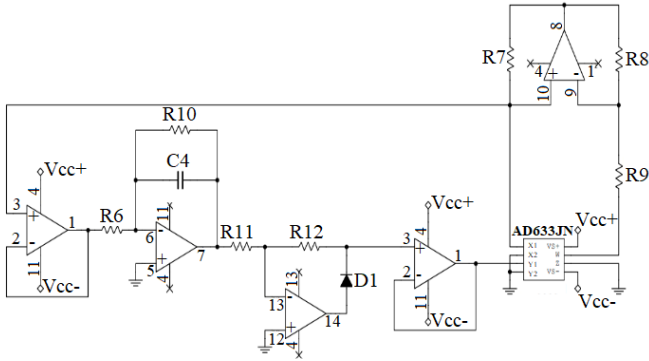

Fig. 24. Modified memristor circuit.

$$\frac{dv_1}{dt} = \frac{v_2 - v_1}{RC_1} + \frac{(G_a - G_b v_0^2)v_1}{C_1}$$

$$\frac{dv_2}{dt} = \frac{v_1 - v_2}{RC_2} - \frac{i_3}{C_2}$$

$$\frac{di_3}{dt} = \frac{v_2}{L} \qquad (28)$$

$$\frac{dv_0}{dt} = -\frac{v_1}{R_1 C_0} - \frac{v_0}{R_2 C_0}$$

Fig. 25 presents the results of the system signals, and Fig. 26 presents the attractors obtained during the simulation.

Fig. 25 shows that none of the signals are outside of the OP-AMP operation range, so the implementation is performed using the values of the components listed in Table VIII.
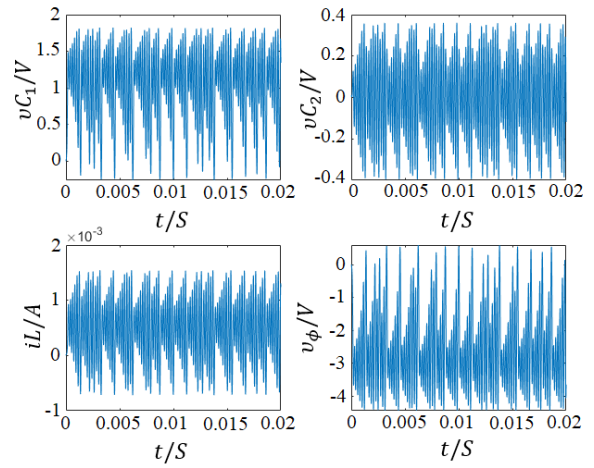

Fig. 25. Signals obtained of the chaotic circuit 5 with the modified memristor.
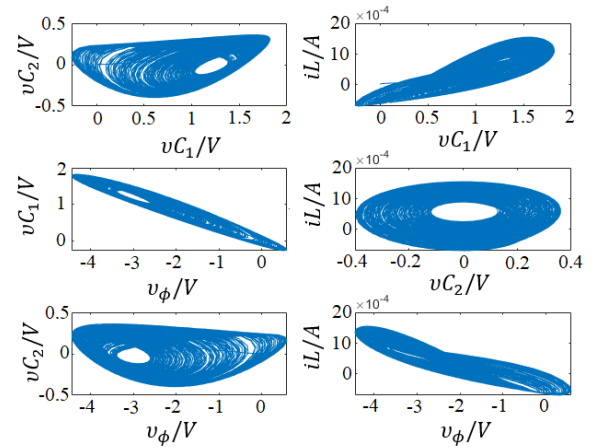
Fig. 26. Attractors obtained of the chaotic circuit 5 with the modified reference memristor.

TABLE VIII
VALUES OF THE ELEMENTS USED IN THE IMPLEMENTATION OF CHAOTIC CIRCUIT 5 BASED ON THE MODIFIED REFERENCE MEMRISTOR.

Element	Value	Element	Value
R1, R2, R3	1.2 $k\Omega$	R6	8.2 $k\Omega$
R4	1.5 $k\Omega$	C4	47 nF
C1	10 nF	R7, R8	2 $k\Omega$
C2	68 nF	R9	1.5 $k\Omega$
C3	6.8 nF	R10, R11, R12	10 $k\Omega$
R5	2 $k\Omega$		

The implemented circuit generates chaotic signals and attractors similar to those obtained in Fig. 27.

Fig. 27. Signals obtained of the chaotic circuit 5 with the modified reference memristor.

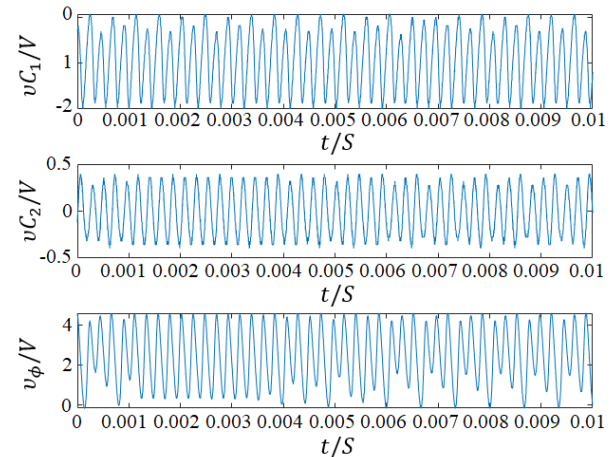The Fig. 28 showcasing the system signals and the attractors obtained in the implementation process.
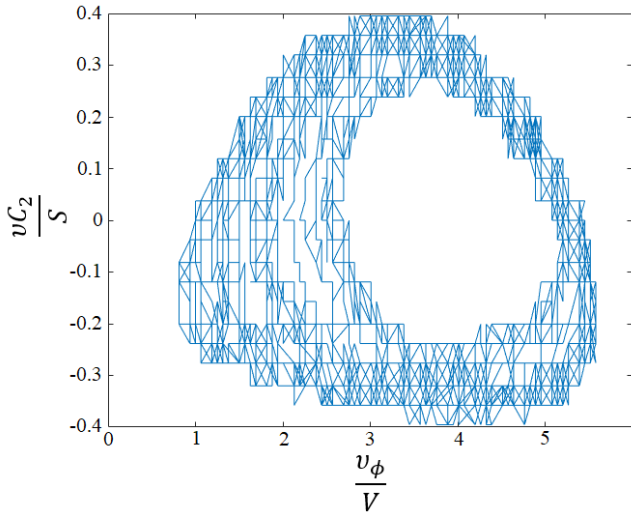

Fig. 28. Attractors obtained of the chaotic circuit 5 with the modified reference memristor.

VI. APPLICATION OF THE CHAOTIC CIRCUIT

As we mentioned, at the beginning of this paper, the utilization of a memristor, in a chaotic circuit, paves the way for random number generation. In this sense, we continue this research and use the chaotic circuit as an entropy source. Specifically, we implemented the system explained in [4], taking the chaotic signals v_(C_1), v_(C_2), and v_ϕ, as inputs of the random number generator. The detailed analysis and results of this second part of the research have already bee10ubmitted for publication and are out of the scope of this paper. However, we want to summarize here, some of the main contributions that we have made.

After testing the six combinations of the order of the chaotic signals, and ten combinations of the RNG parameters that we proposed in [4], such as the number of bits in the ADCs and DACs, and the number of least significant bits taken from the ADC, we have verified that is possible to use this system to generate random numbers with long-range dependence. By simply changing the order of the concatenated signals of both reference and proposed memristor, it was possible to select some significant values of the Hurst parameter, which belong to the interval (0.5,1). This achievement allowed us to further analyze the fractal behavior of the random numbers generated, and we can state that monofractal and multifractal behavior were both detected.

These characteristics suggest that the generator can be used as a reference framework for possible applications in the financial, geophysical, texture modeling, climate prediction, characterization of social behavior, and growth of organisms, among others. We are particularly interested in using it, in an an application that enables establishing the time between R peaks of a synthetic ECG signal generator.

VII. LRD DEPENDENCY TESTING

The detection of the LRD is carried out from the Hurst parameter (H), this can be determined from the variance-time diagram (VT) and the log-scale diagram (LD) [4]. The use of the variance-time diagram is selected as a tool to estimate the value of the Hurst parameter for each discrete sequence obtained.

An example of the resulting VT diagram can be seen in Fig. 29 and Fig. 30 when using discrete sequences with LRD and without LRD respectively.
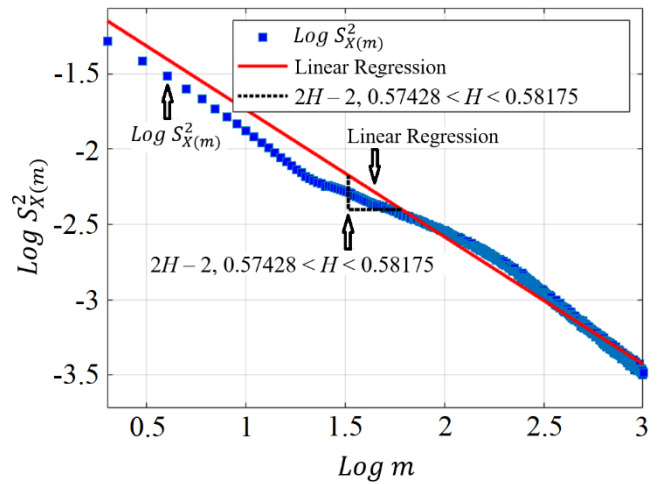
Fig. 29. Variance-Time diagram for a discrete sequence with LRD (H = 0.5780)
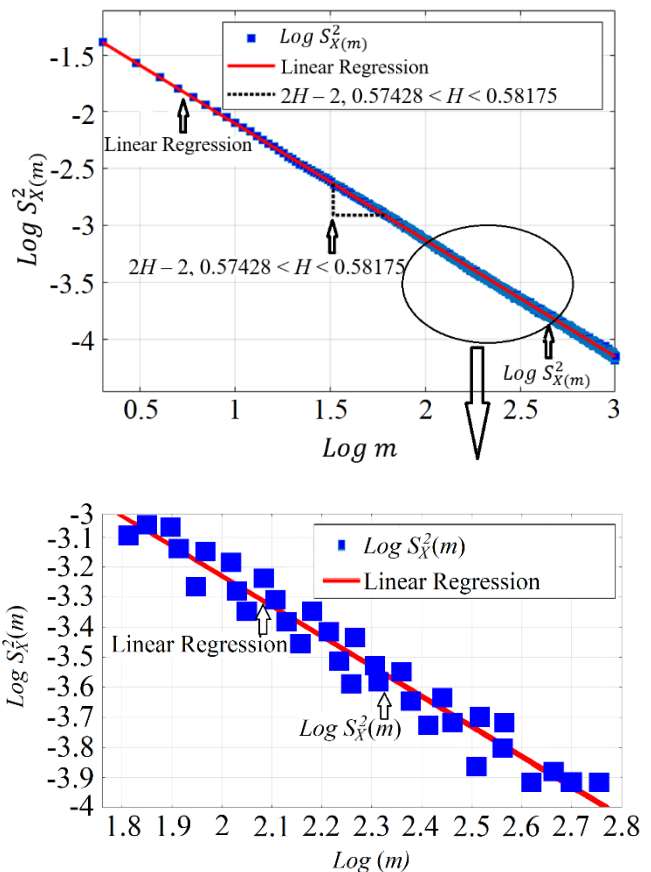
Fig. 30. VT diagram for a discrete sequence without LRD (H = 0.4879)

Table IX records the combinations and the Hurst parameter obtained by the authors of [4], these results are compared with those obtained in the implementation when using the reference memristor (Table X) and proposed memristor (Table XI).

From Table X it can be seen that the Hurst parameter is affected by the order used to generate the sequence. It can also be observed that excluding the models that use 0 bits in the DAC, the rest of the combinations have sequences of numbers that exhibit LRD behavior.

TABLE IX.
COMBINATIONS AND PARAMETERS OBTAINED BY THE AUTHORS OF [4]

	Combination									
	1	2	3	4	5	6	7	8	9	10
b	8	16	12	16	10	8	0	8	0	0
n	8	8	8	10	10	12	12	10	10	10
m	4	7	7	9	8	10	9	10	10	9
H	0.5218	0.5756	0.6510	0.6757	0.6913	0.7196	0.7738	0.7958	0.8171	0.821

TABLE X.
HURST PARAMETERS OBTAINED WITH A CHAOTIC CIRCUIT WITH REFERENCE MEMRISTOR FOR DIFFERENT COMBINATIONS AND SEQUENCES

	Combination									
	1	2	3	4	5	6	7	8	9	10
1	0.704	0.565	0.566	0.599	0.602	0.831	0.434	0.4	0.704	0.389
2	0.602	0.548	0.603	0.709	0.751	0.749	0.434	0.479	0.602	0.469
3	0.602	0.548	0.603	0.709	0.751	0.749	0.380	0.418	0.602	0.469
4	0.701	0.565	0.574	0.708	0.748	0.612	0.434	0.472	0.701	0.467
5	0.602	0.548	0.698	0.714	0.712	0.660	0.384	0.418	0.602	0.47
6	0.70	0.565	0.5695	0.5989	0.5990	0.5152	0.4344	0.4012	0.7013	0.389

TABLE XI.
HURST PARAMETER OBTAINED WITH A CHAOTIC CIRCUIT WITH MEMRISTOR PROPOSED FOR DIFFERENT COMBINATIONS AND SEQUENCES

	Combination									
	1	2	3	4	5	6	7	8	9	10
1	0.4879	0.5198	0.5228	0.4973	0.527	0.4991	0.44	0.4889	0.4762	0.4625
2	0.578	0.7697	0.8019	0.7765	0.5053	0.4871	0.44	0.49	0.4791	0.4636
3	0.4945	0.6252	0.6483	0.6478	0.5549	0.5311	0.33	0.4896	0.4765	0.4633
4	0.833	0.7593	0.7972	0.7719	0.4884	0.4715	0.46	0.4913	0.4787	0.4645
5	0.4985	0.6101	0.6532	0.6568	0.501	0.4784	0.40	0.4927	0.4816	0.4658
6	0.494	0.5145	0.5195	0.5009	0.4917	0.4925	0.48	0.4931	0.481	0.4658

From Table XI it can be seen that the order in which the sequence is generated continues to affect the Hurst parameter. As with the reference memristor, combinations that use 0 bits as DAC resolution still exhibit deterministic behavior.

VIII. CONCLUSIONS

The goal of this project is to determine the problems, conditions, and limitations that arise during the hardware implementation of the proposed memristor model and chaotic circuit. The fact that the memristor is not available commercially as a passive element reveals an impending need for emulator circuits. This leads to the selection of a reference memristor along with its structure according to the criteria defined for the components, thus restricting the solutions to only those comprised of elements available in the national market.

Regarding the memristor proposal, it can be concluded that the limitations are not only related to hardware. As described by Chua, the conditions that characterize a memristor element should be considered when choosing a non-linear function for the emulator. The condition of only being equal at the origin does not allow the use of functions that can have values of zero throughout specific intervals, making it impossible to use half-wave rectifiers or similar devices.

In the development phase, the entropy source poses the most significant constraints - primarily due to potential hindrances in hardware applications stemming from the availability of electronic components. Prioritizing a cost-effective and straightforward implementation, we found it imperative to limit the components used to those readily available commercially. Consequently, we must dismiss the work presented in [4] without conducting a preliminary analysis of elements such as inductances and memristors.

The implementation of a chaotic circuit with different memristor emulators supports the study of the effect of non-linear functions in a chaotic system. These functions generate completely different signals and attractors. Signals and attractors are also affected by the topology used in the integrator of the emulator, as seen with both types of integrators used in this work.

IX. FUTURE WORK

The generation of random numbers based on hardware requires an entropy source as a fundamental element of the implementation process, which is used along with algorithms and mathematical models to deliver a sequence of numbers. Therefore, the creation of an entropy source using the memristor chaotic circuit would have wide applicability for the generation of random numbers.

ACKNOWLEDGMENTS

REFERENCES

[1] Alharbi A. G. and Chowdhury M. H., "Memristor Emulator Circuits," First Edition. Ed. Cham, Switzerland: Springer, pp. 1-18, 2021, doi: 10.1007/978-3-030-51882-0.

[2] Chua L. O., "Memristor—The Missing Circuit Element," IEEE Trans. Circuit Theory, vol. 18, no. 5, pp. 507–519, 1971, doi: 10.1109/TCT.1971.1083337.

[3] Abunahla H., Mohammad B., Jaoude M. A., and Al-Qutayri M., "Novel Hafnium Oxide Memristor Device," IEEE Int. Symp. Circuits Syst., Baltimore, MD, USA, 2017, pp. 1-4, 2017, doi: 10.1109/ISCAS.2017.8050791.

[4] Téllez M., Mejia J., Lopez H., and Hernandez C., "Random Number Generator with Long-Range Dependence and Multifractal Behavior Based on Memristor", Electronics, vol. 9, pp. 1-24, 2020, doi: 10.3390/electronics9101607.

[5] Alharbi A. G., Fouda M. E., and Chowdhury M. H., "A novel memristor emulator based only on an exponential amplifier and CCII+," Proc. IEEE Int. Conf. Electron. Circuits, Syst., vol. 2016, pp. 376–379, 2016, doi: 10.1109/ICECS.2015.7440327.

[6] Alharbi A. G., Khalifa Z. J., Fouda M. E., and Chowdhury M. H., "A new simple emulator circuit for current controlled memristor," Proc. IEEE Int. Conf. Electron. Circuits, Syst., vol. 2016, pp. 288–291, 2016, doi: 10.1109/ICECS.2015.7440305.

[7] Yu D. S., Sun T. T., Zheng C. Y., Iu H. H. C., and Fernando T., "A Simpler Memristor Emulator Based on Varactor Diode," Chinese Phys. Lett., vol. 35, no. 5, pp. 1–4, 2018, doi: 10.1088/0256-307X/35/5/058401.

[8] Wang X. Y., Fitch A. L., Iu H. H. C., Sreeram V., and Qi W. G., "Implementation of an analog model of a memristor based on a light-dependent resistor," Chinese Phys. B., vol. 21, no. 10, pp. 1139–1142, 2012, doi: 10.1088/1674-1056/21/10/108501.

[9] Nguyen V. H., Sohn K. Y., and Song H., "On-printed circuit board emulator with controllability of pinched hysteresis loop for nanoscale TiO 2 thin-film memristor device," J. Comput. Electron., vol. 15, no. 3, pp. 993–1002, 2016, doi: 10.1007/s10825-016-0862-x.

[10] Porro S., Accornero E., Pirri C. F., and Ricciardi C., "Memristive devices based on graphene oxide," Carbon N. Y., vol. 85, no. January, pp. 383–396, 2015, doi: 10.1016/j.carbon.2015.01.011.

[11] Muthuswamy B. and Chua L. O., "Simplest Chaotic Circuit," Int. J. Bifurc. Chaos., vol. 20, no. 05, pp. 1567–1580, 2010, doi:

10.1142/S0218127410027076.

[12] Hu Q., Yu Y., Men L., Lei F., and Zhang H., "Memristor-based chaotic circuit design on image En/decryption," 2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC), Wuhan, China, pp. 56-60, 2016, doi: 10.1109/YAC.2016.7804865.

[13] Li Y., Zhao L., Chi W., Lu S., and Huang X., "Implementation of a New Memristor Based Chaotic System," 2012 Fifth Int. Work. Chaos-fractals Theor. Appl., pp. 92–96, 2012, doi: 10.1109/IWCFTA.2012.75.

[14] Li Y., Huang X., and Guo M., "The generation, analysis, and circuit implementation of a new memristor-based chaotic system," Math. Probl. Eng., 2013, doi: 10.1155/2013/398306.

[15] Cruz J. M. and Chua L. O., "A CMOS IC Nonlinear Resistor for Chua's Circuit," IEEE Trans. Circuits Syst. I Fundam. Theory Appl., vol. 39, no. 12, pp. 985–995, 1992, doi: 10.1109/81.207719.

[16] Xu Q., Lin Y., Bao B., and Chen M., "Multiple attractors in a non-ideal active voltage-controlled memristor-based Chua's circuit," Chaos, Solitons and Fractals., vol. 83, pp. 186–200, 2016, doi: 10.1016/j.chaos.2015.12.007.

[17] Bao B., Luo J., Bao H., Xu Q., Hu Y., and Chen M., "Initial conditions-related dynamical behaviors in PI-type memristor emulator-based canonical Chua's circuit," Circuit World., vol. 44, no. 4, pp. 178–186, 2018, doi: 10.1108/CW-07-2018-0053.

[18] Muthuswamy B., "Implementing memristor-based chaotic circuits," Int. J. Bifurc. Chaos., vol. 20, no. 5, pp. 1335–1350, 2010, doi: 10.1142/S0218127410026514.

[19] Landgraf T., "estimação de parâmetros no sistema caótico de chua utilizando medidas amostradas no tempo : experi ências práticas e," XIX Congresso Brasileiro de Automática, pp. 1085-1092, 2012.

[20] Karim M. Aljebory, Yashar M. Jwmah, and Thabit S. Mohammed, "Classification of EMG Signals: Using DWT Features and ANN Classifier," IAENG International Journal of Computer Science, vol. 51, no.1, pp23-31, 2024.

[21] Ming-Xuan Yan, Jia-Cheng Zhou, Jia-Min Huang, Yu-Jie Jiang, Xiao-Jun Zhang, and Zhi Tao, "ECG Signal Classification Method for Double-threshold Segmented Sparse Representation," IAENG International Journal of Computer Science, vol. 50, no.4, pp1238-1249, 2023.

Juan Polo was born in Bogotá, Colombia. He is a graduate from electronic engineering from Universidad Distrital Francisco José de Caldas, Colombia. His research interests include bioengineering and intelligent systems.

Hans Lopez was born in Barranquilla, Colombia. He is master's graduate in information and communications sciences from Universidad Distrital Francisco José de Caldas, Colombia. He is currently a Titular Professor of electronic engineering at Universidad Distrital Francisco José de Caldas. His research interests include statistical signal processing and biological signal processing.

Cesar Hernandez was born in Villavicencio, Colombia. He has a PhD in engineering from Universidad Nacional de Colombia. He is currently a Titular Professor of electrical engineering programs with Universidad Distrital Francisco José de Caldas. His research interests include mathematical optimization, cognitive radio networks, and intelligent systems.