Research Article

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

Rodrigo Méndez-Ramírez, Adrian Arellano-Delgado, Miguel Murillo-Escobar, and César Cruz-Hernández

1Scientific Research and Advanced Studies Center of Ensenada, B. C. (CICESE), 22080 Ensenada, Mexico
2Engineering, Architecture and Design Faculty, Autonomous Baja California University (CONACYT-UABC), Ensenada, BC, Mexico

Correspondence should be addressed to César Cruz-Hernández; 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 nother 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

\[
\begin{align*}
\dot{x} &= \sigma (y - x), \\
\dot{y} &= rx - xy - y, \\
\dot{z} &= xy -bz,
\end{align*}
\]

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

\[
\begin{align*}
\dot{x} &= -y -z, \\
\dot{y} &= x + ay, \\
\dot{z} &= b + z(x - c),
\end{align*}
\]

where \( x, y, z \) 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

\[
\begin{align*}
\dot{x} &= a(y - x), \\
\dot{y} &= (c - a)x - xz + cy, \\
\dot{z} &= xy -bz,
\end{align*}
\]

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]

\[
\begin{align*}
\dot{x} &= ax + d_1yz, \\
\dot{y} &= cy + d_2xz, \\
\dot{z} &= bz + d_3xy,
\end{align*}
\]

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
\[
\begin{align*}
\dot{x} &= -ax - byz, \\
\dot{y} &= -x + cy, \\
\dot{z} &= d - y^2 - z, 
\end{align*}
\]

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 describes 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

\[
\begin{align*}
\hat{x}_{(n+1)} &= x_n + \tau f(x_n, y_n, z_n), \\
\hat{y}_{(n+1)} &= y_n + \tau g(x_n, y_n, z_n), \\
\hat{z}_{(n+1)} &= z_n + \tau h(x_n, y_n, z_n),
\end{align*}
\]

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:

\[
\begin{align*}
\dot{x}_{(n+1)} &= x_n + \frac{\tau}{2} \left( f(x_n, y_n, z_n) + f^*_{(n+1)} \right), \\
\dot{y}_{(n+1)} &= y_n + \frac{\tau}{2} \left( g(x_n, y_n, z_n) + g^*_{(n+1)} \right), \\
\dot{z}_{(n+1)} &= z_n + \frac{\tau}{2} \left( h(x_n, y_n, z_n) + h^*_{(n+1)} \right),
\end{align*}
\]

where

\[
\begin{align*}
x_{(n+1)} &= x_n + \frac{\tau}{2} \left( f(x_n, y_n, z_n) + f^*_{(n+1)} \right), \\
y_{(n+1)} &= y_n + \frac{\tau}{2} \left( g(x_n, y_n, z_n) + g^*_{(n+1)} \right), \\
z_{(n+1)} &= z_n + \frac{\tau}{2} \left( h(x_n, y_n, z_n) + h^*_{(n+1)} \right),
\end{align*}
\]

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

\[
\begin{align*}
x_{(n+1)} &= x_n + \frac{\tau}{6} \left( k_1 + 2k_2 + 2k_3 + k_4 \right), \\
y_{(n+1)} &= y_n + \frac{\tau}{6} \left( l_1 + 2l_2 + 2l_3 + l_4 \right), \\
z_{(n+1)} &= z_n + \frac{\tau}{6} \left( m_1 + 2m_2 + 2m_3 + m_4 \right),
\end{align*}
\]

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

\[
\begin{align*}
k_1 &= f(x_n, y_n, z_n), \\
l_1 &= g(x_n, y_n, z_n), \\
m_1 &= h(x_n, y_n, z_n),
\end{align*}
\]

\[
\begin{align*}
k_2 &= f(x_n + \frac{\tau}{2} k_1, y_n + \frac{\tau}{2} l_1, z_n + \frac{\tau}{2} m_1), \\
l_2 &= g(x_n + \frac{\tau}{2} k_1, y_n + \frac{\tau}{2} l_1, z_n + \frac{\tau}{2} m_1), \\
m_2 &= h(x_n + \frac{\tau}{2} k_1, y_n + \frac{\tau}{2} l_1, z_n + \frac{\tau}{2} m_1),
\end{align*}
\]

\[
\begin{align*}
k_3 &= f(x_n + \frac{\tau}{2} k_2, y_n + \frac{\tau}{2} l_2, z_n + \frac{\tau}{2} m_2), \\
l_3 &= g(x_n + \frac{\tau}{2} k_2, y_n + \frac{\tau}{2} l_2, z_n + \frac{\tau}{2} m_2), \\
m_3 &= h(x_n + \frac{\tau}{2} k_2, y_n + \frac{\tau}{2} l_2, z_n + \frac{\tau}{2} m_2),
\end{align*}
\]

\[
\begin{align*}
k_4 &= f(x_n + \tau k_3, y_n + \tau l_3, z_n + \tau m_3), \\
l_4 &= g(x_n + \tau k_3, y_n + \tau l_3, z_n + \tau m_3), \\
m_4 &= h(x_n + \tau k_3, y_n + \tau l_3, z_n + \tau m_3),
\end{align*}
\]

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 \(\tau_{\text{max}}\) 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 \(\tau_{\text{max}}\) 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, \(\tau_{\text{max}}\), 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 \(\tau_{\text{max}}\), and Chen and Liu and Chen CSs
Table 1: Parameters, characteristics, and ICs of the five 3D CSs (1)–(5).

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

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

<table>
<thead>
<tr>
<th>CS</th>
<th>(\tau_{\text{max}})</th>
<th>LE(_1)</th>
<th>LE(_2)</th>
<th>LE(_3)</th>
<th>(D_{\text{KY}})</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lorenz (1)</td>
<td>(\leq 0.024)</td>
<td>0.91</td>
<td>0</td>
<td>-14.47</td>
<td>2.062</td>
</tr>
<tr>
<td>Rössler (2)</td>
<td>(\leq 0.005)</td>
<td>49(\mu)</td>
<td>34.9(\mu)</td>
<td>-5.53</td>
<td>2.0142</td>
</tr>
<tr>
<td>Chen (3)</td>
<td>(\leq 0.002)</td>
<td>3.8(\mu)</td>
<td>0.11(\mu)</td>
<td>-23.1(\mu)</td>
<td>2.169</td>
</tr>
<tr>
<td>Liu and Chen (4)</td>
<td>(\leq 0.002)</td>
<td>197(\mu)</td>
<td>-53.2(\mu)</td>
<td>-18.7(\mu)</td>
<td>2.1029</td>
</tr>
<tr>
<td>MACM (5)</td>
<td>(\leq 0.085)</td>
<td>0.05</td>
<td>-64.5(\mu)</td>
<td>-0.243</td>
<td>2.099</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>CS</th>
<th>(\tau_{\text{max}})</th>
<th>LE(_1)</th>
<th>LE(_2)</th>
<th>LE(_3)</th>
<th>(D_{\text{KY}})</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lorenz (1)</td>
<td>(\leq 0.068)</td>
<td>86.6(\mu)</td>
<td>-2.016(\mu)</td>
<td>-0.623(\mu)</td>
<td>2.1358</td>
</tr>
<tr>
<td>Rössler (2)</td>
<td>(\leq 0.191)</td>
<td>0.0273</td>
<td>196(\mu)</td>
<td>-0.24122</td>
<td>2.1141</td>
</tr>
<tr>
<td>Chen (3)</td>
<td>(\leq 0.017)</td>
<td>38.7(\mu)</td>
<td>23.67(\mu)</td>
<td>-0.19558</td>
<td>2.1984</td>
</tr>
<tr>
<td>Liu and Chen (4)</td>
<td>(\leq 0.017)</td>
<td>17.41(\mu)</td>
<td>-17.88(\mu)</td>
<td>-0.15542</td>
<td>2.112</td>
</tr>
<tr>
<td>MACM (5)</td>
<td>(\leq 0.228)</td>
<td>95.29(\mu)</td>
<td>315.64(\mu)</td>
<td>-0.476</td>
<td>2.2006</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>CS</th>
<th>(\tau_{\text{max}})</th>
<th>LE(_1)</th>
<th>LE(_2)</th>
<th>LE(_3)</th>
<th>(D_{\text{KY}})</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lorenz (1)</td>
<td>(\leq 0.1076)</td>
<td>64.17(\mu)</td>
<td>-4.03(\mu)</td>
<td>-1.192(\mu)</td>
<td>2.0504</td>
</tr>
<tr>
<td>Rössler (2)</td>
<td>(\leq 0.251)</td>
<td>35.53(\mu)</td>
<td>339.62(\mu)</td>
<td>-0.402</td>
<td>2.0891</td>
</tr>
<tr>
<td>Chen (3)</td>
<td>(\leq 0.057)</td>
<td>97.48(\mu)</td>
<td>-2.949(\mu)</td>
<td>-0.6509</td>
<td>2.1452</td>
</tr>
<tr>
<td>Liu and Chen (4)</td>
<td>(\leq 0.057)</td>
<td>55.52(\mu)</td>
<td>-1.83(\mu)</td>
<td>-0.389</td>
<td>2.0974</td>
</tr>
<tr>
<td>MACM (5)</td>
<td>(\leq 0.547)</td>
<td>0.23081</td>
<td>-48.55(\mu)</td>
<td>-1.0098</td>
<td>2.1805</td>
</tr>
</tbody>
</table>

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

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:

\[
\text{RMSE} = \sqrt{\frac{1}{N} \sum_{n=1}^{N} (x_{e(n)} - x_{p(n)})^2},
\]

where the state variable \(x_{e(n)}\) is referred to as the estimator value of the CV of CSs (1)–(5), the state variable \(x_{p(n)}\) 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...
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, re-
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 \( x_{(t_0)} \) 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 algo-

\[\begin{align*}
\text{Lorenz} &\quad 0.024 \\
\text{Rössler} &\quad 0.005 \\
\text{Chen} &\quad 0.057 \\
\text{Liu and Chen} &\quad 0.057 \\
\text{MACM} &\quad 0.085 \\
\end{align*}\]

Figure 1: \( \tau_{\text{max}} \) summary of chaotic degradation of the five 3D CSs in their DVs.

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 micro-
controllers, 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 \( x(t_0), y(t_0) \) and \( z(t_0) \) 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 imple-
mentation 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

3. Digital Implementation

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

$$Q_T = \tau \frac{1}{T_{Td}} = \tau f_{Td} = \tau \frac{1}{t_c + t_g} = \tau \frac{1}{t_c + \sum_{j=1}^{N} t_{dac(j)}}. \quad (15)$$

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).
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
using the main core of U1, U2, U3, or U4; the total-graphic time \( t_{fg} \) 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 \( f_{fg} \) and the time \( T_{td} \) are obtained experimentally when the program of NA used is executed in the 4 versions of the ES. Only \( f_c \) depends on the size of the NA used. The size of \( f_{fg} \) depends on the configuration of SPI protocol used in U1–U4.

The equation (15) is a reference to determine the performance of the proposed ES in its four versions V1–V4. In this study, we need to obtain a smaller period of time to determine a greater number of iterations \( f_{fs} \) and the time \( T_{fs} \) are referred by \( t_{Tdac(1–3)} \), respectively. We considered the higher step size \( T_{td} \) and the time \( T_{fs} \) depends on the size of the NA used. 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 \( f_{fs} \), which is very attractive for applications based in chaos such as: master key definition, encryption, and secure communications [9, 37, 43, 56].

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

<table>
<thead>
<tr>
<th>Peripheral number</th>
<th>SPI mode, hardware description</th>
</tr>
</thead>
<tbody>
<tr>
<td>U1</td>
<td>Master, 8-bit PIC16F874A microcontroller</td>
</tr>
<tr>
<td>U2</td>
<td>Master, 16-bit dsPIC33FJ32MC204 microcontroller</td>
</tr>
<tr>
<td>U3</td>
<td>Master, 32-bit PIC32MZ2048ECM064 microcontroller</td>
</tr>
<tr>
<td>U4</td>
<td>Master, EP4CGX150DF31C7 Cyclone IV GX FPGA</td>
</tr>
<tr>
<td>U5</td>
<td>Slave 1, DAC MCP4921 shows ( x(t) )</td>
</tr>
<tr>
<td>U6</td>
<td>Slave 2, DAC MCP4921 shows ( y(t) )</td>
</tr>
<tr>
<td>U7</td>
<td>Slave 3, DAC MCP4921 shows ( z(t) )</td>
</tr>
</tbody>
</table>

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

\[
\begin{align*}
    x_{(n+1)} &= x_n + \tau \left( \sigma (y_n - x_n) \right), \\
    y_{(n+1)} &= y_n + \tau \left( r x_n - x_n z_n - y_n \right), \\
    z_{(n+1)} &= z_n + \tau \left( x_n y_n - b z_n \right).
\end{align*}
\]

Figure 7 shows the results of the Lorenz system simulation using 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 \( V_{dd} = +5 \) V and \( V_{ss} = 0 \) 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 \( T_{td} = 2046 \) \( \mu s \) and \( f_{td} = 488.7 \) ips; this means that we can obtain \( Q_T = 11.7 \) 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 \( x(t) \) and \( z(t) \) of Lorenz system (16) for 1 second. We experimentally obtained \( t_c = 1989 \) \( \mu s \), \( t_{fg} = 57 \mu s \), \( T_{td} = 2046 \) \( \mu s \), and \( f_{td} = 488.7 \) 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.
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 $V_{dd} = +3.3\,\text{V}$ and $V_{ss} = 0\,\text{V}$, 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:

$$
\begin{align*}
    x_{n+1} &= x_n + \tau (ax_n + d_1y_nz_n), \\
    y_{n+1} &= y_n + \tau (cy_n + d_2x_nz_n), \\
    z_{n+1} &= z_n + \tau (bz_n + d_3x_ny_n).
\end{align*}
$$

(17)

Figure 10 shows the simulation results conducted in Proteus of the DV of Liu and Chen CS (17) by using dsPIC33.
According to the results shown in Table 3, the higher step size \( \tau_{\text{max}} = 0.002 \) was chosen, we obtained \( t_c = 237 \mu s \), \( t_{Tg} = 8 \mu s \), \( T_{Td} = 245 \mu s \), and \( f_{Td} = 4082 \text{ips} \); this means that we can reproduce \( Q_T = 8.164 \text{tu} \) 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 \( \tau_{\text{max}} = 0.002 \), we obtained \( t_c = 86 \mu s \), \( t_{Tg} = 3 \mu s \), \( T_{Td} = 89 \mu s \), and \( f_{Td} = 4082 \text{ips} \).
and \( f_{td} = 11236 \) ips; this means that the hardware implementation in the V2 shows better performance because we obtained \( Q_T = 22.5 \) t.u 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
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

\[ x^{*}_{(n+1)} = x_{(n)} + \tau \left( a y_{(n)} - x_{(n)} \right), \]
\[ y^{*}_{(n+1)} = y_{(n)} + \tau \left( (c-a)x_{(n)} - x_{(n)}z_{(n)} + cy_{(n)} \right), \]
\[ z^{*}_{(n+1)} = z_{(n)} + \tau \left( x_{(n)}y_{(n)} - bz_{(n)} \right), \]

where

\[ x_{(n+1)} = x_{(n)} + \frac{\tau}{2} \left( a y_{(n)} - x_{(n)} + x^{*}_{(n+1)} \right), \]
\[ y_{(n+1)} = y_{(n)} + \frac{\tau}{2} \left( (c-a)x_{(n)} - x_{(n)}z_{(n)} + cy_{(n)} + y^{*}_{(n+1)} \right), \]
\[ z_{(n+1)} = z_{(n)} + \frac{\tau}{2} \left( x_{(n)}y_{(n)} - bz_{(n)} + z^{*}_{(n+1)} \right). \]

The performance of the algorithm (18) and (19) in the version 3 is \( t_c = 13.3 \mu s, \ t_{TB} = 3 \mu s, \ T_{TD} = 16.3 \mu s, \) 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.

<table>
<thead>
<tr>
<th>CS</th>
<th>Euler</th>
<th>Heun</th>
<th>RK4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$T_{td}$ (μs)</td>
<td>$f_{td}$ (ips)</td>
<td>$T_{td}$ (μs)</td>
</tr>
<tr>
<td>Lorenz</td>
<td>84</td>
<td>11905</td>
<td>138</td>
</tr>
<tr>
<td>Rössler</td>
<td>81</td>
<td>12345</td>
<td>132</td>
</tr>
<tr>
<td>Chen</td>
<td>84</td>
<td>11905</td>
<td>138</td>
</tr>
<tr>
<td>Liu and Chen</td>
<td>89</td>
<td>11236</td>
<td>143</td>
</tr>
<tr>
<td>MACM</td>
<td>84</td>
<td>11905</td>
<td>138</td>
</tr>
</tbody>
</table>

$T_{td} = 61349$ 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.

<table>
<thead>
<tr>
<th>CS</th>
<th>Euler</th>
<th>Heun</th>
<th>RK4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$T_{td}$ (μs)</td>
<td>$f_{td}$ (ips)</td>
<td>$T_{td}$ (μs)</td>
</tr>
<tr>
<td>Lorenz</td>
<td>10.6</td>
<td>94339</td>
<td>16.3</td>
</tr>
<tr>
<td>Rössler</td>
<td>9.7</td>
<td>103092</td>
<td>12.5</td>
</tr>
<tr>
<td>Chen</td>
<td>10.6</td>
<td>94339</td>
<td>16.3</td>
</tr>
<tr>
<td>Liu and Ch.</td>
<td>10.9</td>
<td>91743</td>
<td>17.1</td>
</tr>
<tr>
<td>MACM</td>
<td>10.6</td>
<td>94339</td>
<td>16.3</td>
</tr>
</tbody>
</table>

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 Qsys tool to obtain the hardware design in FPGA U4. The Qsys 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 Qsys 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 Qsys 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

\[ k_1 = -ax_{(n)} - by_{(n)}z_{(n)}, \]
\[ l_1 = -x_{(n)} + cy_{(n)}, \]
\[ m_1 = d - y_{(n)}^2 - z_{(n)}, \] (20)

where

\[ k_2 = -a\left(x\left(n + \frac{\tau}{2}\right) + \frac{\tau}{2}k_1\right) - b\left(y\left(n + \frac{\tau}{2}\right) + \frac{\tau}{2}l_1\right)\left(z\left(n + \frac{\tau}{2}\right) + \frac{\tau}{2}m_1\right), \]
\[ l_2 = -\left(x\left(n + \frac{\tau}{2}\right) + \frac{\tau}{2}k_1\right) + c\left(y\left(n + \frac{\tau}{2}\right) + \frac{\tau}{2}l_1\right), \]
\[ m_2 = d - \left(z\left(n + \frac{\tau}{2}\right) + \frac{\tau}{2}m_1\right), \] (21)

\[ k_3 = -a\left(x\left(n + \frac{\tau}{2}\right) + \frac{\tau}{2}k_2\right) - b\left(y\left(n + \frac{\tau}{2}\right) + \frac{\tau}{2}l_2\right)\left(z\left(n + \frac{\tau}{2}\right) + \frac{\tau}{2}m_2\right), \]
\[ l_3 = -\left(x\left(n + \frac{\tau}{2}\right) + \frac{\tau}{2}k_2\right) + c\left(y\left(n + \frac{\tau}{2}\right) + \frac{\tau}{2}l_2\right), \]
\[ m_3 = d - \left(z\left(n + \frac{\tau}{2}\right) + \frac{\tau}{2}m_2\right), \] (22)

\[ k_4 = -a\left(x\left(n + \tau\right) + \tau k_3\right) - b\left(y\left(n + \tau\right) + \tau l_3\right)\left(z\left(n + \tau\right) + \tau m_3\right), \]
\[ l_4 = -\left(x\left(n + \tau\right) + \tau k_3\right) + c\left(y\left(n + \tau\right) + \tau l_3\right), \]
\[ m_4 = d - \left(y\left(n + \tau\right) + \tau l_3\right)^2 - \left(z\left(n + \tau\right) + \tau m_3\right), \]

(23)

\[ x_{(n+1)} = x_{(n)} + \frac{\tau}{6}\left(k_1 + 2k_2 + 2k_3 + k_4\right), \]
\[ y_{(n+1)} = y_{(n)} + \frac{\tau}{6}\left(l_1 + 2l_2 + 2l_3 + l_4\right), \]
\[ z_{(n+1)} = z_{(n)} + \frac{\tau}{6}\left(m_1 + 2m_2 + 2m_3 + m_4\right). \] (24)

The performance of the algorithm (20)–(24) in the V4 is \( t_c = 156\,\mu s, t_{\text{id}} = 3\,\mu s, T_{\text{id}} = 159\,\mu s, \) and \( f_{\text{id}} = 6289\) 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 \( Q \tau = 62.89 \) 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 \( Q\tau = 3440.1 \) 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 \( Q\tau \) generated in 1 second considering the results obtained of \( \tau_{\text{max}} \) shown in Tables 3–5, and the results of \( T_{\text{id}} \) and \( f_{\text{id}} \) shown in Tables 7–10. We obtained the best performance of \( Q\tau \) 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 \( Q\tau \) 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
QT 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

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 $\tau_{\text{max}} = 0.057$. 
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.

<table>
<thead>
<tr>
<th>CS</th>
<th>Euler</th>
<th>Heun</th>
<th>RK4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$T_{td}$ (μs)</td>
<td>$f_{td}$ (ips)</td>
<td>$T_{td}$ (μs)</td>
</tr>
<tr>
<td>Lorenz</td>
<td>40</td>
<td>25000</td>
<td>70</td>
</tr>
<tr>
<td>Rössler</td>
<td>37</td>
<td>27027</td>
<td>53</td>
</tr>
<tr>
<td>Chen</td>
<td>40</td>
<td>25000</td>
<td>70</td>
</tr>
<tr>
<td>L. and Ch.</td>
<td>45</td>
<td>22222</td>
<td>83</td>
</tr>
<tr>
<td>MACM</td>
<td>40</td>
<td>25000</td>
<td>70</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>CS</th>
<th>$Q_T$ for V1-PIC</th>
<th>$Q_T$ for V2-dsPIC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Euler</td>
<td>Heun</td>
</tr>
<tr>
<td>Lorenz</td>
<td>11.729</td>
<td>6.9552</td>
</tr>
<tr>
<td>Rössler</td>
<td>2.9065</td>
<td>1.8115</td>
</tr>
<tr>
<td>Chen</td>
<td>0.977</td>
<td>0.5796</td>
</tr>
<tr>
<td>L. and Ch.</td>
<td>0.8064</td>
<td>0.4524</td>
</tr>
<tr>
<td>MACM</td>
<td>41.05</td>
<td>24.3432</td>
</tr>
</tbody>
</table>

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).

<table>
<thead>
<tr>
<th>CS</th>
<th>$Q_T$ for V3—PIC</th>
<th>$Q_T$ for V4—FPGA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Euler</td>
<td>Heun</td>
</tr>
<tr>
<td>Lorenz</td>
<td>2264</td>
<td>4172</td>
</tr>
<tr>
<td>Rössler</td>
<td>515</td>
<td>15280</td>
</tr>
<tr>
<td>Chen</td>
<td>189</td>
<td>1043</td>
</tr>
<tr>
<td>L. and Ch.</td>
<td>183</td>
<td>994</td>
</tr>
<tr>
<td>MACM</td>
<td>8019</td>
<td>13988</td>
</tr>
</tbody>
</table>

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.
On the other hand, a robustness diagram based on the variation of two critical parameters in the five 3D CSs in their DVs was presented in [29]. Figures 17(g)–17(i) represent the variation of the 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 σ 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, d1 = −1, and d2 = d3 = 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 used 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 τ 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 electronic/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.
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


