

## Research Article

# Degradation Analysis of Chaotic Systems and their Digital Implementation in Embedded Systems

**Rodrigo Méndez-Ramírez** <sup>1</sup>, **Adrian Arellano-Delgado**,<sup>2</sup> **Miguel Murillo-Escobar** <sup>1</sup>,  
and **César Cruz-Hernández** <sup>1</sup>

<sup>1</sup>Scientific Research and Advanced Studies Center of Ensenada, B. C. (CICESE), 22080 Ensenada, Mexico

<sup>2</sup>Engineering, Architecture and Design Faculty, Autonomous Baja California University (CONACYT-UABC), Ensenada, BC, Mexico

Correspondence should be addressed to César Cruz-Hernández; [ccruz@cicese.mx](mailto:ccruz@cicese.mx)

Received 8 September 2019; Accepted 14 October 2019; Published 4 December 2019

Guest Editor: Serdar Çiçek

Copyright © 2019 Rodrigo Méndez-Ramírez et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Digital implementation of chaotic systems (CSs) has attracted increasing attention from researchers due to several applications in engineering, e.g., in areas as cryptography and autonomous mobile robots, where the properties of chaotic systems are strongly related. The CSs in the continuous version (CV) need to be discretized where chaotic degradation must be analyzed to guarantee preservation of chaos. In this paper, we present a degradation analysis of five three-dimensional CSs and the necessary conditions to implement the discretized versions (DVs) of Lorenz, Rössler, Chen, Liu and Chen, and Méndez-Arellano-Cruz-Martínez (MACM) CSs. Analytical and numerical analyses of chaos degradation are conducted by using the time series method; the maximum discrete step size and the Lyapunov Exponents (LEs) are computed by using the Euler, Heun, and fourth-order Runge-Kutta (RK4) numerical algorithms (NAs). We conducted comparative studies of performance based on time complexity of the five proposed CSs in their DVs by using four embedded systems (ESs) based on three families of Microchip microcontrollers 8-bit PIC16F, 16-bit dsPIC33FJ, and 32-bit PIC32MZ (of low-cost electronic implementation) and a Field Programmable Gate Array (FPGA). Based on the results, the intervals at control parameters to guarantee chaos are proposed, which improves the performance characteristics of the five proposed CSs in their DVs based on digital applications.

## 1. Introduction

In recent years, scientific community has become interested in chaotic systems (CSs) due to their potential application in several areas of engineering, where the properties of chaos are desired, such as high sensitivity to initial conditions, high entropy, topology complexity, ergodicity, among others [1–6]. Electronic implementations based on chaos have been developed for digital applications, e.g., as pseudorandom sequence generator [7], synchronization of optical networks [8], image encryption [9], chaotic trajectories for autonomous mobile robot [10], chaotic radar [11], among others [12–17]. Lorenz is the first 3D CS reported in the literature [18]. Since then, many CSs in 3D and 4D with different features and properties have been reported [19–29]. Moreover, literature

reports chaotic maps (discrete by nature) desirable properties at such applications, e.g., the logistic map in 1D [5], Hénon map in 2D [30], among others.

The 3D CSs can be implemented electronically in their continuous or discretized versions; their continuous versions (CVs) can be implemented using operational amplifiers [27–29, 31]. On the other hand, distinct numerical algorithms (NAs) are used to implement the discretized versions (DVs) of the 3D CSs [32–34]. Software tools, such as Matlab or Labview, allow to simulate and reproduce the CSs in their DVs using NAs as Euler, Heun, and RK4 [35], where a small step size is considered to compare their DV versus their CV [32–34].

The literature reports digital implementations of CSs in their DVs for different applications by using embedded

systems (ESs) such as microcontrollers where the main cores are 8-bit PIC18F microcontroller [36]; 16-bit dsPIC microcontroller [37]; 32-bit microcontrollers such as PIC32 [29, 38], ARM Cortex-M3 [39], DSP [40], and Altera and Xilinx FPGAs [7, 32–34, 41, 42]; system on chip (SoC) that contains fast processors as NanoPC-T3 Plus [43]; and Raspberry Pi 3 [44], among others.

Recently, the literature reports degradation studies of 3D CSs in their DVs by using the NA of Euler; a robustness diagram for control parameters guarantee chaos is presented, and its digital implementation is conducted in a microcontroller PIC32 [29].

The methods equivalent used to conduct arithmetic and logical calculations inside of a microprocessor—or its equivalent microcontroller as main core of an ES—are based on numerical standards. Microchip Technology Inc. is the manufacturer of PIC, dsPIC, and PIC32 microcontrollers; their numerical results are based by the IEEE-754 Compliant Floating Point Routines [45]. On the other hand, Altera-Intel is an FPGA manufacturer, and their numerical results of simulations are represented in IEEE-754 (2008) [46, 47]. The NAs are simulated by using Matlab, and its results are similar in comparison with the compilers used by Microchip microcontrollers and software for design used by Altera-Intel FPGA because both are based on IEEE-754 [35, 46, 47].

The FPGA has powerful simulation tools to reproduce chaotic dynamics of CSs in their DVs by using digital signal processing (DSP) modules as a complementary tool for Matlab/Simulink software, e.g., Altera Simulink/DSP Builder and Xilinx System Generator BlockSet [48, 49].

In this paper, we present a degradation analysis of the five 3D CSs in their DVs to determine the performance of implementation in four versions of an ES, the time complexity, and the intervals of control parameters to guarantee chaos are obtained. The results of this paper can be of great interest for digital applications of chaos in engineering. To our knowledge, the literature does not report comparative studies of digital degradation of five 3D CSs in their DVs by using the NAs of Euler, Heun, and RK4, where its performance is conducted in microcontrollers of 8, 16, and 32 bits, and FPGA.

The paper is organized as follows: In Section 2, the normalized version of five three-dimensional CSs are presented, numerical analyses calculating the Lyapunov exponents are conducted to verify the chaotic behavior using the Euler, Heun, and RK4 NAs where a Root-Mean-Square Error (RMSE) analysis is conducted to compare their continuous and DVs. Section 3 presents the digital implementation on ES with 8-bit PIC16F, 16-bit dsPIC33, 32-bit PIC32MZ microcontrollers, and the Altera FPGA Cyclone IV GX, where the performance and the robustness digital diagram to guarantee the chaos is proposed. Finally, conclusions of this work are reported in Section 4.

## 2. Degradation Analysis

In this section, we describe the normalized equations of the 3D Lorenz, Rössler, Chen, Liu and Chen, and MACM CSs to obtain their DVs by using the NAs of E, H, and RK4. The

time series method is used to obtain the degradation limits by calculating the LEs of the five 3D CSs in their continuous and DVs [50, 51]. We analyzed the accuracy of the trajectories of state variable  $x$  of the 3D Lorenz, Rössler, Chen, Liu and Chen, and MACM CSs by calculating the RMSE. All the numerical results and methods described in this section are conducted by using Matlab [35].

**2.1. Normalized 3D CSs.** This subsection briefly describes the normalized version of the Lorenz, Rössler, Chen, Liu and Chen, and MACM CSs, and the difference between them is given by the complexity of their dynamics, the representation of their nonlinear functions, and parameters.

First, we consider the Lorenz system, which is a well-known example of a CS. Lorenz's three-variable model provides a practical test case with qualitatively realistic properties [18]; it is represented by the nonlinear state equations described as

$$\begin{aligned}\dot{x} &= \sigma(y - x), \\ \dot{y} &= rx - xz - y, \\ \dot{z} &= xy - bz,\end{aligned}\tag{1}$$

where  $x$ ,  $y$ , and  $z$  are the state variables and the standard parameter values for Lorenz's chaotic attractor are  $\sigma = 10$ ,  $r = 28$ , and  $b = 8/3$ . We also consider the Rössler system introduced by Rössler in 1976 [19], which is described by

$$\begin{aligned}\dot{x} &= -y - z, \\ \dot{y} &= x + ay, \\ \dot{z} &= b + z(x - c),\end{aligned}\tag{2}$$

where  $x$ ,  $y$ , and  $z$  are the state variables, the Rössler system presents chaotic behavior for the following parameter values:  $a = 0.2$ ,  $b = 0.2$ , and  $c = 5.7$ . Similarly, the Chen system is introduced as a dual system of the Lorenz system in 1999 [23] and is described by

$$\begin{aligned}\dot{x} &= a(y - x), \\ \dot{y} &= (c - a)x - xz + cy, \\ \dot{z} &= xy - bz,\end{aligned}\tag{3}$$

where  $x$ ,  $y$ , and  $z$  are the state variables and the Chen system presents chaotic behavior for the following parameter values:  $a = 35$ ,  $b = 3$ , and  $c = 28$ . Moreover, the Liu and Chen system was introduced in 2002, and its description is given by [24]

$$\begin{aligned}\dot{x} &= ax + d_1 yz, \\ \dot{y} &= cy + d_2 xz, \\ \dot{z} &= bz + d_3 xy,\end{aligned}\tag{4}$$

where  $x$ ,  $y$ , and  $z$  are the state variables, this nonlinear system presents a chaotic behavior when the following condition  $ab + ac + bc \neq 0$  is met. It can create a complex 2-scrolls attractor from the following parameter values:  $d_1 = -1$ ,  $d_2 = d_3 = 1$ ,  $a = 5$ ,  $c = -10$ , and  $b = -3.4$ . Recently, the MACM CS was proposed in 2017 [29], which is described by

$$\begin{aligned}\dot{x} &= -ax - byz, \\ \dot{y} &= -x + cy, \\ \dot{z} &= d - y^2 - z,\end{aligned}\quad (5)$$

where  $x$ ,  $y$ , and  $z$  are the state variables and the MACM system presents chaotic behavior for the following parameter values:  $a = b = 2$ ,  $c = 0.5$ , and  $d = 4$ .

In this study, we use the same initial conditions (ICs)  $x_0 = y_0 = z_0 = 1$  for the CSs (1)–(5). Table 1 shows the summary of control parameters, critical parameters, nonlinearities, and ICs of the five 3D CSs (1)–(5) [29].

The literature reports the validation of chaos calculating the limits of the LEs by using the time series method by Wolf and Briggs [50, 51], and the LEs and fractal dimension, commonly known as Kaplan–Yorke dimension  $D_{KY}$ , of the five 3D CSs (1)–(5) in their CVs are computed by using the proposed time series method. Table 2 shows the LEs and the fractal dimension results of CSs (1)–(5) in their CVs.

**2.2. Numerical Algorithms.** NAs of Euler, Heun, and RK4 are used to obtain the DV of the CSs (1)–(5), the step size as referred to as  $\tau$ , and  $n$  is the iteration number that represents the time in DV. The nonlinear functions  $f$ ,  $g$ , and  $h$  of the five 3D CSs of Lorenz, Rössler, Chen, Liu and Chen, and MACM in their DVs describe the states  $x$ ,  $y$ , and  $z$ , respectively.

Euler algorithm presents just one step, and it is easy to implement because it requires less arithmetic operations [35]. The Euler NA is described by

$$\begin{aligned}x_{(n+1)} &= x_{(n)} + \tau f(x_{(n)}, y_{(n)}, z_{(n)}), \\ y_{(n+1)} &= y_{(n)} + \tau g(x_{(n)}, y_{(n)}, z_{(n)}), \\ z_{(n+1)} &= z_{(n)} + \tau h(x_{(n)}, y_{(n)}, z_{(n)}).\end{aligned}\quad (6)$$

The Heun is the second NA implemented [35]; this method is known as trapezoidal in two steps where the first step predicts and the second step corrects. The NA of Heun is described as follows:

$$\begin{aligned}x_{(n+1)}^* &= x_{(n)} + \tau f(x_{(n)}, y_{(n)}, z_{(n)}), \\ y_{(n+1)}^* &= y_{(n)} + \tau g(x_{(n)}, y_{(n)}, z_{(n)}), \\ z_{(n+1)}^* &= z_{(n)} + \tau h(x_{(n)}, y_{(n)}, z_{(n)}),\end{aligned}\quad (7)$$

where

$$\begin{aligned}x_{(n+1)} &= x_{(n)} + \frac{\tau}{2} \left( f(x_{(n)}, y_{(n)}, z_{(n)}) + x_{(n+1)}^* \right), \\ y_{(n+1)} &= y_{(n)} + \frac{\tau}{2} \left( g(x_{(n)}, y_{(n)}, z_{(n)}) + y_{(n+1)}^* \right), \\ z_{(n+1)} &= z_{(n)} + \frac{\tau}{2} \left( h(x_{(n)}, y_{(n)}, z_{(n)}) + z_{(n+1)}^* \right).\end{aligned}\quad (8)$$

The third NA is the RK4; this algorithm is one of the most widely used methods for solving differential equations [35]. The NA of RK4 is given by

$$\begin{aligned}x_{(n+1)} &= x_{(n)} + \frac{\tau}{6} (k_1 + 2k_2 + 2k_3 + k_4), \\ y_{(n+1)} &= y_{(n)} + \frac{\tau}{6} (l_1 + 2l_2 + 2l_3 + l_4), \\ z_{(n+1)} &= z_{(n)} + \frac{\tau}{6} (m_1 + 2m_2 + 2m_3 + m_4),\end{aligned}\quad (9)$$

where  $k_1, k_2, k_3$ , and  $k_4$  are referred to as coefficients of the first equation, similarly, the parameters  $l_1, l_2, l_3$ , and  $l_4$  are referred to as coefficients of the second equation, and the parameters  $m_1, m_2, m_3$ , and  $m_4$  are referred to as coefficients of the third equation. The coefficients described in system (9) are given by

$$\begin{aligned}k_1 &= f(x_{(n)}, y_{(n)}, z_{(n)}), \\ l_1 &= g(x_{(n)}, y_{(n)}, z_{(n)}), \\ m_1 &= h(x_{(n)}, y_{(n)}, z_{(n)}),\end{aligned}\quad (10)$$

$$\begin{aligned}k_2 &= f\left(x_{(n)} + \frac{\tau}{2}k_1, y_{(n)} + \frac{\tau}{2}l_1, z_{(n)} + \frac{\tau}{2}m_1\right), \\ l_2 &= g\left(x_{(n)} + \frac{\tau}{2}k_1, y_{(n)} + \frac{\tau}{2}l_1, z_{(n)} + \frac{\tau}{2}m_1\right), \\ m_2 &= h\left(x_{(n)} + \frac{\tau}{2}k_1, y_{(n)} + \frac{\tau}{2}l_1, z_{(n)} + \frac{\tau}{2}m_1\right),\end{aligned}\quad (11)$$

$$\begin{aligned}k_3 &= f\left(x_{(n)} + \frac{\tau}{2}k_2, y_{(n)} + \frac{\tau}{2}l_2, z_{(n)} + \frac{\tau}{2}m_2\right), \\ l_3 &= g\left(x_{(n)} + \frac{\tau}{2}k_2, y_{(n)} + \frac{\tau}{2}l_2, z_{(n)} + \frac{\tau}{2}m_2\right), \\ m_3 &= h\left(x_{(n)} + \frac{\tau}{2}k_2, y_{(n)} + \frac{\tau}{2}l_2, z_{(n)} + \frac{\tau}{2}m_2\right),\end{aligned}\quad (12)$$

$$\begin{aligned}k_4 &= f(x_{(n)} + \tau k_3, y_{(n)} + \tau l_3, z_{(n)} + \tau m_3), \\ l_4 &= g(x_{(n)} + \tau k_3, y_{(n)} + \tau l_3, z_{(n)} + \tau m_3), \\ m_4 &= h(x_{(n)} + \tau k_3, y_{(n)} + \tau l_3, z_{(n)} + \tau m_3).\end{aligned}\quad (13)$$

Finally, the coefficients described in (10)–(13) are placed in (9); they as whole represent the NA of RK4 (9)–(13).

**2.3. Degradation Analysis of the 3D CSs in Their DVs.** In this subsection, the maximum step size is referred to as  $\tau_{\max}$ , and it is computed by using the time series method considering one positive LE as the condition to guarantee chaos in the DV of the five 3D CSs [50, 51]. LEs to obtain the chaotic degradation of the five CSs in their DVs are computed by using the NAs of Euler (6), only the  $\tau_{\max}$  were reported in [29]; in this study, the LEs and  $D_{KY}$  are added, and their results are presented in Table 3. In addition, we computed LEs,  $\tau_{\max}$ , and  $D_{KY}$  of the five CSs in their DVs by using the NAs of Heun (7) and (8) and RK4 (9)–(13), and their results are presented in Tables 4 and 5, respectively.

The results obtained in Tables 3–5 show that MACM CS presents the higher  $\tau_{\max}$ , and Chen and Liu and Chen CSs

TABLE 1: Parameters, characteristics, and ICs of the five 3D CSs (1)–(5).

CS	Control parameter	Critical parameter	Nonlinearities	Initial condition $(x_0, y_0, z_0)$
Lorenz (1)	$\sigma = 10,$ $r = 8/3,$ $b = 28$	$\sigma$	2	(1, 1, 1)
Rössler (2)	$a = 0.2,$ $b = 0.2,$ $c = 5.7$	$c$	1	(1, 1, 1)
Chen (3)	$a = 35,$ $b = 3,$ $c = 28$	$a$	2	(1, 1, 1)
Liu and Chen (4)	$a = 5,$ $c = -10,$ $b = -3.4,$ $d_1 = -1,$ $d_2 = d_3 = 1.$	$c$	3	(1, 1, 1)
MACM (5)	$a = b = 2,$ $c = 0.5,$ $d = 4.$	$c, d$	2	(1, 1, 1)

TABLE 2: LEs of the CSs (1)–(5) in their CVs.

CS	LE <sub>1</sub>	LE <sub>2</sub>	LE <sub>3</sub>	$D_{KY}$
Lorenz (1)	0.91	0	-14.47	2.062
Rössler (2)	0.07	0	-5.39	2.012
Chen (3)	2.02	0	-12.02	2.168
Liu and Chen (4)	0.87	0	-9.27	2.093
MACM (5)	0.24	0	-2.74	2.087

TABLE 3: Analysis of chaos degradation for DV of 3D CSs by using the NA of Euler (6).

CS	$\tau_{\max}$	LE <sub>1</sub>	LE <sub>2</sub>	LE <sub>3</sub>	$D_{KY}$
Lorenz	$\leq 0.024$	0.039	-62.8 $\mu$	-0.389	2.101
Rössler	$\leq 0.005$	49 $\mu$	34.9 $\mu$	-5.53	2.0142
Chen	$\leq 0.002$	3.8 m	0.11 m	-23.1 m	2.169
Liu and Chen	$\leq 0.002$	197 $\mu$	-53.21 $\mu$	-18.72 m	2.1029
MACM	$\leq 0.085$	0.05	-64.58 $\mu$	-0.243	2.099

TABLE 4: Analysis of chaos degradation for DV of 3D CSs by using the NA of Heun (7) and (8).

CS	$\tau_{\max}$	LE <sub>1</sub>	LE <sub>2</sub>	LE <sub>3</sub>	$D_{KY}$
Lorenz	$\leq 0.068$	86.6 m	-2.016 m	-0.6231	2.1358
Rössler	$\leq 0.191$	0.0273	196 $\mu$	-0.24122	2.1141
Chen	$\leq 0.017$	38.7 m	23.67 $\mu$	-0.19558	2.1984
Liu and Chen	$\leq 0.017$	17.41 m	-17.88 $\mu$	-0.15542	2.112
MACM	$\leq 0.228$	95.29 m	315.64 $\mu$	-0.476	2.2006

TABLE 5: Analysis of chaos degradation for DV of 3D CSs by using the NA of RK4 (9)–(13).

CS	$\tau_{\max}$	LE <sub>1</sub>	LE <sub>2</sub>	LE <sub>3</sub>	$D_{KY}$
Lorenz	$\leq 0.1076$	64.17 m	-4.03 m	-1.1924	2.0504
Rössler	$\leq 0.251$	35.53 m	339.62 $\mu$	-0.402	2.0891
Chen	$\leq 0.057$	97.48 m	-2.949 m	-0.6509	2.1452
Liu and Chen	$\leq 0.057$	55.52 m	-1.83 m	-0.389	2.0974
MACM	$\leq 0.547$	0.23081	-48.55 m	-1.0098	2.1805

present the smallest  $\tau_{\max}$  in comparison with Lorenz. In the case of Rössler system, we recommend to use  $\tau_{\max} = 0.005$  because, for higher values, the dynamical chaotic behavior is lost and dynamic behavior of limit cycle for step sizes within the intervals  $0.006 \leq \tau \leq 0.091$  is shown; for higher values of  $\tau = 0.091$ , the LEs cannot be computed using the time series method with the NA (6). Rössler system supports an interval small of  $\tau_{\max}$  because the chaotic dynamics diverges. Nevertheless, Rössler system presents a high interval in comparison Lorenz system by using the NAs of Heun (7) and (8) and RK4 (9)–(13), and their results are described in Tables 4 and 5.

Figure 1 illustrates the comparison of  $\tau_{\max}$  obtained with the results of the NAs described in Tables 3–5. NA of RK4 presents higher  $\tau_{\max}$  in comparison of the NAs of Euler, and Heun. The DV of MACM system presents a higher  $\tau_{\max}$  than the DV of the CSs of Lorenz, Rössler, Chen, and Liu and Chen systems.

**2.4. Performance of Chaotic Behavior.** We use RMSE to compare the performance and accuracy of the trajectory of state  $x$  of the CSs (1)–(5) in their CVs respect to their DVs by using the NAs of Euler (6), Heun (7) and (8), and RK4 (9)–(13). The RMSE is defined as follows:

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_{e(n)} - x_{(n)})^2}, \quad (14)$$

where the state variable  $x_{e(n)}$  is referred to as the estimator value of the CV of CSs (1)–(5), the state variable  $x_{(n)}$  is referred to as the predicted estimated value of DV of CSs (1)–(5), and  $n$  is referred to as the total number samples. The Ordinary Differential Equation (ODE) function number 45 (ODE45) of MATLAB is considered to reproduce the CV of the CSs (1)–(5), although strictly it is also a discretized representation, this algorithm is based on an explicit Runge–Kutta

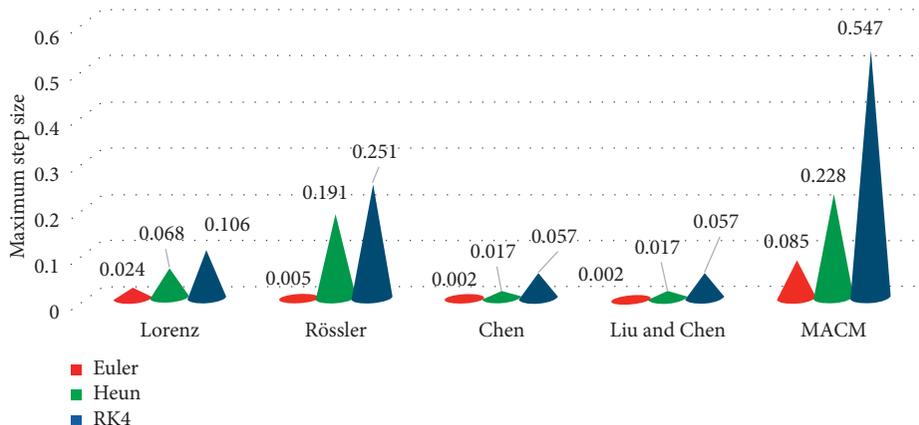


FIGURE 1:  $\tau_{\max}$  summary of chaotic degradation of the five 3D CSs in their DVs.

4-5 formula, it is a single-step solver and needs only the solution at the immediately preceding point time [52, 53].

Numerical tests of the 3D Lorenz, Rössler, Chen, Liu and Chen, and MACM CSs in their DVs are conducted for each Euler (6), Heun (7) and (8), and RK4 (9)–(13) NA, respectively; we considered  $n = 30000$  samples, a small step size  $\tau = 0.001$ , and same parameters and initial conditions are shown in Table 1.

The error calculation is compared with respect to state variable  $x_{(n)}$  of the five 3D CSs in their DVs, and their trajectory errors are referred to as follow:  $e_1$  represents the difference between ODE45 and Euler algorithm (6),  $e_2$  represents the difference between ODE45 and Heun algorithm (7) and (8), and  $e_3$  represents the difference between ODE45 and RK4 algorithm (9)–(13).

Figure 2 shows the comparison of the evolution of the state variable  $x_{(n)}$  of the Lorenz, Rössler, Chen, Liu and Chen, and MACM CSs in their continuous version using ODE45 and DVs using the NAs of (6)–(13). Figure 3 shows the trajectory errors  $e_1$ – $e_3$  of the trajectories of set of CSs shown in Figure 2. Figure 2(b) shows that the trajectories of the Rössler system (2) have no changes; they are practically the same trajectories for the DVs with respect to the continuous version (2), and Figure 3(b) shows that the Rössler system has minimal errors for  $e_1$ – $e_3$ . Considering the numerical results of the trajectories of state  $x$  of the set of five 3D CSs shown in Figures 2 and 3, the RMSE comparison is obtained by using (14), and its result is shown in Figure 4. Chen and Liu and Chen systems show high RMSE, and to a lesser extent in the Lorenz system, followed by this value, the MACM system exhibits a low RMSE value, but the Rössler system exhibits the lowest RMSE (see Figure 4).

Rössler system guarantees a better conservation of the chaos considering the proposed  $n$  samples and  $\tau$  step size by using the NAs of Euler (6), Heun (7)–(8), and RK4 (9)–(13).

### 3. Digital Implementation

In this section, we present the necessary conditions to implement the NAs of the five 3D CSs in their DVs considering the described studies in Section 2.

The digital implementation is carried out in an ES which main core is represented in four different hardware versions: 8-bit PIC16F, 16-bit dsPIC33, and 32-bit PIC32MZ microcontrollers, and one Cyclone IV GX FPGA. Table 6 shows the hardware description of the four versions of the ES.

Microcontrollers U1–U3, FPGA U4, and DACs U5–U7 are configured according to the performance recommended by their manufacturers—the SPI protocol was configured in the master mode from specification of U1–U4 by using 12 bits of resolution. The DAC U5, U6, and U7 represent the state variables  $x_{(t)}$ ,  $y_{(t)}$ , and  $z_{(t)}$ , respectively, and its software configuration is given from U1–U4. Figure 5 shows the hardware description for the four versions of ES. Version 1 (V1) represents the hardware implementation by using U1, Version 2 (V2) represents the hardware implementation by using U2, Version 3 (V3) represents the hardware implementation by using U3, and Version 4 (V4) represents the hardware implementation by using U4.

Initially, the NAs Euler (6), Heun (7) and (8), and RK4 (9)–(13) of the 3D Lorenz, Rössler, Chen, Liu and Chen, and MACM CSs in their DVs are simulated by using Matlab, the numerical standard of Matlab is based on IEEE-754 standard for floating point representation [35]. The compilers used to program and implement the NAs Euler (6), Heun (7) and (8), and RK4 (9)–(13) of the 3D Lorenz, Rössler, Chen, Liu and Chen, and MACM CSs in their DVs inside of U1–U4 are based on C language, the microcontrollers U1–U3 have similar standard IEEE-754 which is referred to as Compliant Floating Point Routines AN575 [45]. According to the Altera-Intel manufacturer, the FPGA U4 is based on the IEEE-754 standard (2008) [46, 47]. The compilers of the manufacturer Mikroelektronika are used to program U1–U3 [54]. The FPGA U4 is programmed by using the set tools of Quartus II to design the hardware, and Eclipse compiler is used to design the software. Therefore, the numerical results conducted in the proposed four versions V1–V4 of the ES are similar because Matlab and the C compilers to program U1–U3 and U4 have the IEEE-754 standard [35, 44–46].

In order to conduct the simulations of the NA, we used the Proteus Virtual System Modeling (VSM) Software, in special, the VSM for Microchip version that contains the device libraries of some families of 8-bit PIC and 16-bit

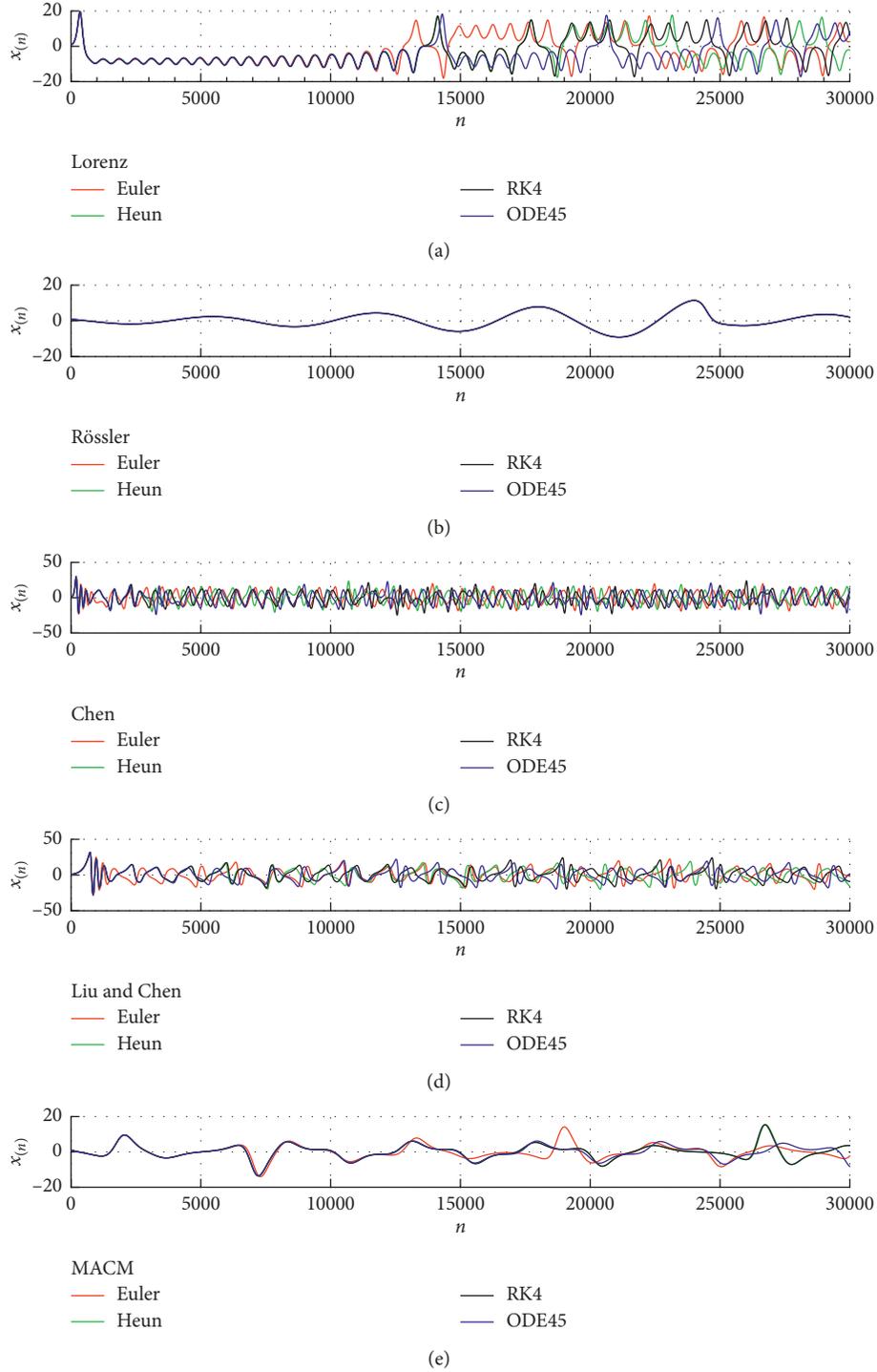


FIGURE 2: Evolution of state variable  $x(n)$  of the CSs: (a) Lorenz (2), (b) Rössler (2), (c) Chen (3), (d) Liu and Chen (4), and (e) MACM (5) in their CVs with respect to their DVs by using the NAs of Euler (6), Heun (7)–(8), and RK4 (9)–(13).

dsPIC33 microcontrollers, and the schematic capture tool was used to simulate the complete V1 and V2 ESs [55]. The numerical results of V1–V4 proposed and their equivalences between simulation and implementation are carried out by using the methods described in [29, 37, 38].

The total quantity of iterations  $Q_T$  is referred to as the maximum number of  $n$  iterations generated in 1 second,

and it is represented in time units (tu), and the CSs (2)–(6) are represented in three dimensions, i.e., we are considering  $N=3$  dimensions, and the  $Q_T$  representation is given by

$$Q_T = \tau \frac{1}{T_{Td}} = \tau f_{Td} = \tau \frac{1}{t_c + t_{Tg}} = \tau \frac{1}{t_c + \sum_{j=1}^N t_{Tdac(j)}}, \quad (15)$$

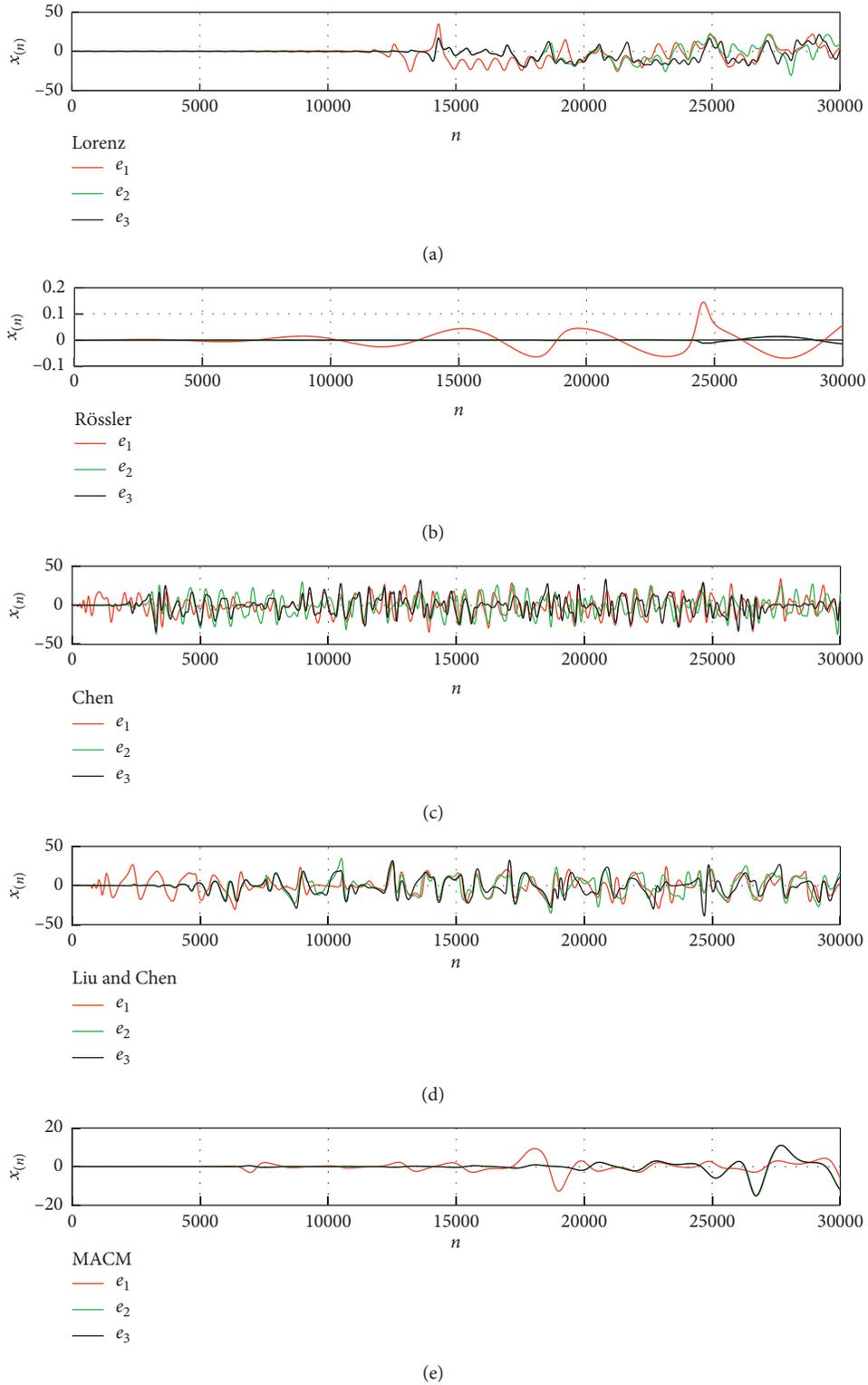


FIGURE 3: Errors  $e_1$ ,  $e_2$ , and  $e_3$  shown in Figure 2 for the CSs: (a) Lorenz (1), (b) Rössler (2), (c) Chen (3), (d) Liu and Chen (4), and (e) MACM (5).

where the time period  $T_{Td}$  is considered as the total-decoding-time that the algorithm needs to reproduce an iteration  $n$ , and  $f_{Td}$  represents the maximum number of

iterations  $n$  that the ES generates in 1 second (ips); the frequency  $f_{Td}$  is the reciprocal of  $T_{Td}$ . The time complexity  $t_c$  is the time that NA need to reproduce one iteration  $n$  by

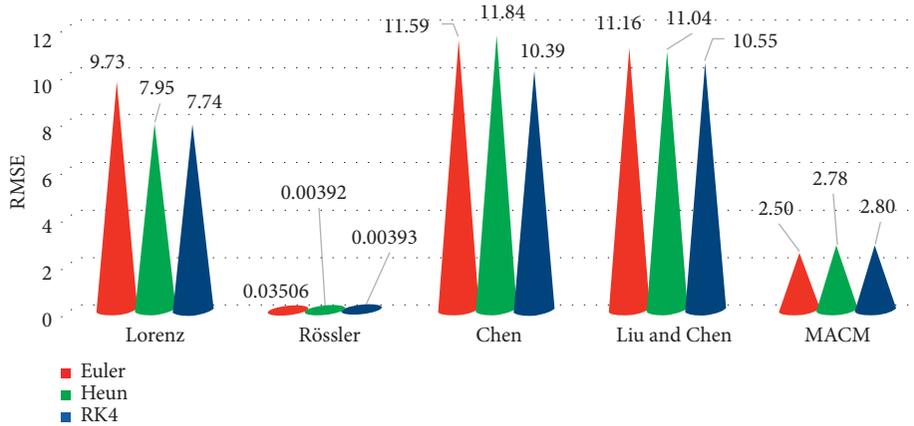


FIGURE 4: RMSE comparison of trajectory performance of state  $x$  of five 3D CSs (1)–(5) in their CVs with respect to their DVs using the NAs (6)–(13).

TABLE 6: Hardware description of the ES for V1–V4.

Peripheral number	SPI mode, hardware description
U1	Master, 8-bit PIC16F874A microcontroller
U2	Master, 16-bit dsPIC33FJ32MC204 microcontroller
U3	Master, 32-bit PIC32MZ2048ECM064 microcontroller
U4	Master, EP4CGX150DF31C7 Cyclone IV GX FPGA
U5	Slave 1, DAC MCP4921 shows $x_{(t)}$
U6	Slave 2, DAC MCP4921 shows $y_{(t)}$
U7	Slave 3, DAC MCP4921 shows $z_{(t)}$

using the main core of U1, U2, U3, or U4; the total-graphic time  $t_{Tg}$  is the time that the three DACs U5–U7 need to represent the state variables  $x_{(t)}$ ,  $y_{(t)}$ , and  $z_{(t)}$ , and the time required for each DAC (U5–U7) is referred by  $t_{Tdac(1-3)}$ , respectively.

The measurement of the iterations  $f_{Tg}$  and the time  $T_{Td}$  are obtained experimentally when the program of NA used is executed in the 4 versions of the ES. Only  $t_c$  depends on the size of the NA used. The size of  $t_{Tg}$  depends of the configuration of SPI protocol used in U1–U4.

The equation (15) is a reference to determine the performance of the proposed ES in their four versions V1–V4. In this study, we need to obtain a smaller period of time to determine a greater number of iterations  $n$  and a higher step size to guarantee the chaos proposed in Tables 3–5; their values depend on the NA of the CSs proposed and the version of the ES that will be implemented. Given these considerations, we will obtain a higher  $Q_T$ . This means that the state variables of the 3D Lorenz, Rössler, Chen, Liu and Chen, and MACM CSs in their DVs by using the NAs (6)–(13) have a better representation of  $f_{Td}$ , which is very attractive for applications based in chaos such as: master key definition, encryption, and secure communications [9, 37, 43, 56].

**3.1. Embedded System with 8-Bit PIC Microcontroller.** First, we implemented the ES in the V1 by using the microcontroller PIC16F874A U1. We used Proteus to conduct the electronic simulations of the NAs of the CSs

proposed, and the schematic diagram is shown in Figure 6.

Euler algorithm (6) is used to obtain the DV of the Lorenz system and its corresponding system is given by

$$\begin{aligned}
 x_{(n+1)} &= x_{(n)} + \tau(\sigma(y_{(n)} - x_{(n)})), \\
 y_{(n+1)} &= y_{(n)} + \tau(rx_{(n)} - x_{(n)}z_{(n)} - y_{(n)}), \\
 z_{(n+1)} &= z_{(n)} + \tau(x_{(n)}y_{(n)} - bz_{(n)}).
 \end{aligned} \tag{16}$$

Figure 7 shows the results of the Lorenz system simulation in its DV (16), we considered the higher step size  $\tau_{max} = 0.024$  according to the results shown in Table 3. The voltage supplied for the ES in V1 is  $V_{dd} = +5\text{ V}$  and  $V_{ss} = 0\text{ V}$ , and an external crystal of 16 MHz is used according to the datasheet of U1. The test in the version 1 on the ES is conducted, and we obtained  $T_{Td} = 2046\ \mu\text{s}$  and  $f_{Td} = 488.7\text{ ips}$ ; this means that we can obtain  $Q_T = 11.7\text{ tu}$  in 1 second by using V1 in the Proteus simulator.

To conduct the hardware implementation on V1 of ES, we used the same electrical parameters used in the simulation of CS (16). Figure 8 illustrates the implementation results of the algorithm (16) by using the 8-bit PIC microcontroller U1. Figure 8(b) shows the time evolution of the state variables  $x_{(n)}$  and  $z_{(n)}$  of Lorenz system (16) for 1 second. We experimentally obtained  $t_c = 1989\ \mu\text{s}$ ,  $t_{Tg} = 57\ \mu\text{s}$ ,  $T_{Td} = 2046\ \mu\text{s}$ , and  $f_{Td} = 488.7\text{ ips}$  considering the same  $\tau_{max} = 0.024$ ; this means that the simulation conducted by using Proteus and the hardware implementation in the version 1 is consistent, because both have the same units of  $Q_T = 11.7$  time in 1 second.

The NAs of Euler (6), Heun (7) and (8), and RK4 (9)–(13) of the 3D Lorenz, Rössler, Chen, Liu and Chen, and MACM CSs in their DVs are executed in the V1 of the ES; the NA of RK4 (9)–(13) cannot be executed using U1 because the size of its program flash memory is small and it only supports 4K bytes. We obtained the performance of the ES in the V1, and the time complexity and the iterations per second are detailed in Table 7.

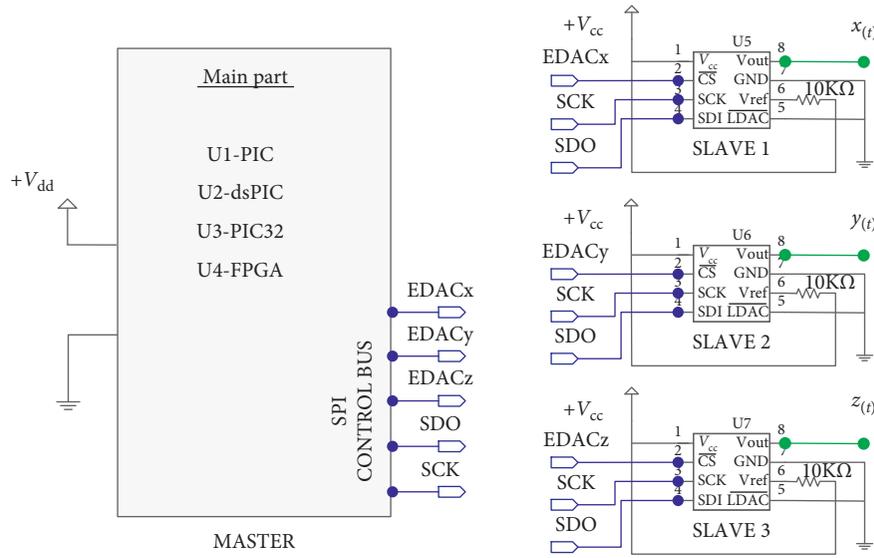


FIGURE 5: Schematic circuit of ES for V1-V4.

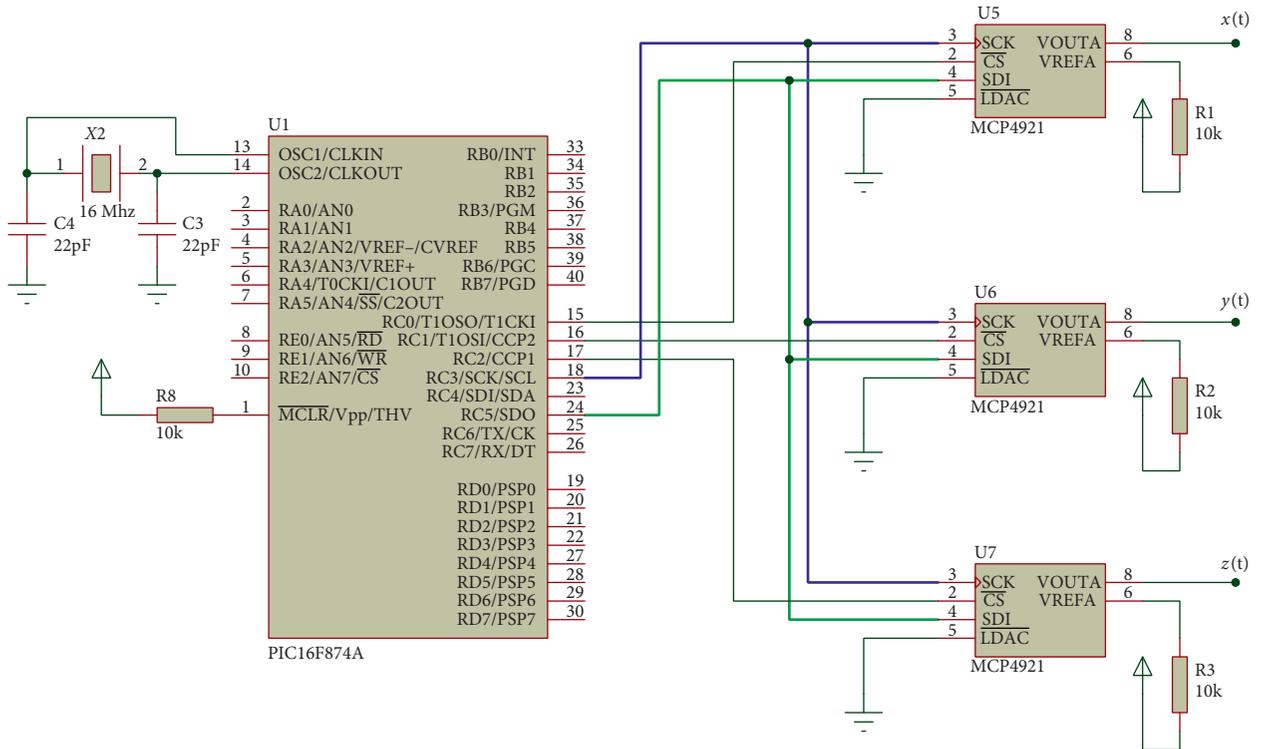


FIGURE 6: Schematic of the simulation of ES in V1 with the Proteus software.

3.2. *Embedded System with dsPIC Microcontroller.* Second implementation is conducted; the NAs of Euler (6), Heun (7) and (8), and RK4 (9)–(13) of the 3D Lorenz, Rössler, Chen, Liu and Chen, and MACM CSs in their DVs are implemented by using the ES in its version 2. Figure 9 illustrates the simulation and implementation of dsPIC33 U2 in Proteus by using the V2 of the ES. The voltage supplied for the ES in V2 is  $V_{dd} = +3.3$  V and  $V_{ss} = 0$  V, and an external crystal of 10 MHz is used according to the datasheet of U2. We conducted an example to implement the Liu and Chen

CS in its DV by using the Euler algorithm (6), and its algorithm is described as follows:

$$\begin{aligned} x_{(n+1)} &= x_{(n)} + \tau(ax_{4(n)} + d_1y_{(n)}z_{(n)}), \\ y_{(n+1)} &= y_{(n)} + \tau(cy_{4(n)} + d_2x_{(n)}z_{(n)}), \\ z_{(n+1)} &= z_{(n)} + \tau(bz_{(n)} + d_3x_{(n)}y_{(n)}). \end{aligned} \quad (17)$$

Figure 10 shows the simulation results conducted in Proteus of the DV of Liu and Chen CS (17) by using dsPIC33

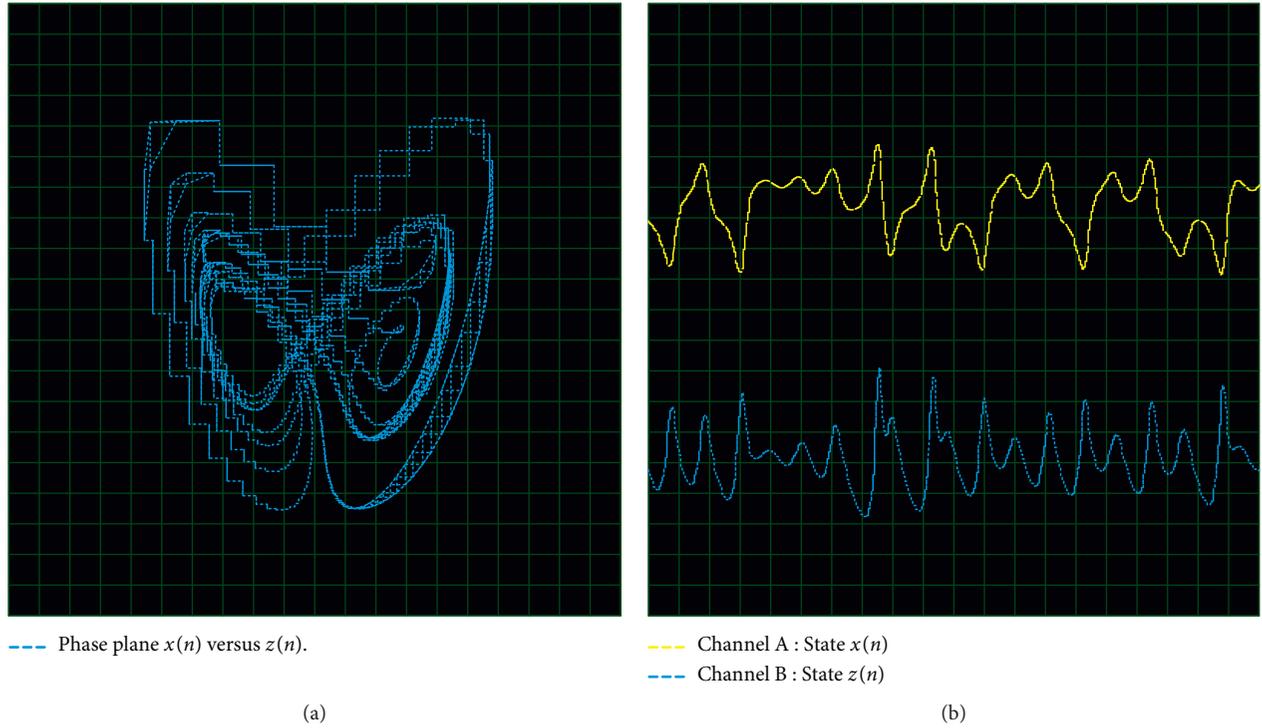


FIGURE 7: Simulation of ES in the V1 of Lorenz system in its DV by using  $\tau_{\max} = 0.024$ . (a) Phase plane  $x(n)$  versus  $z(n)$  and (b) evolution of state variables  $x(n)$  and  $z(n)$ .

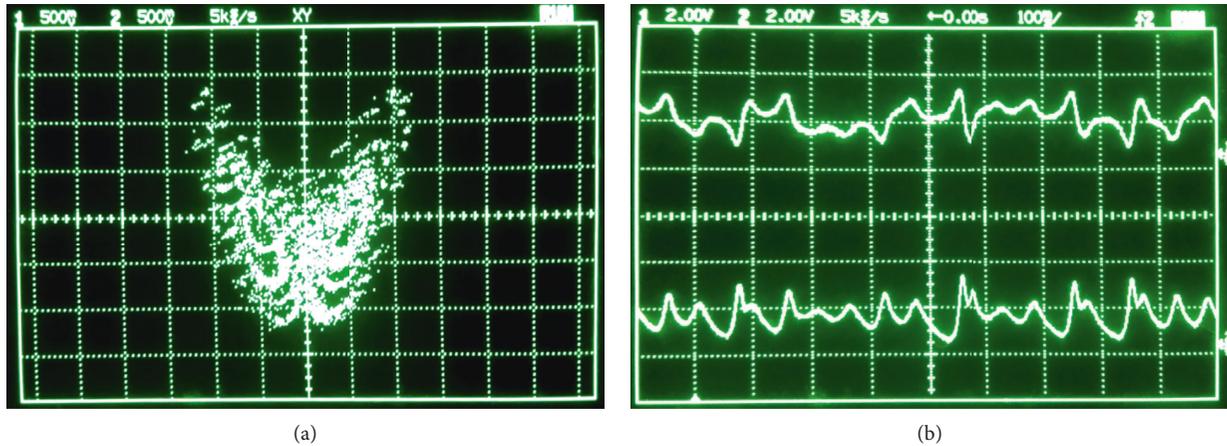


FIGURE 8: Implementation of Lorenz CS (15) in its DV by using the V1 of ES: (a) Phase plane  $x(n)$  versus  $z(n)$  and (b) evolution of state variables  $x(n)$  and  $z(n)$ .

TABLE 7: Performance of ES in the V1 of the 3D Lorenz, Rössler, Chen, Liu and Chen, and MACM CSs in their DVs by using Euler (6), and Heun (7) and (8) NAs.

CS	Euler		Heun	
	$T_{Td}$ ( $\mu s$ )	$f_{Td}$ (ips)	$T_{Td}$ ( $\mu s$ )	$f_{Td}$ (ips)
Lorenz	2046	488.7	3450	289.8
Rössler	1720	581.3	2760	362.3
Chen	2046	488.7	3450	289.8
Liu and Chen	2480	403.2	4420	226.2
MACM	2046	488.7	3450	289.8

U2. According to the results shown in Table 3, the higher step size  $\tau_{\max} = 0.002$  was chosen, we obtained  $t_c = 237 \mu s$ ,  $t_{Tg} = 8 \mu s$ ,  $T_{Td} = 245 \mu s$ , and  $f_{Td} = 4082$  ips; this means that we can reproduce  $Q_T = 8.164$  tu in 1 second. Figure 11(a) illustrates the digital oscilloscope of Proteus simulator, it only allows a reduced quantity of samples to show the plane phase  $x(n)$  versus  $z(n)$ .

Figure 11 shows the implementation results of the algorithm (17) of the ES in its V2 considering the same  $\tau_{\max} = 0.002$ , we obtained  $t_c = 86 \mu s$ ,  $t_{Tg} = 3 \mu s$ ,  $T_{Td} = 89 \mu s$ ,

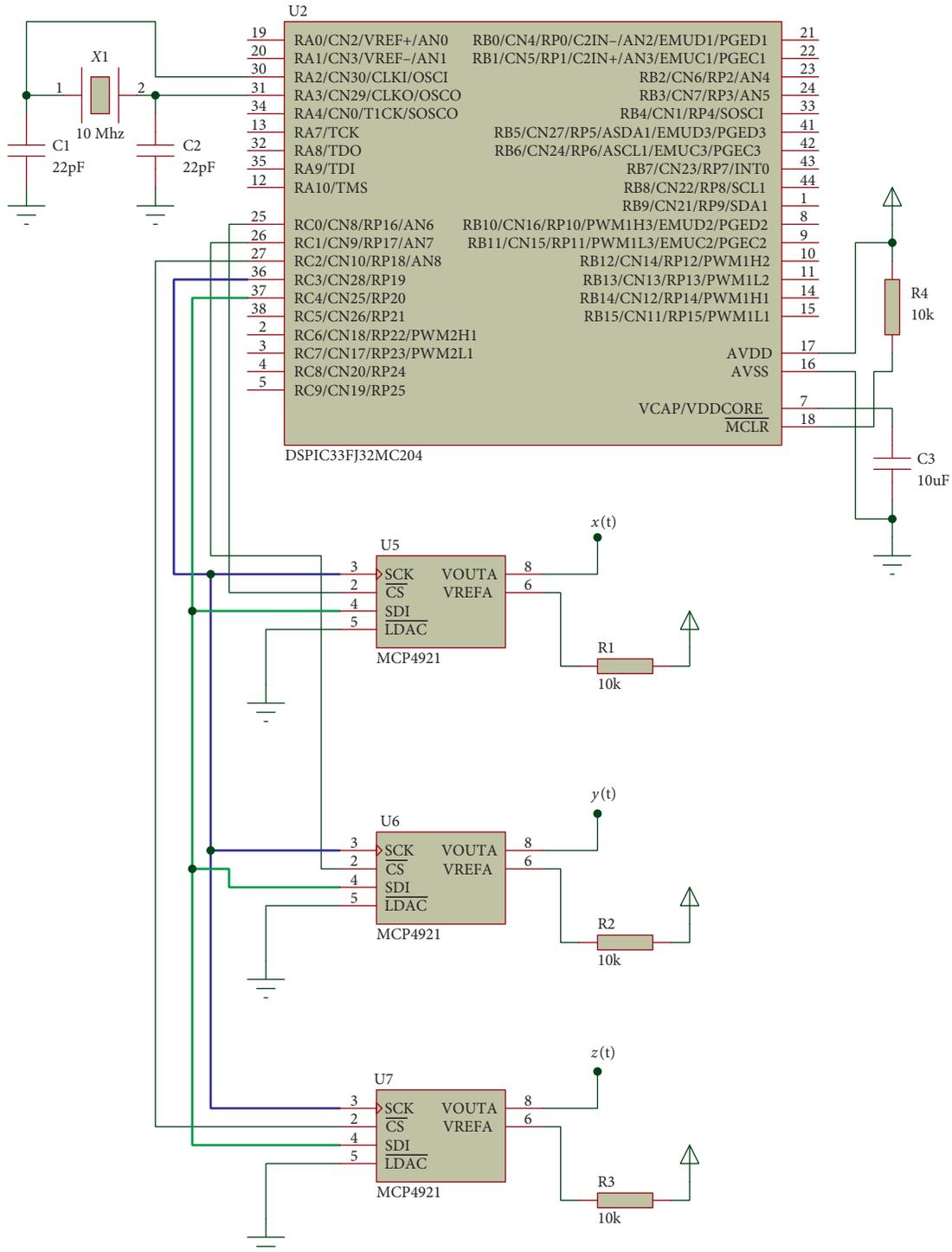


FIGURE 9: Schematic of the simulation of ES in V2 by using Proteus.

and  $f_{Td} = 11236$  ips; this means that the hardware implementation in the V2 shows better performance because we obtained  $Q_T = 22.5$  tu in 1 second.

Table 8 shows the summarized performance of the 3D Lorenz, Rössler, Chen, Liu and Chen, and MACM CSs in their DV by using the NAs of Euler (6), Heun (7) and (8), and RK4 (9)–(13)—the values of time complexity and its

equivalence are defined in ips. All the NAs can be executed by using dsPIC33 U2 because its program flash memory is 32 Kbytes, and it allows to execute large NAs.

3.3. *Embedded System with PIC32 Microcontroller.* A novel family of PIC32MZ microcontrollers is used to implement the

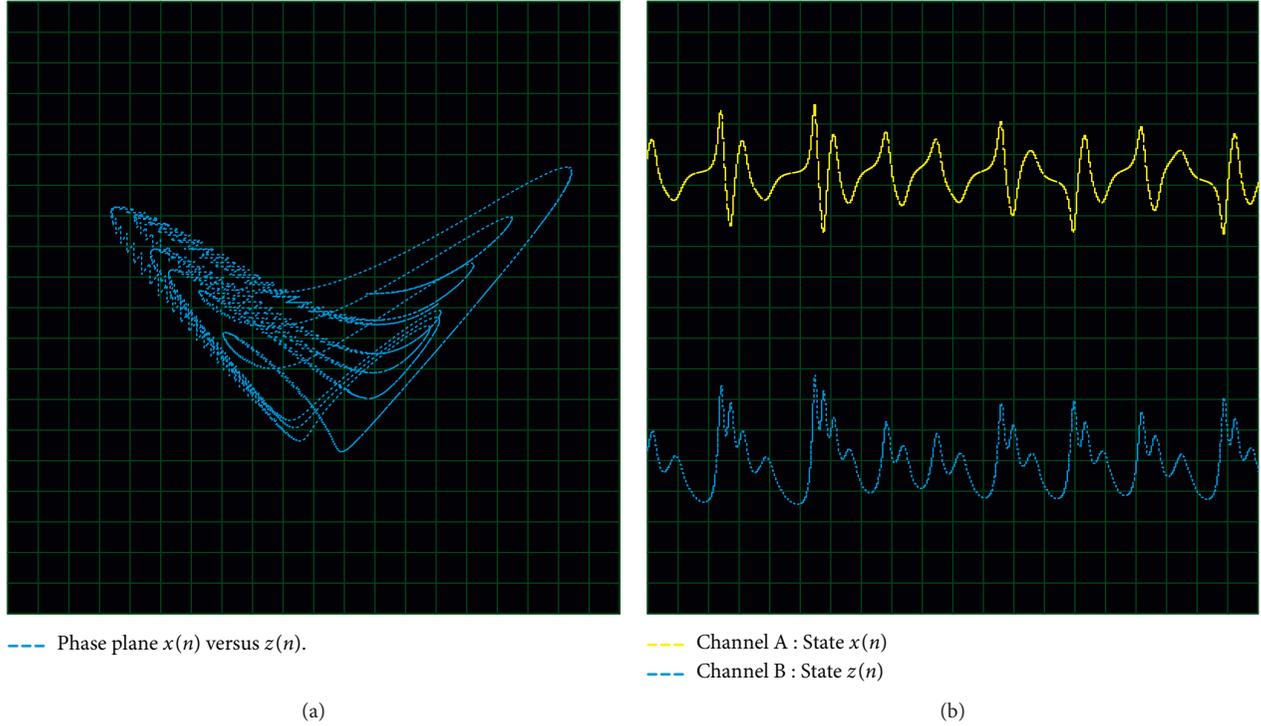


FIGURE 10: Simulation of ES in V2 of Liu and Chen in its DV by using  $\tau_{\max} = 0.002$ . (a) Phase plane  $x(n)$  versus  $z(n)$  and (b) evolution of state variables  $x(n)$  and  $z(n)$ .

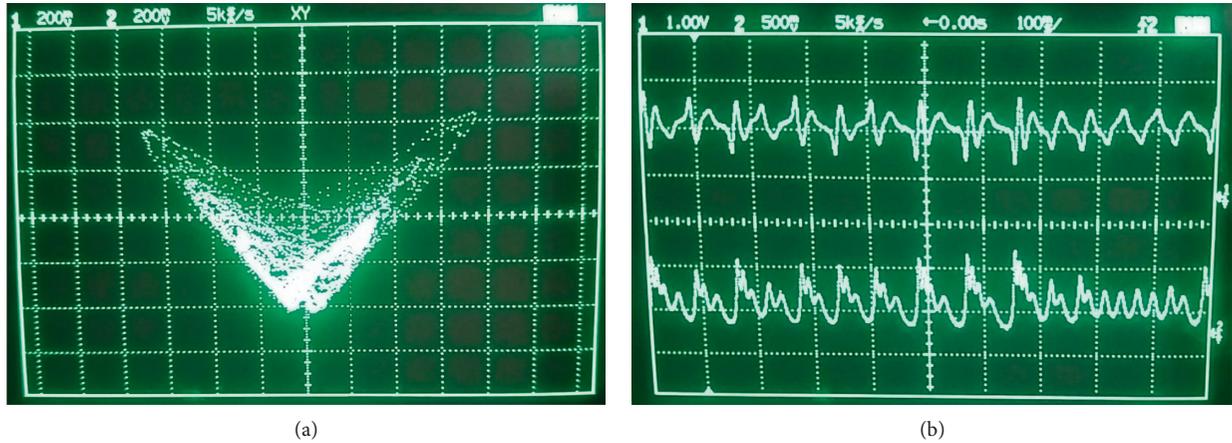


FIGURE 11: Implementation of ES in V2 of Liu and Chen CS (16) in its DV. (a) Phase plane  $x(n)$  versus  $z(n)$  and (b) evolution of state variables  $x(n)$  and  $z(n)$ .

V3 of ES. The PIC32MZ is not supported by the Proteus simulator; only Mikro C for PIC32 compiler was used to simulate and execute the NAs. Chen CS in its DV was chosen to carry out the test in the V3 of the ES by using Heun algorithm (7) and (8), and its algorithm is given by

$$\begin{aligned} x_{(n+1)}^* &= x_{(n)} + \tau(a(y_{(n)} - x_{(n)})), \\ y_{(n+1)}^* &= y_{(n)} + \tau((c-a)x_{(n)} - x_{(n)}z_{(n)} + cy_{(n)}), \\ z_{(n+1)}^* &= z_{(n)} + \tau(x_{(n)}y_{(n)} - bz_{(n)}), \end{aligned} \quad (18)$$

where

$$\begin{aligned} x_{(n+1)} &= x_{(n)} + \frac{\tau}{2}(a(y_{(n)} - x_{(n)}) + x_{(n+1)}^*), \\ y_{(n+1)} &= y_{(n)} + \frac{\tau}{2}((c-a)x_{(n)} - x_{(n)}z_{(n)} + cy_{(n)} + y_{(n+1)}^*), \\ z_{(n+1)} &= z_{(n)} + \frac{\tau}{2}(x_{(n)}y_{(n)} - bz_{(n)} + z_{(n+1)}^*). \end{aligned} \quad (19)$$

The performance of the algorithm (18) and (19) in the version 3 is  $t_c = 13.3 \mu\text{s}$ ,  $t_{Tg} = 3 \mu\text{s}$ ,  $T_{Td} = 16.3 \mu\text{s}$ , and

TABLE 8: Performance of ES in the V2 of the 3D Lorenz, Rössler, Chen, Liu and Chen, and MACM CSs in their DVs by using Euler (6), Heun (7) and (8), and RK4 (9)–(13) NAs.

CS	Euler		Heun		RK4	
	$T_{Td}$ ( $\mu$ s)	$f_{Td}$ (ips)	$T_{Td}$ ( $\mu$ s)	$f_{Td}$ (ips)	$T_{Td}$ ( $\mu$ s)	$f_{Td}$ (ips)
Lorenz	84	11905	138	7246	245	4082
Rössler	81	12345	132	7575	233	4291
Chen	84	11905	138	7246	245	4082
Liu and Chen	89	11236	143	6993	271	3690
MACM	84	11905	138	7246	245	4082

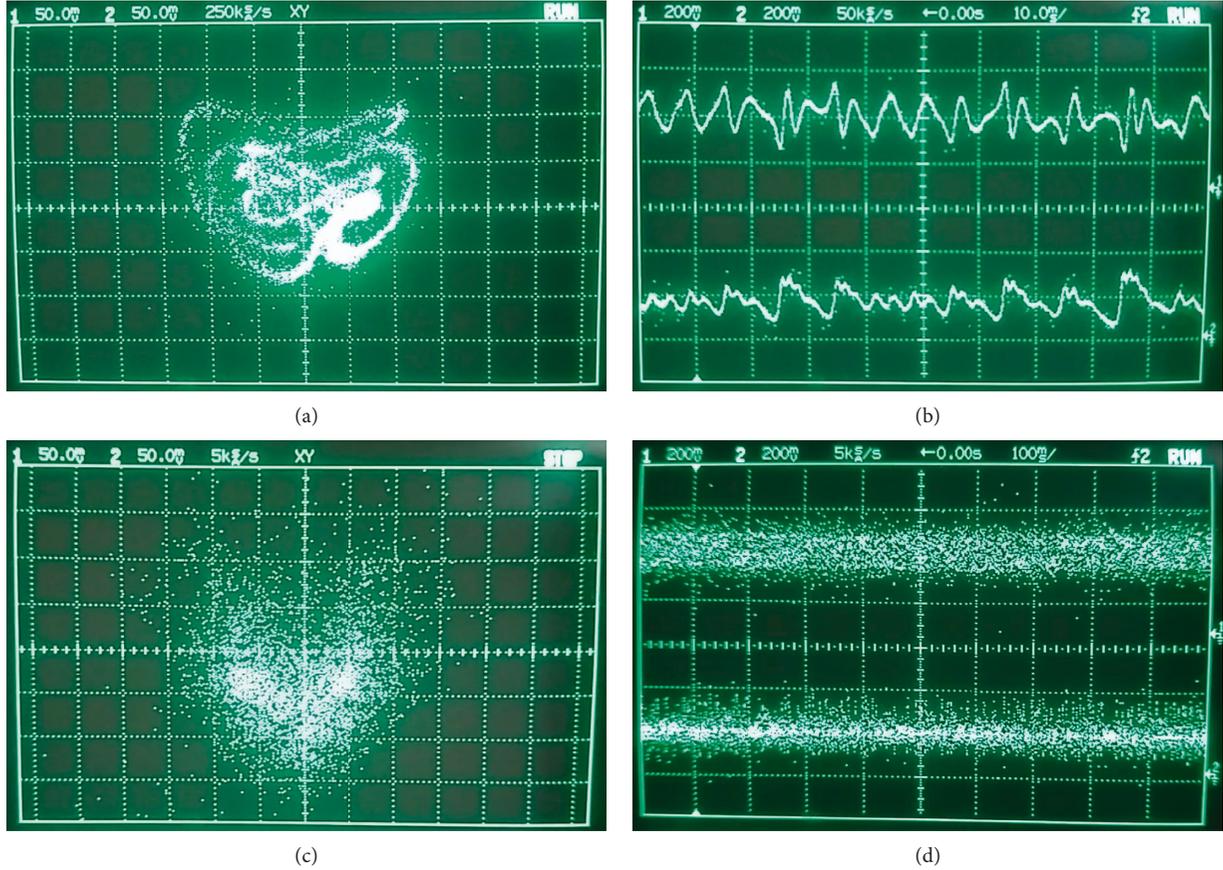


FIGURE 12: Implementation of ES in the V3 of Chen CS (18) and (19) in its DV. (a) Phase plane  $x_{(n)}$  versus  $z_{(n)}$ , (b) evolution of state variables  $x_{(n)}$  and  $z_{(n)}$  by using  $\tau = 0.001$ , (c) phase plane  $x_{(n)}$  versus  $z_{(n)}$ , and (d) evolution of state variables  $x_{(n)}$  and  $z_{(n)}$  by using  $\tau_{\max} = 0.017$ .

$f_{Td} = 61349$  ips. In order to obtain a comparison between the step sizes obtained, Figure 12 shows two implementations of the algorithm (18) and (19). For the first test, we considered a small step size  $\tau = 0.001$ ; this means that we obtained  $Q_T = 20$  time units in 1 second as is shown in Figure 12(b). For the second test, we used the higher step size  $\tau_{\max} = 0.017$  as it was described in Table 4. The maximum chaotic degradation is obtained for  $Q_T = 340$  time units in 1 second, as is shown in Figure 12(d).

Table 9 shows the time complexity and frequency, expressed in ips, and the performance of ES in V3 by using the NAs of Euler (6), Heun (7) and (8), and RK4 (9)–(13).

TABLE 9: Performance of ES in the V3 of the 3D Lorenz, Rössler, Chen, Liu and Chen, and MACM CSs in their DVs by using Euler (6), Heun (7) and (8), and RK4 (9)–(13) NAs.

CS	Euler		Heun		RK4	
	$T_{Td}$ ( $\mu$ s)	$f_{Td}$ (ips)	$T_{Td}$ ( $\mu$ s)	$f_{Td}$ (ips)	$T_{Td}$ ( $\mu$ s)	$f_{Td}$ (ips)
Lorenz	10.6	94339	16.3	61349	30	33333
Rössler	9.7	103092	12.5	80000	25.8	38759
Chen	10.6	94339	16.3	61349	30	33333
L. and Ch.	10.9	91743	17.1	58479	31.5	31746
MACM	10.6	94339	16.3	61349	30	33333

3.4. *Embedded System Implemented with FPGA by Using Nios Microcontroller.* We introduce a novel method to design an embedded microcontroller in FPGA U4 considering the

similar software and hardware conditions used in the previous implementations of ESs. We create a project in Quartus II (version 12.1) by using the  $Q_{\text{sys}}$  tool to obtain the hardware design in FPGA U4. The  $Q_{\text{sys}}$  tool is used to define the specifications of hardware described in a complex arrangement of modules inside FPGA U4; this hardware specification is named Entity, and it can be conducted using the block diagram of Quartus II, e.g., the hardware design of Entity U4 includes a microcontroller as the main processor (its fast version is referred to as Nios II/f which internal clock is configured to 150 MHz), one program memory, peripherals of control, and external ports, among others; to carry out the implementation of the Entity in an FPGA using the Nios II microcontroller, control modules, and other  $Q_{\text{sys}}$  tools, we recommend reviewing [47]. The hardware implementation is based on the specifications of the FPGA U4 that is included in the hardware of the Terasic DE2i-150 board.

Once the Entity is defined on the FPGA U4, the pins of the SPI control-bus are configured and addressing using the expansion header of the DE2i-150 board where the global peripheral input-output (GPIO) port is configured to write the DACs U4–U6 because the DE2i-150 board does not contain internal DACs. Figure 13 shows the block diagram and the result of the Entity design using  $Q_{\text{sys}}$  of Quartus II.

Once the hardware structure on FPGA U4 is finished, the Eclipse compiler (version IDE for C/C++ developers) is used to program and implement the NAs of the DV of the 3D Lorenz, Rössler, Chen, Liu and Chen, and MACM CSs in the V4 of the ES. For this example, we used the RK4 algorithm (9)–(13) to obtain the DV of the MACM CS; the algorithm is given by

$$\begin{aligned} k_1 &= -ax_{(n)} - by_{(n)}z_{(n)}, \\ l_1 &= -x_{(n)} + cy_{(n)}, \\ m_1 &= d - y_{(n)}^2 - z_{(n)}, \end{aligned} \quad (20)$$

where

$$\begin{aligned} k_2 &= -a\left(x_{(n)} + \frac{\tau}{2}k_1\right) - b\left(y_{(n)} + \frac{\tau}{2}l_1\right)\left(z_{(n)} + \frac{\tau}{2}m_1\right), \\ l_2 &= -\left(x_{5(n)} + \frac{\tau}{2}k_{15}\right) + c\left(y_{5(n)} + \frac{\tau}{2}l_{15}\right), \\ m_2 &= d - \left(y_{(n)} + \frac{\tau}{2}l_1\right)^2 - \left(z_{(n)} + \frac{\tau}{2}m_1\right), \\ k_3 &= -a\left(x_{(n)} + \frac{\tau}{2}k_2\right) - b\left(y_{(n)} + \frac{\tau}{2}l_2\right)\left(z_{(n)} + \frac{\tau}{2}m_2\right), \\ l_3 &= -\left(x_{(n)} + \frac{\tau}{2}k_2\right) + c\left(y_{(n)} + \frac{\tau}{2}l_2\right), \\ m_3 &= d - \left(y_{(n)} + \frac{\tau}{2}l_2\right)^2 - \left(z_{(n)} + \frac{\tau}{2}m_2\right), \end{aligned} \quad (21)$$

$$(22)$$

$$\begin{aligned} k_4 &= -a\left(x_{(n)} + \tau k_3\right) - b\left(y_{5(n)} + \tau l_3\right)\left(z_{(n)} + \tau m_3\right), \\ l_4 &= -\left(x_{(n)} + \tau k_3\right) + c\left(y_{(n)} + \tau l_3\right), \\ m_4 &= d - \left(y_{(n)} + \tau l_3\right)^2 - \left(z_{(n)} + \tau m_3\right), \end{aligned} \quad (23)$$

$$\begin{aligned} x_{(n+1)} &= x_{(n)} + \frac{\tau}{6}(k_1 + 2k_2 + 2k_3 + k_4), \\ y_{(n+1)} &= y_{(n)} + \frac{\tau}{6}(l_1 + 2l_2 + 2l_3 + l_4), \\ z_{(n+1)} &= z_{(n)} + \frac{\tau}{6}(m_1 + 2m_2 + 2m_3 + m_4). \end{aligned} \quad (24)$$

The performance of the algorithm (20)–(24) in the V4 is  $t_c = 156 \mu\text{s}$ ,  $t_{\text{Tg}} = 3 \mu\text{s}$ ,  $T_{\text{Td}} = 159 \mu\text{s}$ , and  $f_{\text{Td}} = 6289 \text{ ips}$ . As in the previous case, we conduct a comparison between the step sizes obtained. Figure 14 shows the result of two implementations of the algorithms (20)–(24) in the V4. For the first test, we considered a small step size  $\tau = 0.01$ ; Figure 14(b) shows a fewer number of  $Q_T = 62.89$  time units generated in 1 second. For the second test, we used the higher step size  $\tau_{\text{max}} = 0.547$  described in Table 5, a large number of time units are obtained  $Q_T = 3440.1$  in one second, and the maximum chaotic degradation is illustrated in Figure 14(d).

Table 10 shows the results of the ES in the V4; the time complexity and frequency, expressed in ips, were obtained by using the NAs of Euler (6), Heun (7) and (8), and RK4 (9)–(13).

**3.5. Results of Embedded System for V1–V4.** In order to summarise the studies presented in the previous section, we conducted a comparison considering the performance of the 4 versions in the ES of the 3D Lorenz, Rössler, Chen, Liu and Chen, and MACM CSs in their DVs. We used (15) to obtain the time units  $Q_T$  generated in 1 second considering the results obtained of  $\tau_{\text{max}}$  shown in Tables 3–5, and the results of  $T_{\text{Td}}$  and  $f_{\text{Td}}$  shown in Tables 7–10. We obtained the best performance of  $Q_T$  considering  $\tau_{\text{max}}$  of the 3D Lorenz, Rössler, Chen, Liu and Chen, and MACM CSs in their DVs using the NA of Euler, Heun, and RK4, and the summarized results are presented in Tables 11 and 12.

For a better understanding, Figure 15 illustrates the trajectories of the first state  $x_{(n)}$  of the 3D Lorenz, Rössler, Chen, Liu and Chen, and MACM CSs in their DVs by using the NA of Euler (8); the performance of each state  $x_{(n)}$  is implemented in the V1-PIC of ES—the results are expressed in the time-units quantity  $Q_T$  that the ES in the V1 generates in 1 second, and the  $t$  axe is included to compare the performance of CSs used. Figures 15(a), 15(d), and 15(e) show chaotic dynamics more compactly in the trajectories of state  $x_{(n)}$ . Otherwise, Figures 15(b) and 15(c) show trajectories of the state  $x_{(n)}$  less compact.

Figure 16 shows the same trajectories of the first state  $x_{(n)}$  of the 3D Lorenz, Rössler, Chen, Liu and Chen, and MACM CSs in their DVs by using the same Euler algorithm (8); now the performance is conducted in the V2-dsPIC of the ES—the results also are expressed in the time-units quantity

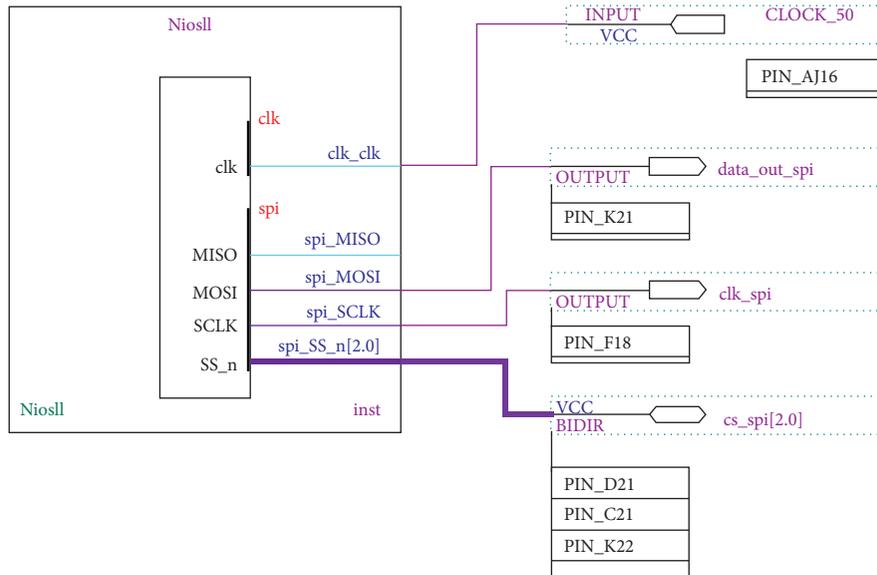


FIGURE 13: Schematic diagram of ES in V4 to implement the Nios II Entity and pins distribution on the FPGA U4.

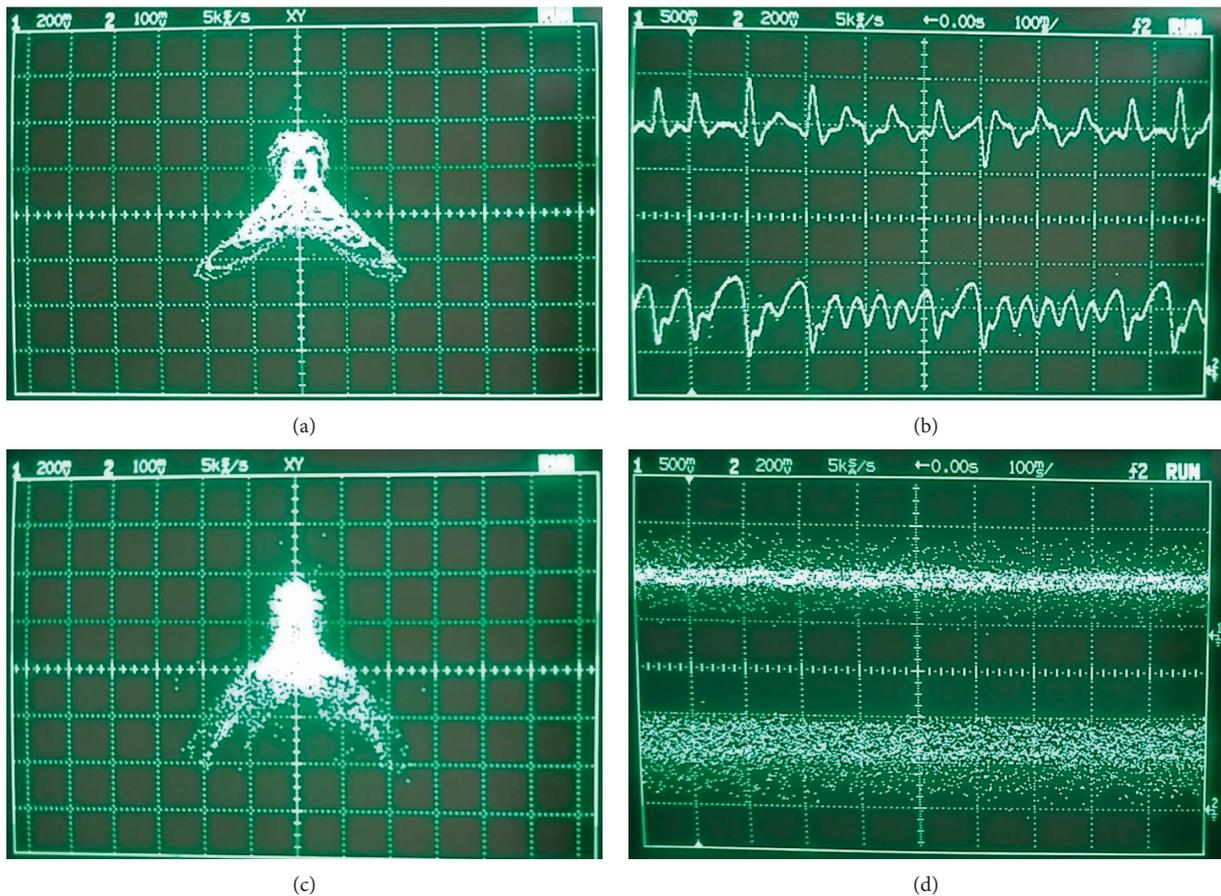


FIGURE 14: Implementation of ES in V4 of the MACM CS (20)–(24) in its DV. (a) Phase plane  $x_{(n)}$  versus  $z_{(n)}$ , (b) evolution of state variables  $x_{(n)}$  and  $z_{(n)}$  using  $\tau = 0.01$ , (c) phase plane  $x_{(n)}$  versus  $z_{(n)}$ , and (d) evolution of state variables  $x_{(n)}$  and  $z_{(n)}$  using  $\tau_{\max} = 0.057$ .

$Q_T$  that the ES in its version 2 generates in  $t = 1$  second. Rössler system presents only one nonlinearity, Figures 15(b) and 16(b) illustrate slow changes in their chaotic dynamics, which means that Rössler system is not tempting to be

implemented in ESs slower like the 8-bit microcontrollers. Figures 15(e) and 16(e) show that MACM CS (represented by (5) in its CV) is most rich in chaos, i.e., it is verified that this CS in its DV presents more rich chaotic dynamics than

TABLE 10: Performance of ES in the V4 of the 3D Lorenz, Rössler, Chen, Liu and Chen, and MACM CSs in their DVs by using Euler (6), Heun (7) and (8), and RK4 (9)–(13) NAs.

CS	Euler		Heun		RK4	
	$T_{Td}$ ( $\mu$ s)	$f_{Td}$ (ips)	$T_{Td}$ ( $\mu$ s)	$f_{Td}$ (ips)	$T_{Td}$ ( $\mu$ s)	$f_{Td}$ (ips)
Lorenz	40	25000	70	14286	159	6289
Rössler	37	27027	53	18868	135	7407
Chen	40	25000	70	14286	159	6289
L. and Ch.	45	22222	83	12048	167	5988
MACM	40	25000	70	14286	159	6289

TABLE 11: Performance of the ES expressed in  $Q_T$  of the 3D Lorenz, Rössler, Chen, Liu and Chen, and MACM CSs in their DVs using (6)–(8) for V1, and (6)–(13) for V2.

CS	$Q_T$ for V1-PIC		$Q_T$ for V2-dsPIC		
	Euler	Heun	Euler	Heun	RK4
Lorenz	11.729	6.9552	285.7	492.7	432.7
Rössler	2.9065	1.8115	61.7	1446.8	1077
Chen	0.977	0.5796	23.8	123.2	232.7
L. and Ch.	0.8064	0.4524	22.5	118.9	210.3
MACM	41.05	24.3432	1011.9	1652.1	2232.9

TABLE 12: Performance of the ES expressed in  $Q_T$  of V3-V4 for 3D Lorenz, Rössler, Chen, Liu and Chen, and MACM CSs in their DVs using (6)–(13).

CS	$Q_T$ for V3—PIC32			$Q_T$ for V4—FPGA		
	Euler	Heun	RK4	Euler	Heun	RK4
Lorenz	2264	4172	3533	600	971.4	666.6
Rössler	515	15280	9729	135	3603.8	1859.2
Chen	189	1043	1900	50	242.9	358.5
L. and Ch.	183	994	1810	44	204.8	341.3
MACM	8019	13988	18233	2125	3257.2	3440.1

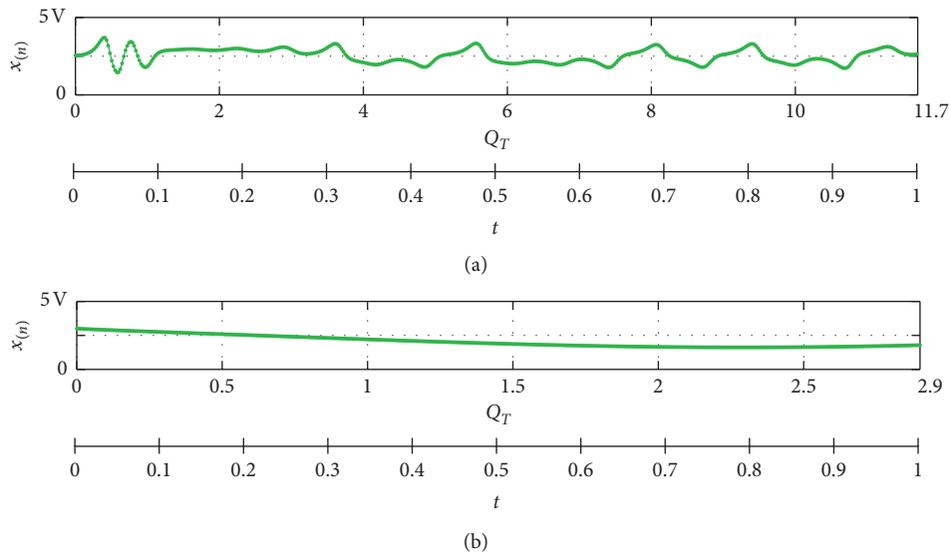


FIGURE 15: Continued.

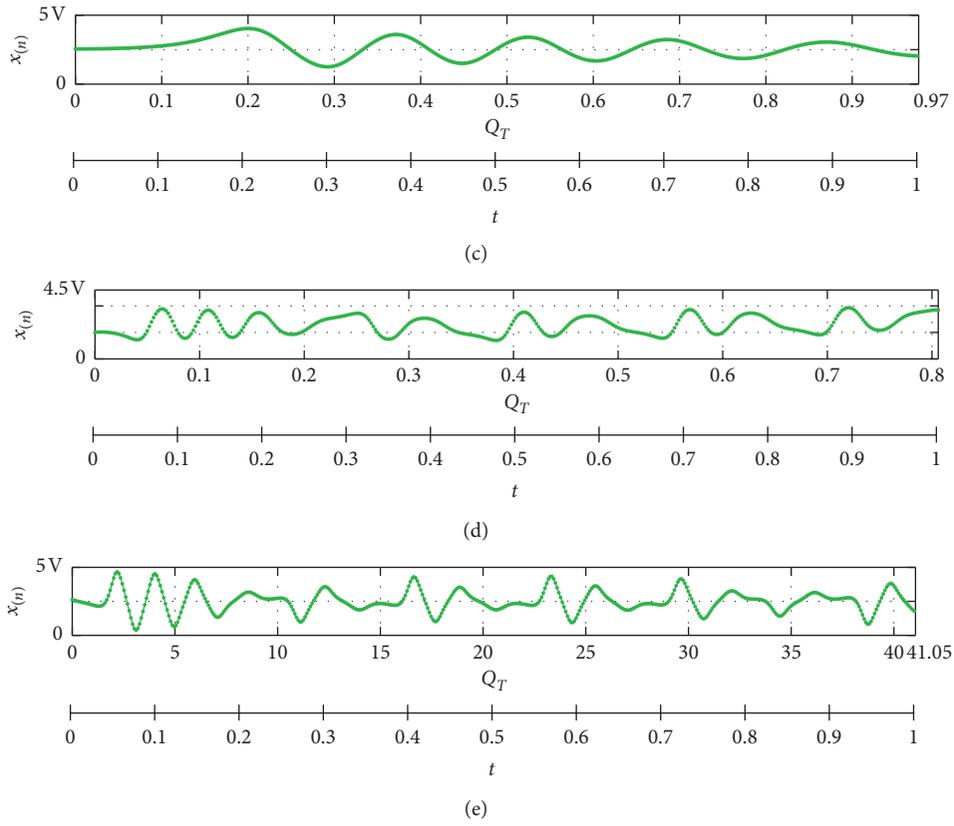


FIGURE 15: Time series of the five CSs of Lorenz, Rössler, Chen, Liu and Chen, and MACM in their DVs by using the Euler NA to compare the performance of version 1 PIC of ES expressed  $Q_T$  for  $t = 1$  second: (a)  $x_{(n)}$  of Lorenz, (b)  $x_{(n)}$  of Rössler, (c)  $x_{(n)}$  of Chen, (d)  $x_{(n)}$  of Liu and Chen, and (e)  $x_{(n)}$  of MACM.

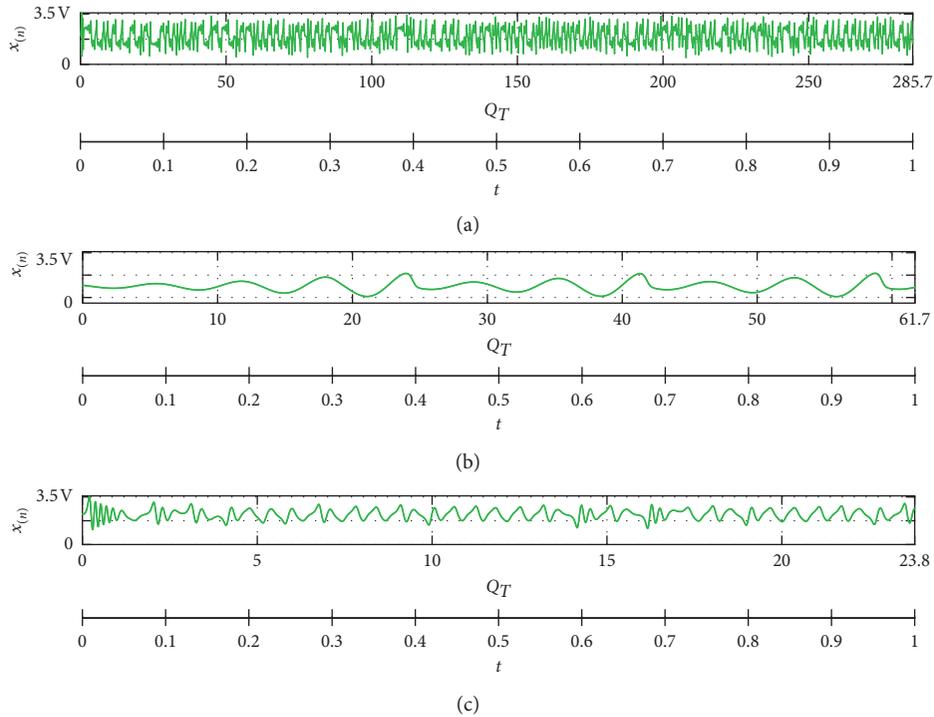


FIGURE 16: Continued.

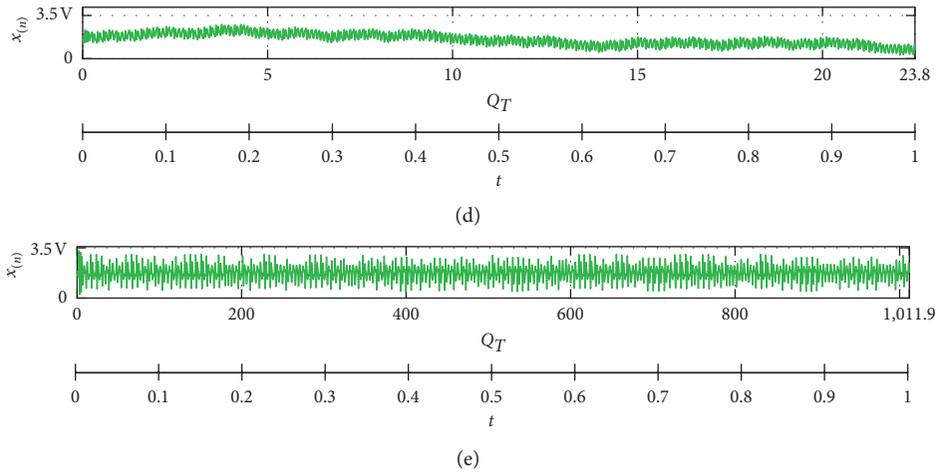


FIGURE 16: Time series of the five CSs of Lorenz, Rössler, Chen, Liu and Chen, and MACM in their DVs by using the Euler NA to compare the performance of version 2 dsPIC of ES expressed  $Q_T$  for  $t = 1$  second: (a)  $x_{(n)}$  of Lorenz, (b)  $x_{(n)}$  of Rössler, (c)  $x_{(n)}$  of Chen, (d)  $x_{(n)}$  of Liu and Chen, and (e)  $x_{(n)}$  of MACM.

the others, Lorenz, Rössler, Chen, and Liu and Chen CSs, in their DVs [29].

#### 4. Robustness in Digital Implementation of CSs in DV

The literature reports robustness in ESs considering the characteristics of software and hardware [57]. Regarding the hardware used in the 4 versions of the ES, it has two ways of powering: the first way is through the USB port connected to a laptop or desktop PC, e.g., for the V4 of FPGA device; the second way is using an external power supply for the V1–V3. The ESs V1–V4 energized using an external battery makes them portable, which allows the autonomy of each of them.

On the other hand, a robustness diagram based on the variation of two critical parameters in the five 3D CSs in their DVs using the NAs (6)–(13) was conducted. Figure 17 shows the diagram that determines the regions for two parameters which the existence of chaos is guaranteed for a specific step size obtained in Tables 3–5, each point in the diagram represents the maximum Lyapunov exponent ( $LE_{\max}$ ). If we have  $LE_{\max} > 0$ , the dynamics are chaotic denoted in yellow color; otherwise, the blue color is used, previous work was reported in [29]. Figures 17(a)–17(c) represent the variation of the parameters  $\sigma$  versus  $r$  of the Lorenz system in its DV, the parameter  $b$  was fixed in  $8/3$ . Figures 17(d)–17(f) represent the variation of the parameters  $b$  versus  $c$  of the Rössler system in its DV (the parameter  $a$  was fixed in 0.2). Figures 17(g)–17(i) represent the variation of the parameters  $b$  versus  $c$  of the Chen CS in its DV (the parameter  $a$  was fixed in 35). Figures 17(j)–17(l) represent the variation of the parameters  $b$  versus  $c$  of the Liu and Chen CS in its DV (the parameters were fixed with  $a = 5$ ,  $d_1 = -1$ , and  $d_2 = d_3 = 1$ ). Finally, Figures 17(m)–17(o) represent the variation of the parameters  $b$  versus  $d$  of the MACM CS in its DV (the parameters were fixed with  $a = 2$  and  $c = 0.5$ ). To generate Figures 17(a), 17(d), 17(g), 17(j), and 17(m), we use the Euler NA (6), for Figures 17(b), 17(e), 17(h), 17(k), and 17(n), we use the Heun algorithm (7) and

(8), and for Figures 17(c), 17(f), 17(i), 17(l), and 17(o), we used the NA of RK4 (9)–(13).

Furthermore, it is easy to note that if a value of step size  $\tau$  less than that considered in yellow color is used, then the chaos regions increase. Taking into account the fact that the preservation of chaos in the DV of the set of 5 CSs in their DVs is robust for the variation of two parameters, considering the characteristics of software and hardware the proposed ES in V1–V4, and the benefits of digital systems, as the elimination of the typical wear of the analog systems, it is stated that the electronical/digital implementation presented in this work is robust.

#### 5. Conclusions

In this paper, we have presented analytical, numerical, and experimental studies of chaos degradation of the Lorenz, Rössler, Chen, Liu and Chen, and MACM three-dimensional chaotic systems (CSs) in their discretized versions (DV) by using the numerical algorithms (NAs) of Euler, Heun, and fourth-order Runge–Kutta (RK4). We obtained a novel robustness diagram with the variation of two parameters of the five CSs in their DVs to guarantee the chaos existence where the maximum step size was found by using Euler, Heun, and fourth-order Runge–Kutta (RK4) NAs, the degradation studies showed that the DV of MACM CS exhibits higher chaotic degradation, while the Chen and Liu Chen presented a lower degradation. The step-size values founded can be used, e.g., for encryption applications, as one more parameter, considering the intervals shown in this paper to guarantee chaos. In addition, the step size obtained can be used in others families of 8-, 16-, and 32-bit microcontrollers, DSP, or FPGA where the DVs of these five 3D CSs studied are desired for general purposes.

The numerical studies of Root-Mean-Square Error (RMSE) have shown better performance in the Rössler system, it provided interesting and attractive results with respect to the DVs of the Lorenz, Chen, Liu and

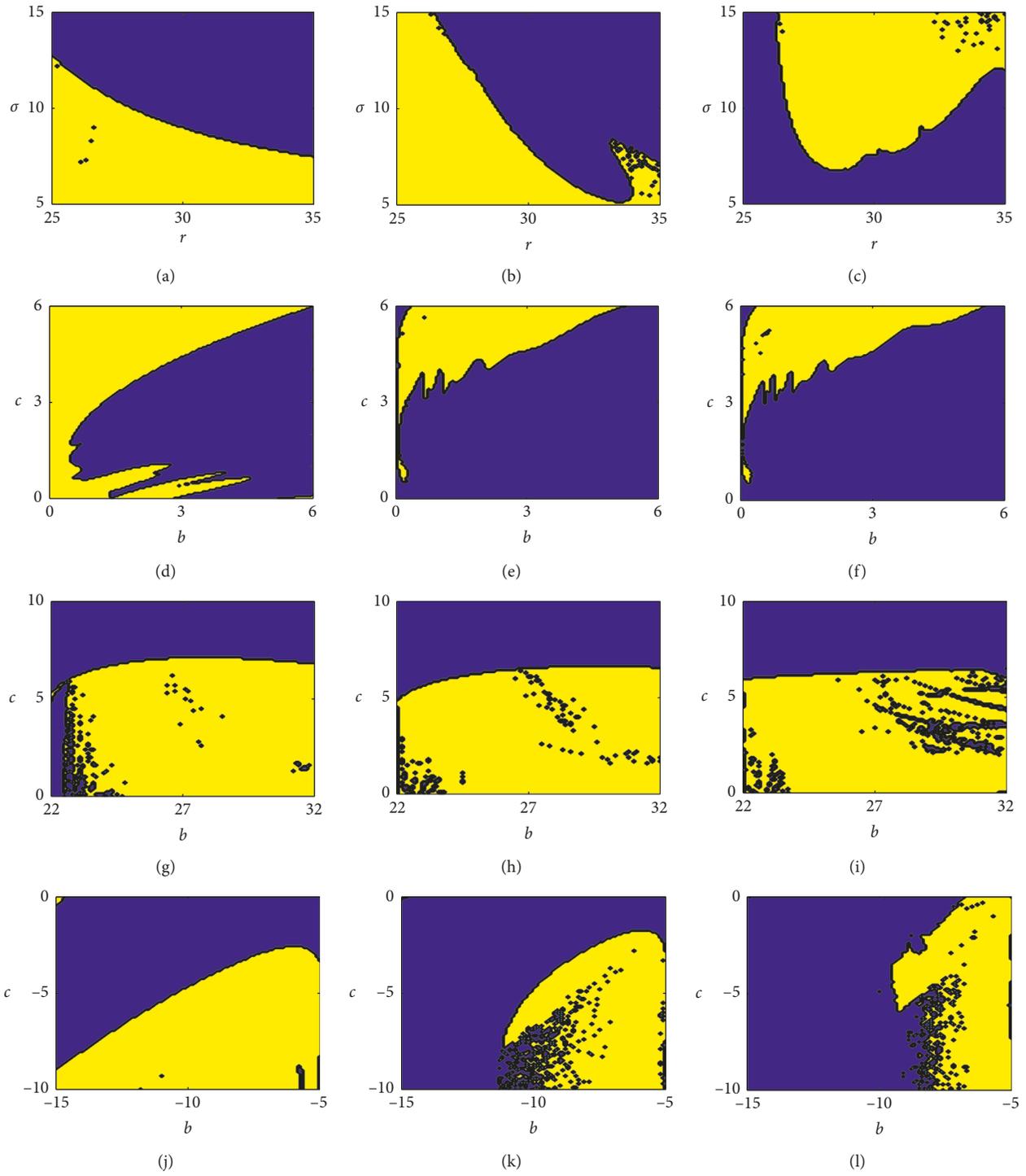


FIGURE 17: Continued.

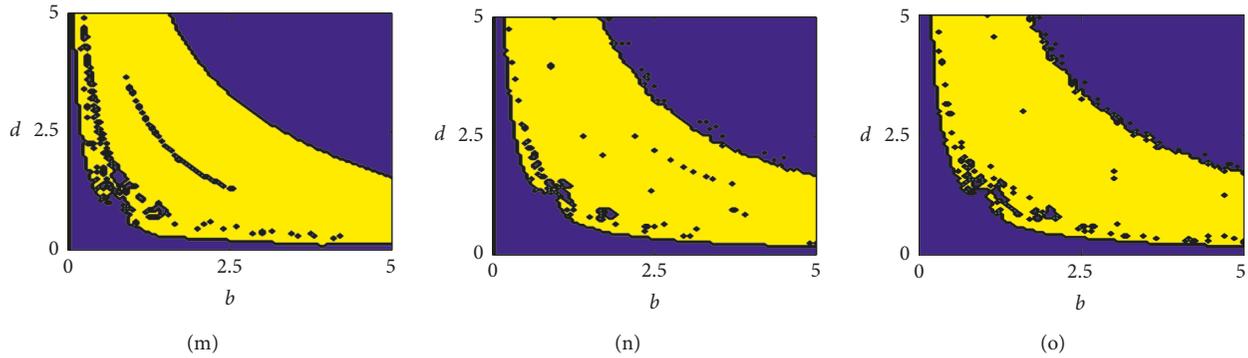


FIGURE 17: Robustness diagram of chaos existence varying two parameters at intervals of 0.05 and with  $\tau$  according to five 3D CSs in their DVs using the NA (6)–(13): chaos (yellow); no chaos (blue).

Chen, and MACM CSs using the numerical algorithms Euler, Heun, and RK4; it had slow changes in their dynamics and showed small variations in their trajectories to same initial conditions considering 30000 samples. Similar results of RMSE and step sizes were obtained from Chen and Liu and Chen systems although the Liu and Chen system has 3 nonlinearities, one more than Chen system.

All the results of the three-dimensional five chaotic systems in their discretized versions were implemented by using four versions of the embedded system (ES) where the 16-bit dsPIC33 showed better alternative to simulate, reproduce, and implement the numerical algorithms of Euler, Heun, and RK4. The 32-bit PIC32MZ presented the best performance in time complexity and is an interesting alternative to implement and obtain good performance for applications where iteration speed is desirable such as synchronization and multimedia encryption, among others.

As future work, complementary degradation studies of 4D hyperchaotic systems will be conducted for encryption and synchronization applications and their implementation in other families of the ESs.

## Data Availability

The data used to support the findings of this study are included within the article.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

This work was supported by the CONACYT, México, under Research Grant 166654 (A1-S-31628).

## References

- [1] S. H. Strogatz, *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering*, Perseus Books, Massachusetts, MA, USA, 1994.
- [2] J.-P. Eckmann and D. Ruelle, "Ergodic theory of chaos and strange attractors," *Reviews of Modern Physics*, vol. 57, no. 3, pp. 617–656, 1985.
- [3] A. Argyris, D. Syvridis, L. Larger et al., "Chaos-based communications at high bit rates using commercial fibre-optic links," *Nature*, vol. 438, no. 7066, pp. 343–346, 2005.
- [4] L. Gámez-Guzmán, C. Cruz-Hernández, R. M. López-Gutiérrez, and E. E. García-Guerrero, "Synchronization of Chua's circuits with multi-scroll attractors: application to communication," *Communications in Nonlinear Science and Numerical Simulation*, vol. 14, no. 6, pp. 2765–2775, 2009.
- [5] R. M. May, "Simple mathematical models with very complicated dynamics," *Nature*, vol. 261, no. 5560, pp. 459–467, 1976.
- [6] S. Boccaletti, C. Grebogi, Y.-C. Lai, H. Mancini, and D. Maza, "The control of chaos: theory and applications," *Physics Reports*, vol. 329, no. 3, pp. 103–197, 2000.
- [7] Ch. Wang, Ch. Fan, K. Feng, X. Huang, and Q. Ding, "Analysis of the time series generated by a new high-dimensional discrete chaotic system," *Complexity*, vol. 2018, Article ID 9818520, 11 pages, 2018.
- [8] A. Arellano-Delgado, R. López-Gutiérrez, C. Cruz-Hernández, C. Posadas-Castillo, L. Cardoza-Avenidaño, and H. Serrano-Guerrero, "Experimental network synchronization via plastic optical fiber," *Optical Fiber Technology*, vol. 19, no. 93, pp. 93–98, 2016.
- [9] M. Murillo-Escobar, C. Cruz-Hernández, F. Abundiz-Pérez, R. López-Gutiérrez, and O. Acosta Del Campo, "A RGB image encryption algorithm based on total plain image characteristics and chaos," *Signal Processing*, vol. 109, pp. 109–161, 2015.
- [10] Y. Nakamura and A. Sekiguchi, "The chaotic mobile robot," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 6, pp. 898–904, 2001.
- [11] L. Fan-Yi and L. Jia-Ming, "Chaotic radar using nonlinear laser dynamics," *IEEE Journal of Quantum Electronics*, vol. 40, no. 6, pp. 815–820, 2004.
- [12] K. Junxiang, Y. Lilin, H. Tongtong, H. Ye, X. Guangqiong, and H. Weisheng, "Time delay concealment in feedback chaotic systems with dispersion in loop," *IEEE Photonics Journal*, vol. 9, no. 2, Article ID 7200808, 2017.
- [13] M. Chen, S.-Y. Shao, P. Shi, and Y. Shi, "Disturbance-observer-based robust synchronization control for a class of fractional-order chaotic systems," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 64, no. 4, pp. 417–421, 2017.

- [14] C.-H. Huang, C.-H. Lin, and C.-L. Kuo, "Chaos synchronization-based detector for power-quality disturbances classification in a power system," *IEEE Transactions on Power Delivery*, vol. 26, no. 2, pp. 944–953, 2011.
- [15] G. Doménech, T. J. Kazmierski, R. Ruiz, and J. D. Ruiz, "Architectural synthesis of high level analogue VHDL-AMS descriptions using netlist extraction from parse trees," *Electronic Letters*, vol. 36, no. 20, pp. 1680–1682, 2000.
- [16] S. Sahin and C. Guzelis, "'Chaotification' of real systems by dynamic state feedback," *IEEE Antennas and Propagation Magazine*, vol. 52, no. 6, pp. 222–233, 2010.
- [17] A. Veeraraghavan, A. Srivastava, A. K. Roy-Chowdhury, and R. Chellappa, "Rate-invariant recognition of humans and their activities," *IEEE Transactions on Image Processing*, vol. 18, no. 6, pp. 1626–1639, 2009.
- [18] E. N. Lorenz, "Deterministic nonperiodic flow," *Journal of the Atmospheric Sciences*, vol. 20, no. 2, pp. 160–141, 1963.
- [19] O. E. RöSSLer, "An equation for continuous chaos," *Physics Letters A*, vol. 57, no. 5, pp. 397–398, 1976.
- [20] J. Lü and G. Chen, "A new chaotic attractor coined," *International Journal of Bifurcation and Chaos*, vol. 12, no. 3, pp. 659–661, 2002.
- [21] L. O. Chua, M. Komuro, and T. Matsumoto, "The double scroll family," *IEEE Transactions on Circuits and Systems*, vol. 33, no. 11, pp. 1073–1118.
- [22] Ch. Liu, T. Liu, L. Liu, and K. Liu, "A new chaotic attractor," *Chaos, Solitons & Fractals*, vol. 22, no. 5, pp. 1031–1038, 2005.
- [23] G. Chen and T. Ueta, "Yet another chaotic attractor," *International Journal of Bifurcation and Chaos*, vol. 9, no. 7, pp. 1465–1466, 1999.
- [24] W. B. Liu and G. Chen, "A new chaotic system and its generation," *International Journal of Bifurcation and Chaos*, vol. 12, pp. 261–267, 2002.
- [25] J. C. Sprott, "Some simple chaotic jerk functions," *American Journal of Physics*, vol. 65, no. 6, pp. 537–543, 1997.
- [26] J. C. Sprott, "Some simple chaotic flows," *Physical Review E*, vol. 50, no. 2, pp. R647–R650, 1994.
- [27] L. Xiong, Y. Lu, Y. Zhang, and X. Zhang, "A novel memductor-based chaotic system and its applications in circuit design and experimental validation," *Complexity*, vol. 2019, Article ID 3870327, 23 pages, 2019.
- [28] V.-T. Pham, S. Jafari, C. Volos, A. Giakoumis, S. Vaidyanathan, and T. Kapitaniak, "A chaotic system with equilibria located on the rounded square loop and its circuit implementation," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 63, no. 9, pp. 878–882, 2016.
- [29] R. Méndez-Ramírez, C. Cruz-Hernández, A. Arellano-Delgado, and R. Martínez-Clark, "A new simple chaotic Lorenz-type system and its digital realization using a TFT touch-screen display embedded system," *Complexity*, vol. 2017, Article ID 6820492, 16 pages, 2017.
- [30] M. Hénon, "A two-dimensional mapping with a strange attractor," *Communications in Mathematical Physics*, vol. 50, no. 1, pp. 69–77, 1976.
- [31] C. Cruz-Hernández and A. A. Martynyuk, "Advances in chaotic dynamics with applications," *Series: Stability, Oscillations, and Optimization of Systems*, Cambridge Scientific Publishers, vol. 4, p. 432, Cambridge, UK, 2010.
- [32] I. Koyuncu, A. T. Ozcerit, and I. Pehlivan, "Implementation of FPGA-based real time novel chaotic oscillator," *Nonlinear Dynamics*, vol. 77, no. 1-2, pp. 49–59, 2014.
- [33] M. S. Azzaz, C. Tanougast, S. Sadoudi, R. Fellah, and A. Dandache, "A new auto-switched chaotic system and its FPGA implementation," *Communications in Nonlinear Science and Numerical Simulation*, vol. 18, no. 7, pp. 1792–1804, 2016.
- [34] L. Cong and W. Xiaofu, "Design and realization of an FPGA-based generator for chaotic frequency hopping sequences," *IEEE Transactions on Circuits and Systems I*, vol. 48, no. 5, pp. 521–532, 2001.
- [35] W. Yang, W. Cao, T.-S. Chung, and J. Morris, *Applied Numerical Methods Using Matlab*, John Wiley & Sons, Hoboken, NJ, USA, 2005.
- [36] E. Köse and A. Mühürçü, "Realization of a digital chaotic oscillator by using a low cost microcontroller," *Engineering Review*, vol. 37, no. 3, pp. 341–348, 2017.
- [37] R. Méndez-Ramírez, A. Arellano-Delgado, C. Cruz-Hernández, F. Abundiz-Perez, and R. Martínez-Clark, "Chaotic digital cryptosystem by using SPI protocol and its dsPICs implementation," *Frontiers of Information Technology & Electronic Engineering*, vol. 19, no. 2, pp. 165–179, 2018.
- [38] R. Méndez-Ramírez, A. Arellano-Delgado, C. Cruz-Hernández, and R. M. Lopez-Gutiérrez, "Degradation analysis of generalized Chua's circuit generator of multi-scroll chaotic attractors and its implementation on PIC32," in *Proceedings of the Future Technologies Conference (FTC)*, pp. 1034–1039, San Francisco, CA, USA, December 2016.
- [39] G. S. Thenge, S. S. Mulla, and A. B. Patki, "System identification using chaos theory on ARM cortex M3 processor development board," in *Proceedings of the 2015 IEEE International Advance Computing Conference (IACC)*, pp. 768–771, Bangalore, India, June 2015.
- [40] B. Cai, G. Wang, and F. Yuan, "Pseudo random sequence generation from a new chaotic system," in *Proceedings of the 2015 IEEE 16th International Conference on Communication Technology (ICCT)*, pp. 863–867, Hangzhou, China, October 2015.
- [41] M. L. Barakat, A. S. Mansingka, A. G. Radwan, and K. N. Salama, "Hardware stream cipher with controllable chaos generator for colour image encryption," *IET Image Processing*, vol. 8, no. 1, pp. 33–43, 2014.
- [42] E. Tlelo-Cuautle, J. J. Rangel-Magdaleno, A. D. Pano-Azucena, P. J. Obeso-Rodelo, and J. C. Nunez-Perez, "FPGA realization of multi-scroll chaotic oscillators," *Communications in Nonlinear Science and Numerical Simulation*, vol. 27, no. 1–3, pp. 66–80, 2015.
- [43] Z. Lin, J. Liu, J. Lian, Y. Ma, and X. Zhang, "A novel fast image encryption algorithm for embedded systems," *Multimedia Tools and Applications*, vol. 78, pp. 6969–6987, 2019.
- [44] A. Flores-Vergara, E. E. García-Guerrero, E. Inzunza-González et al., "Implementing a chaotic cryptosystem in a 64-bit embedded system by using multiple-precision arithmetic," *Nonlinear Dynamics*, vol. 95, no. 1, pp. 497–516, 2019.
- [45] Microchip Technology Inc., AN575. *IEEE-754 Compliant Floating Point Routines, DS00575B*, Microchip Technology Inc., Chandler, AZ, USA, 1997.
- [46] IEEE Computer Society, *IEEE Standard for Floating-point Arithmetic-Redline. IEEE Std 754-2008—Redline*, IEEE Computer Society, New York, NY, USA, 2008.
- [47] J. M. McNichols, E. J. Balster, W. F. Turri, and K. L. Hill, "Design and implementation of an embedded NIOS II system for JPEG2000 tier II encoding," *International Journal of Reconfigurable Computing*, vol. 2013, Article ID 140234, 9 pages, 2013.
- [48] S. Chernyi and A. Zhilenkov, "Modeling of complex structures for the ship's power complex using Xilinx system," *Transport and Telecommunication Journal*, vol. 16, no. 1, pp. 73–82, 2015.

- [49] B. Muthuswamy and S. Banerjee, "A route to chaos using FPGAs," *Experimental Observations*, Vol. 1, Springer, Berlin, Germany, 2014.
- [50] A. Wolf, J. B. Swift, H. L. Swinney, and J. A. Vastano, "Determining Lyapunov exponents from a time series," *Physica D: Nonlinear Phenomena*, vol. 16, no. 3, pp. 285–317, 1985.
- [51] K. Briggs, "An improved method for estimating Liapunov exponents of chaotic time series," *Physics Letters A*, vol. 151, no. 1-2, pp. 27–32, 1990.
- [52] J. R. Dormand and P. J. Prince, "A family of embedded Runge-Kutta formulae," *Journal of Computational and Applied Mathematics*, vol. 6, no. 1, pp. 19–26, 1980.
- [53] L. F. Shampine and M. W. Reichelt, "The MATLAB ODE suite," *SIAM Journal on Scientific Computing*, vol. 18, no. 1, pp. 1–22, 1997.
- [54] Mikroelektronika, *Programing dsPIC MCUs in C*, Mikroelektronika, Belgrade, Serbia, 2018.
- [55] A. Datta, D. Mukherjee, and H. Saha, "A dsPIC based novel digital sinusoidal pulse-width modulation technique for voltage source inverter applications," *Microprocessors and Microsystems*, vol. 38, no. 7, pp. 649–658, 2014.
- [56] M. A. Murillo-Escobar, C. Cruz-Hernández, L. Cardoza-Avenidaño, and R. Méndez-Ramírez, "A novel pseudorandom number generator based on pseudo randomly enhanced logistic map," *Nonlinear Dynamics*, vol. 86, no. 1, pp. 407–425, 2017.
- [57] S. Fraser, D. Campara, C. Chilly et al., "Fostering software robustness in an increasingly hostile world," in *Proceedings of the Companion to the 20th Annual ACM SIGPLAN Conference*, San Diego, CA, USA, October 2005.

