entropy

MDPI

Article

From Continuous-Time Chaotic Systems to Pseudo Random Number Generators: Analysis and Generalized Methodology

Luciana De Micco [1,2,3,*] and Maximiliano Antonelli [1,2,3] and Osvaldo Anibal Rosso [4]

[1] Facultad de Ingeniería, Universidad Nacional de Mar del Plata (UNMdP), Juan B. Justo 4302, Mar del Plata B7608FDQ, Argentina; maxanto@fi.mdp.edu.ar
[2] Instituto de Investigaciones Científicas y Tecnológicas en Electrónica (ICyTE), Juan B. Justo 4302, Mar del Plata B7608FDQ, Argentina
[3] Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), Rivadavia 1917, Buenos Aires C1033AAJ, Argentina
[4] Instituto de Física, Universidade Federal de Alagoas (UFAL), Maceió 57072-900, Brazil; oarosso@gmail.com
* Correspondence: ldemicco@fi.mdp.edu.ar; Tel.: +54-223-481-6600 (ext. 257)

Abstract: The use of chaotic systems in electronics, such as Pseudo-Random Number Generators (PRNGs), is very appealing. Among them, continuous-time ones are used less because, in addition to having strong temporal correlations, they require further computations to obtain the discrete solutions. Here, the time step and discretization method selection are first studied by conducting a detailed analysis of their effect on the systems' statistical and chaotic behavior. We employ an approach based on interpreting the time step as a parameter of the new "maps". From our analysis, it follows that to use them as PRNGs, two actions should be achieved (i) to keep the chaotic oscillation and (ii) to destroy the inner and temporal correlations. We then propose a simple methodology to achieve chaos-based PRNGs with good statistical characteristics and high throughput, which can be applied to any continuous-time chaotic system. We analyze the generated sequences by means of quantifiers based on information theory (permutation entropy, permutation complexity, and causal entropy × complexity plane). We show that the proposed PRNG generates sequences that successfully pass Marsaglia Diehard and NIST (National Institute of Standards and Technology) tests. Finally, we show that its hardware implementation requires very few resources.

Keywords: PRNG; statistical properties; NIST; diehard; chaos; permutation entropy; permutation complexity

Citation: De Micco, L.; Antonelli, M.; Rosso, O.A. From Continuous-Time Chaotic Systems to Pseudo Random Number Generators: Analysis and Generalized Methodology. *Entropy* **2021**, *23*, 671. https://doi.org/10.3390/e23060671

Academic Editor: Luca Faes

Received: 22 April 2021
Accepted: 24 May 2021
Published: 26 May 2021

1. Introduction

Many engineering applications require the utilization of random numbers, such as in the area of communication, encryption, codification, and modulation [1–3].

The use of chaotic systems, such as Pseudo-Random Number Generators, has recently grown because of the multiple advantages they present over stochastic algorithms [4–7]. In [8], the authors propose a five-step encryption algorithm. One of these parts is a chaotic systems module, where the system chooses between different number generators. In the conclusions, the authors highlight the importance of its randomness and present digital degradation as a subject to study. It is well known that both chaotic maps and continuous-time chaotic systems have internal correlations (they can be easily seen in 3D plots of their outputs) that prevent them from being used as PRNGs. The picture is even worse in the case of continuous-time chaotic systems that, unlike chaotic maps, present strong temporal correlations. In the literature, there are studies that promise to generate good PRNGs from continuous-time chaotic systems that can be used in cryptography and secure communications. In general, they fail in three main aspects: speed, randomness, and generality. These qualities are essential for any PRNG. A slow PRNG is useless for almost all applications, for example, for real-time encryption. To ensure security, the sequences

have to pass statistical tests, such as Diehard or NIST. Otherwise, there exist tools capable of detecting the inner correlations and thus capable of breaking the security. Finally, generality, because a good chaotic PRNG should not depend on a single chaotic system. It should work with other chaotic systems [9].

In [10,11], the authors employ a skipping technique to enhance the randomness of the chaotic outputs (called self-cascading in [11]). Instead of iterating with the original map f, it uses its d-times iterated one f^d [12]. Actually, this technique hides the correlations rather than destroying them. The higher the iteration used, the less the structures can be seen; however, the sequences still keep their internal correlations (lacks in randomness).

If the iterations are alternated between different maps, the method is called switching; this is also proposed in [11] (called hybrid-cascading there), in [13,14]. The limitation is that the maps must have common convergence domains, or at least common areas, which are not easy to find (lacks in generality).

Some works [15,16] iterate chaotic systems using floating point architecture and complex integration algorithms (such as Runge Kutta) and apply some type of post-processing or coding to eliminate the internal structures. Floating point operations, as well as complex integration algorithms, require many calculations. This, added to the post-processing calculations, limits the output speed (lacks in speed) and also requires lots of resources. Another frequently used method is to introduce external disturbances to the system. In these cases, the randomness of the final system turns out to be that of the disturbing system; unlike what may be expected, the randomness is not added, which is the case of [13,14].

A successful technique for obtaining random outputs from continuous time chaotic systems is the discarding method ([17–20], called the deep-zoom method in the latter). It basically consists of dismissing the most significant bits of each output, and it exploits the fact that chaos analytically relies on the infinitesimal depth of precision digits used. However, to maintain chaotic oscillation, they are forced to use a high number of bits (even floating point arithmetic) and complex temporal discretization methods. Furthermore, due to the internal and temporal correlations of these systems, a low number of bits for the PRNG can be taken (lacks in speed) at each iteration.

There are works that propose to generate random sequences by applying fractional calculus to existing chaotic systems or even using new defined fractional chaotic systems [21,22]. The novelty is that continuous-time systems of less than three dimensions can exhibit chaotic behavior. However, the typical internal structures of chaotic systems remain in the sequences generated by these systems. So, they are in the same situation as mentioned before. What is more, the procedure of generalizing the integer derivative and integral orders to real and complex numbers requires more calculations, and it is not clear if having fewer dimensions is an advantage; for example, in our approach, we take advantage of the three dimensions as we extract bits from the three variables, so to increase throughput.

Traditionally, continuous-time chaotic systems have not been the preferred choice over chaotic maps mainly due to the strong time correlation and the extra computations they require to perform the time-discretization.

There exist an ample variety of numerical algorithms to solve ordinary differential equations. Which of them to choose will depend on the final objective. Here, rather than generating what some researchers call the "true solutions", the interest is to obtain the most random output while keeping the chaotic behavior. Meanwhile, it is desired to employ the least amount of resources in terms of hardware, so the simplest method would be preferable. The drawback is that, in general, the simplest numerical methods produce the trajectories to converge or tend to cycles with short periods.

Here, we analyze numerical integration algorithms looking for the one that (i) maximizes the randomness degree, and (ii) requires the fewest resources regarding its hardware implementation.

The time correlation is related to the integration step, Δt. The lower Δt, the more time-correlated the output would be. However, a large Δt could result in the system losing

its chaotic behavior. Therefore, choosing the appropriate step Δt is not a trivial task. We propose a point of view where the continuous system becomes a discrete map, and the time step used is seen as an extra parameter of this new map. This enables us to characterize the system in terms of Δt and use statistical quantifiers as well as nonlinear tools to describe the dynamics' evolution of the maps.

Our goal is to propose an extremely simple modification applied to the digitalized continuous-time chaotic system that keeps it oscillating and, at the same time, breaks the internal structures and the time correlation of the outputs, which allows us to apply the discarding method, but discarding a minimum number of bits, so as not to lose speed. Using the standard Marsaglia's Diehard and NIST tests, we show that the resulting map can generate high-quality random numbers to ensure security. Finally, our method is general as it can be applied to any continuous-time chaotic system. We also present the resources needed by a hardware implementation in an FPGA board of the proposed PRNG and compare them with the ones of the original map, showing that the circuit complexity remains almost the same.

The rest of the paper is organized as follows. Section 2 presents the ordinary differential equations that concern us, briefly describes the methods that we consider for the time discretization of those systems and presents the proposed modification over the Euler method. Section 3 gives a short review of the quantifiers employed to characterize the maps' chaoticity and randomness. In Section 4, the obtained results when applying the proposed methodology to the Rössler system are presented. There, we develop new maps that emerge from applying the numerical methods and the proposed modification. Finally, in Section 5, we draw some concluding remarks.

2. Continuous-Time Chaotic Systems

The general form of a typical continuous-time chaotic system is as follows:

$$\dot{u} = f(t, u) \tag{1}$$

where $f(t, u)$ are nonlinear functions of time and the states variables u. Given an initial value $u(t_0) = u_0$, these systems have a determined evolution. However, there is no general formula to solve this kind of equation; in fact, most of these first-order differential equations cannot be analytically solved [23]. That is why particular time-dependent solutions are most often sought with numerical means. Thus, multiple techniques that approximate the output of the system have emerged. Among the ample range of possibilities for making such a job, the choice of one of them depends on several factors. When an exact reproduction of the continuous system dynamics is required, powerful numerical methods that involve pre-iterations and variable time-steps are mandatory. However, when using these systems, such as PRNGs, the criteria change for choosing which method to use changes. It switches to the ones that allow the output to meet the required properties along with the strong restrictions regarding the hardware implementation, i.e., minimize the required resources and latency, maximize throughput and operation frequency. Considering the above, we focus on fixed integration step methods. In this context, we evaluate the following three well-known numerical methods for the time-digitization of continuous-time systems.

2.1. Fourth Order Runge–Kutta Method (RK4)

The main idea of this method is the precalculation of stages at various points using samples of f to obtain the next step [24].

$$u_{t+\Delta t} = u_t + \frac{\Delta t}{6}(k_1 + 2k_2 + 2k_3 + k_4), \tag{2}$$

$$\begin{cases} k_1 = \Delta t f(t, u_t), \\ k_2 = \Delta t f\left(t + \frac{\Delta t}{2}, u_t + \frac{k_1}{2}\right), \\ k_3 = \Delta t f\left(t + \frac{\Delta t}{2}, u_t + \frac{k_2}{2}\right), \\ k_4 = \Delta t f(t + \Delta t, u_t + k_3), \end{cases}$$

where u_t is the discrete-time state variable and Δt is the time step size.

2.2. Heun's Method (HUN)

Heun's method considers the tangent lines to the solution curve at both ends of the interval. This method requires two stages of calculation as follows:

$$\begin{aligned} u_{t+\Delta t} &= u_t + \frac{\Delta t}{2}[f(t, u_t) + f(t + \Delta t, \tilde{u_t})], \\ \tilde{u_t} &= u_t + \Delta t f(t, u_t) \end{aligned} \tag{3}$$

2.3. Euler's Method (EUR)

Among all numerical procedures for solving ordinary differential equations with a given initial value, the simplest one is Euler's method in which differentials are approximated by a trapezoid with base Δt, as Equation (4) shows. Euler's method is a one-step algorithm; that is, in order to calculate the variables at the time $t + \Delta t$, it is only necessary to know the values at the previous instant.

$$\dot{u_t} \approx \frac{u_{t+\Delta t} - u_t}{\Delta t}, \tag{4}$$

where:

$$u_{t+\Delta t} = u_t + \Delta t f(u_t) \tag{5}$$

2.4. Modified Euler Proposed Method (EUR_MOD)

Choosing a large Δt would be believed to help de-correlate the output of the system and thereby improve its randomness. However, the largest Δt before the system loses its chaotic behavior is not big enough to break its temporal structures. There are numerous proposals to increase the randomness of chaotic systems [25–27]. In some of them, post-processing the outputs is proposed; however, this idea adds hardware and increases latency. Other works propose to disturb the system with external noise, to switch between one or more chaotic maps [28]. Then, complexity is added to the resulting circuit, but the achieved randomness of the final system is just that of the disturbance. As said, the objective is to destroy the temporal correlations while keeping the chaotic oscillation of the system. Furthermore, it is desired to minimize the hardware resources and increase throughput and speed. Our idea is to combine the time-digitalization process with the randomization one. Thus, we have selected Euler's method as is the simplest one and thereby will require the least amount of hardware to be implemented. Based on Equation (5), with the idea of breaking the temporal structure, we apply the following modification:

$$\begin{cases} x_{t+\Delta t} = x_t + \Delta t f_x(x_t, y_t, z_t) + p_1 z_t (-1)^{x_t \ mod \ 2} \\ y_{t+\Delta t} = y_t + \Delta t f_y(x_t, y_t, z_t) + p_2 x_t (-1)^{y_t \ mod \ 2} \\ z_{t+\Delta t} = z_t + \Delta t f_z(x_t, y_t, z_t) + p_3 y_t (-1)^{z_t \ mod \ 2} \end{cases} \tag{6}$$

where $u_t \ mod \ 2$ returns the remainder of a division after u_t is divided by 2. It returns 1 if u_t is odd, or 0 if it is even. The parameters p_1, p_2 and $p_3 \in [0, 1]$, so for the particular case where $p_1 = p_2 = p_3 = 0$, the map converges to ROS_{EUR} map.

The modification consists of incorporating one extra term into each function. This term is simply another state variable multiplied by 0.5. That term will be added or subtracted from the function depending on the parity of the current state variable.

3. Quantifiers

The time-digitization of continuous systems that turns them into maps generates changes in their dynamics. Chaoticity, stochasticity, and mixing properties change, so the following tools are used to analyze them.

3.1. Maximum Lyapunov Exponent

A chaotic orbit (chaotic attractor) is aperiodic, meaning that it never repeats exactly itself, and the oscillation persists for a time tending to infinity. The attractor's movements exhibit sensitive dependence on the initial conditions. This means that two trajectories that start very close, quickly diverge; thus, they will have very different futures. The practical implication of this is that long-term prediction becomes impossible, as small uncertainties are rapidly amplified. The separation $\delta(t)$ between two trajectories of the same system that initially differ δ_0 evolves exponentially in the way of Equation (7):

$$||\delta(t)|| \sim ||\delta_0||e^{\lambda t} \tag{7}$$

Therefore, neighboring trajectories separate exponentially fast. The number λ is called the Lyapunov exponent. When this exponent is positive, it is said that the system has a time horizon beyond which the prediction fails at tolerance a. Actually, λ depends on the trajectory that is being studied. Therefore, it must be averaged over many points of the same trajectory to estimate its true value. In addition, each system has as many Lyapunov exponents as dimensions. The largest of them, known as the maximum Lyapunov exponent (MLE), is of special significance since a positive value indicates the possible existence of chaos [29,30]. Nevertheless, this is a necessary but not sufficient condition of chaoticity since a divergent system can have positive MLE. Therefore, for a system to be chaotic, in addition to having some positive Lyapunov exponent, it must have a bounded non-divergent trajectory in the phase plane.

3.2. Bifurcation Diagram

A bifurcation diagram allows studying the changes in the qualitative or topological structure of the trajectories of a dynamical system. It shows the visited values of a system as a function of a certain parameter. It allows differentiating areas of the parameter in which the system behaves like fixed points, periodic orbits, or chaotic attractors [31]. We can say that bifurcation occurs in a dynamical system when a small smooth change of a parameter causes a sudden 'qualitative' or topological change in the dynamical system's behavior.

3.3. Probability Density Function (PDF)

The randomness quantifiers used here are functional of the PDF P associated with the data sequence under analysis. The determination of a PDF can be done using several different methods [32], and their applicability depends on particular characteristics of the data, such as stationarity, time series length, parameter variation, and level of noise contamination. The PDF and the sample space are inextricably linked so it is a nontrivial problem to obtain the optimal PDF to extract the desired information. Here, we have employed the Bandt and Pompe approach, and this PDF is able to satisfactorily show the temporal correlations of higher orders [33,34]. The delay method has been used to extract time causal information from the sequences. A delayed reconstruction in D dimensions is formed by the vectors x_n given as:

$$\mathrm{x}_n = (x_{n-(D-1)v}, x_{n-(D-2)v}, \ldots, x_{n-v}, x_n) \tag{8}$$

The lag or delay time v is the time difference in the number of samples (or in time units $\tau = v\Delta t$) between adjacent components of the delay vectors. A good estimate of the lag time is very difficult to obtain. If τ is small compared to the internal time scales of the system, successive elements of the delay vectors are strongly correlated, whereas if τ is very large, successive elements are already almost independent. Among the existing

proposals, we have adopted the first zero of the autocorrelation function of the signal as the τ value [30]. This algorithm to extract the Bandt–Pompe PDF has been widely addressed and described by previous works [35].

3.3.1. Normalized Shannon Entropy

The well-known normalized Shannon entropy denotes the amount of "disorder" a system presents. It has been shown to be able to successfully characterize determinism and stochasticity of generated sequences [32]. This information theory quantifier is a functional of the probability density function and is defined by the normalized Shannon expression (Equation (9)):

$$H[P] = -\frac{\sum_{i=1}^{N} p_i \ln(p_i)}{\ln(N)}; \quad (9)$$

where N is the number of elements of the alphabet. We denote permutation entropy (H_{BP}) as the result of applying the normalized entropy to the PDF proposed by Band and Pompe, which quantifies the causality of the symbolic series discarding amplitude information.

3.3.2. Statistical Complexity Measure

A statistical complexity measure, denoted by C, is a general indicator of structure or correlation. This measure vanishes in the extreme ordered and disordered limits ("boundary conditions"). During the last decade and a half, different measures of statistical complexity have been proposed [36]. Here, we have adopted the functional form introduced by López Ruiz et al. [37] with the modifications advanced by Lamberti et al. [38], given by Equation (10), [39].

$$C[P] = Q_j[P, P_e]H[P], \quad (10)$$

where P_e is the uniform distribution, and Q_j is the so-called "disequilibrium", defined in terms of the extensive Jensen–Shannon divergence, which in turn induces a squared metric, [39] (Equation (11)).

$$Q_j[P, Pe] = Q_0\left\{S\left[\frac{(P, P_e)}{2}\right] - \frac{S[P]}{2} - \frac{S[P_e]}{2}\right\}, \quad (11)$$

where Q_0 is the normalization constant, Equation (12), and is obtained when the system is deterministic; that is, only one component of P is equal to one, and the remaining components are equal to zero:

$$Q_0 = -2\left[\frac{(N+1)}{N}\ln(N+1) - \ln(2N) + \ln(N)\right]^{-1}. \quad (12)$$

This quantifier detects internal structures from the symbol source when it is applied to the Bandt and Pompe PDF; thus, we denoted permutation complexity C_{BP} as the resulting quantity.

The juxtaposition in a two-dimensional graph of the quantifiers H_{BP} and C_{BP} has demonstrated to be particularly efficient to reveal properties of the underlying processes from some measurable or observable quantity, called causal Eentropy $\times$ complexity plane [40]. High values of C_{BP} correspond to time series with immersed structures, which occurs with chaotic series. On the other hand, the point $C_{BP} = 0$ and $H_{BP} = 1$ are that of a sequence with no internal correlations. There are many relevant applications of the $H_{BP} \times C_{BP}$ plane; for example, in [34], Rosso et al. use this plane to discriminate between stochastic and chaotic series, in [41], the authors employ it as a tool for distinguishing songs, and Zunino and Ribeiro utilize it to discriminate image textures [42], just to mention a few.

3.4. Statistical Randomness Tests

For a sequence to be suitable to be used as PRNG, it is necessary to successfully pass statistical tests. Here, we employed NIST Statistical Test Suite and Marsaglia Diehard tests.

3.4.1. NIST Statistical Test Suite

The NIST SP 800-22 test suite [43] consists of 15 statistical randomness tests that are applied to binary data stream files. It requires the size of each sequence length to be of the order 10^3 to 10^7. For each test, it yields p-values, and it also checks the proportion of passing sequences and the uniform distribution of the p-values.

3.4.2. Marsaglia Diehard Tests

The 15 statistical tests that make up the Diehard battery should be applied independently over files of several million 32-bit integers. Their output is a statistical p-value. To evidence randomness, each test output should be uniformly distributed between 0 and 1. The tests should be repeated multiple times with different integer sets to demonstrate the robustness of outcomes.

4. Results

To show the proposed method, the well-known Rössler system is used here. This continuous-time chaotic system is defined by the following set of coupled ordinary differential Equations [44]:

$$\begin{cases} \dot{x} = -y - z, \\ \dot{y} = x + ay, \\ \dot{z} = b + z(x - c). \end{cases} \tag{13}$$

Applying the digitalizing methods mentioned in Section 2, the following maps, which include Δt as a new parameter, are obtained:

- ROS_{RK4} map, Rössler system digitalized by the 4th order Runge Kutta method.
- ROS_{HUN} map, Rössler system digitalized by the Heun method.
- ROS_{EUR} map, Rössler system digitalized by the Euler method.
- ROS_{EUR_MOD} map, Rössler system digitalized by our proposed Euler modified method.

We have employed parameters $a = 0.2$, $b = 0.2$, and $c = 5.7$ that assure chaotic behavior of the continuos-time Rössler system. In the case of the ROS_{EUR_MOD} map, the parameters $p_1 = p_2 = p_3 = 0.4$ were used unless specified otherwise. Since our objective is to utilize the systems as PRNGs, based on subsequent experiments, we have followed three main steps:

1. First, we analyzed the chaotic behavior when the systems are digitalized in time; focusing on the impact on the dynamic of each discretization method and its dependence on Δt (Section 4.1). Therefore, we calculate the MLE [29] and bifurcation diagrams of the emerged maps. Note that at this point, we do not consider amplitude discretization of the systems. Therefore, we employ a floating-point arithmetic (IEEE 754 double-precision standard) for the calculations.
2. The second step deals with the amplitude digitization effect (Section 4.2). Then, we analyze the statistical properties, focusing on achieving the highest randomness.
3. Finally, we present the hardware implementation of the obtained PRNG that is based on the proposed modification to the system digitalized in time by Euler's method and iterated using signed fixed-point architecture. We also show the resources needed to implement it in an FPGA board (Section 4.3).

4.1. Time Digitization Analysis

In all cases, the quantifiers are averaged over 100 surrogates starting at different initial conditions, a transitory of 8×10^6 is first deleted, and the maps are then iterated 10^6 times. The minimum Δt is iterated 10^6 times, and for higher Δt, the iterations are decreased so as to cover the same attractor window time. In order to understand the maps' behaviors, Figure 1 shows the $3D$ phase space for some Δt values of the ROS_{HUN} map. There, it can be seen how attractors change and evolve. It is clearly shown that even though the

continuous-time system attractor is blurred by the increase of Δt, new attractors appear, and these attractors may be even more chaotic than those of the continuous-time systems (depicted by their MLE value). As expected, smaller values of Δt reproduce an attractor closer to that of the continuous-time system; however, in many cases, they converge to short cycles or fixed points, losing their chaotic behavior.

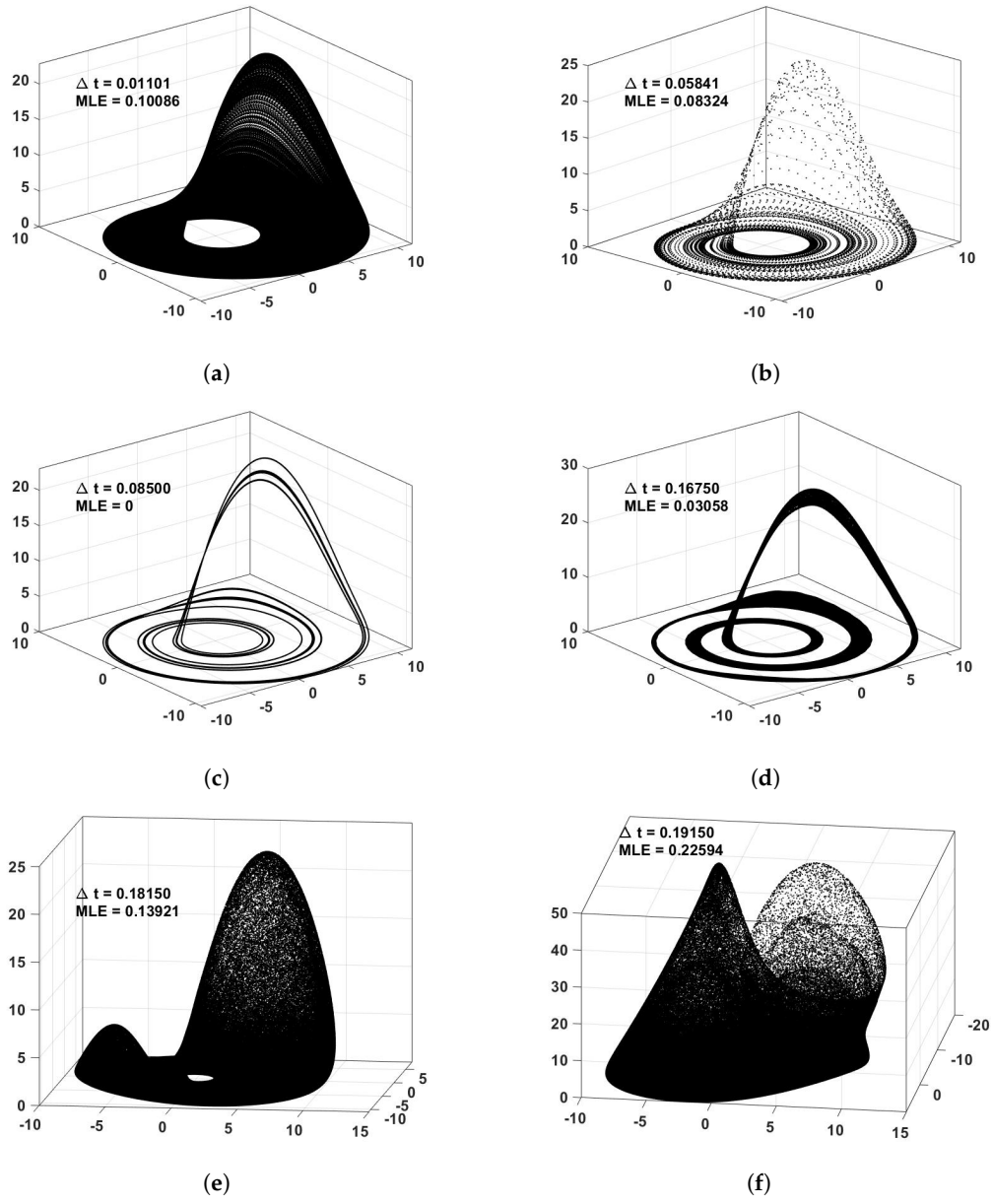


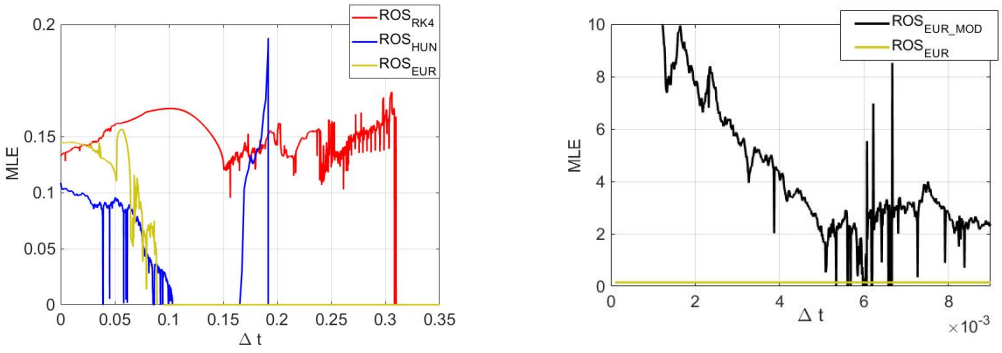
Figure 1. Attractors of the ROS_{HUN} map for different values of Δt and their MLE value (**a**–**f**).

4.1.1. Topological Analysis

We have used Sprott's method to calculate the MLE of the maps using Δt as a parameter, [45]. Figure 2a shows how the MLE varies with Δt in the case of the time-digitalizing Rössler system using the three methods. Each resulting map presents a different behavior regarding the existence of chaoticity with Δt as a parameter. As it may be supposed, the ROS_{RK4} map (red line) seems to preserve the chaotic behavior for larger values of Δt, while the ROS_{EUR} and ROS_{HUN} maps (yellow and blue lines, respectively) behave similarly. In the case of the ROS_{HUN} map, it presents an isolated range of Δt between $\sim$0.17 and $\sim$0.19, where the system behaves chaotically, and it also presents isolated low values of MLE for some time steps indicating low or no chaoticity. The ROS_{EUR} map is the first one that

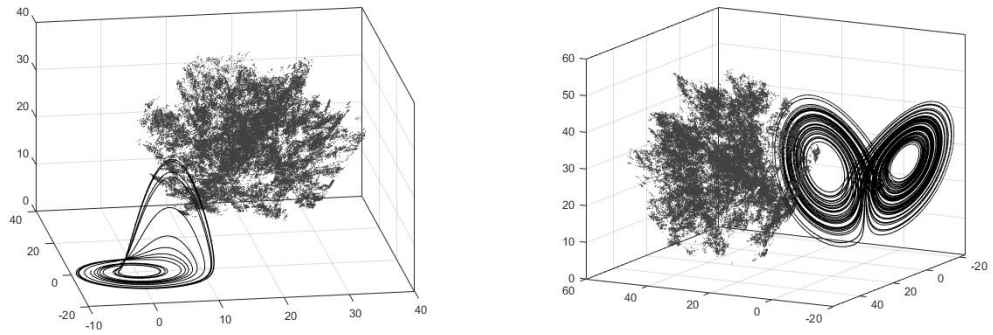
losses its chaotic behavior for higher time steps. It is interesting to note that even though all three maps are derived from the same system for small values of Δt, the MLE does not show similar values. The reason may be the accumulated arithmetic errors that prevent following the continuous-time attractors. In addition, it can be seen that there are some cases where larger time steps present higher values of MLE.

In Figure 2b, it can be seen that the proposed modification increases the chaotic behavior of the system. The ROS_{EUR_MOD} map presents higher values of MLE than the ROS_{EUR} map. To show the generality of the proposed method, Figure 3 shows the $3D$ phase of Rössler and Lorenz systems digitalized by Euler's method (ROS_{EUR} and LOR_{EUR}) in black, and their modified maps (ROS_{EUR_MOD} and LOR_{EUR_MOD}) in gray. It can be seen how the proposed modification breaks the inner structures and temporal correlations of the sequences and keeps the chaotic behavior. This enables the retainment of more bits for the PRNG output and, in this way, increases the throughput.



(**a**) (**b**)

Figure 2. MLE with Δt as a parameter for the Rössler system using different time-discretization algorithms and the proposed method, with floating-point arithmetic. (**a**) MLE for ROS_{RK4}, ROS_{HUN}, and ROS_{EUR} maps. (**b**) MLE for ROS_{EUR} and ROS_{EUR_MOD} maps.



(**a**) (**b**)

Figure 3. Phase spaces of two well-known continuous-time chaotic systems digitalized by the Euler method and their modified versions. In black are the classical systems, and in gray are the modified versions. It can be seen that, by using the proposed method, the original attractors have been broken and the spaces are more uniformly filled. (**a**) Attractors of ROS_{EUR} and ROS_{EUR_MOD} maps. (**b**) Attractors of LOR_{EUR} and LOR_{EUR_MOD} maps.

Regarding the bifurcation diagram, we have built the diagrams using Poincaré maps, which is the intersection with a certain surface. Then, the bifurcation diagrams show all the visited values by the systems [31]. Figure 4 shows the bifurcation diagram of the ROS_{HUN} map superimposed with the MLE (red line). It can be seen how the MLE is able to effectively predict the chaoticity of the map. Within the chaotic region, some isolated gaps that correspond to low chaoticity can be seen. These gaps match with low values of

the MLE. It can be seen that from $\Delta t \sim 0.105$, it completely loses its chaoticity and it stays on a periodic cycle until $\Delta t \sim 0.167$. The darker areas of the chaotic region imply that the system, while being in the state of chaos, spends more time there than in the lightly shaded regions. The most interesting places inside that region are the "white spaces", which have an important role in the transition to chaos. The "white regions" and their boundaries also show the instability of the initial conditions, another important aspect of the chaos.

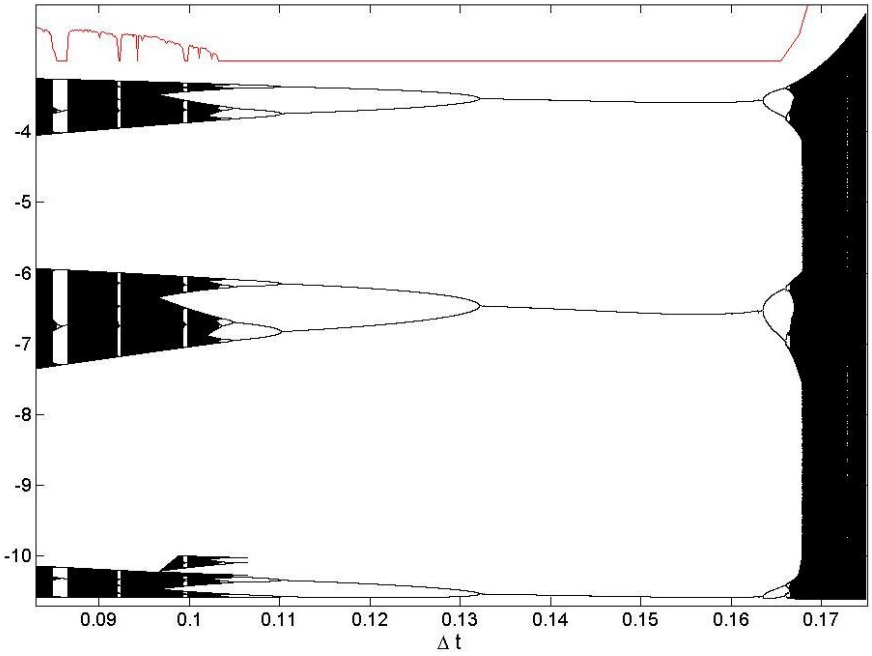


Figure 4. The ROS_{HUN} bifurcation diagram with Δt as a parameter, for the z_t variable with the plane $x_t = 0$ and the MLE superimposed (red line). Note that the y-axis values correspond to the plane, not the MLE.

4.1.2. Statistical Analysis

Up to this point, we have only analyzed the chaoticity of the maps; however, we do not have information about the randomness that their outputs present. The applications in which these maps are intended to be used require that their sequences, in addition to being chaotic, have no internal structures and all their possible outputs appear in a balanced way. To evaluate this, we calculate the randomness quantifiers described in Section 3. Each quantifier has been averaged over 100 files. Every surrogate file starts with a different initial condition, a transitory of 8×10^6 iterations was first deleted, and the maps were then iterated 10^6 times.

To extract causal informati on by calculating H_{BP} from these observations, we employ here x state variable, $v = 1$, and $D = 6$.

Figure 5 shows the plane $H_{BP} \times C_{BP}$ for the Rössler system time-digitalized with the three mentioned methods using different values of Δt. The continuous curves correspond to the boundaries of values for the statistical complexity, as a function of the value of the normalized Shannon entropy [38]. It can be seen that in the cases where the system is unmodified (ROS_{RK4}, ROS_{HUE}, and ROS_{EUR} maps), the quantifiers remain in the same area, that is, strong correlations and poor balance of values. When the proposed modification is applied, the quantifiers move towards the area of chaotic maps. The output of the system slightly improves the balance of its values and also increases its inner correlations.

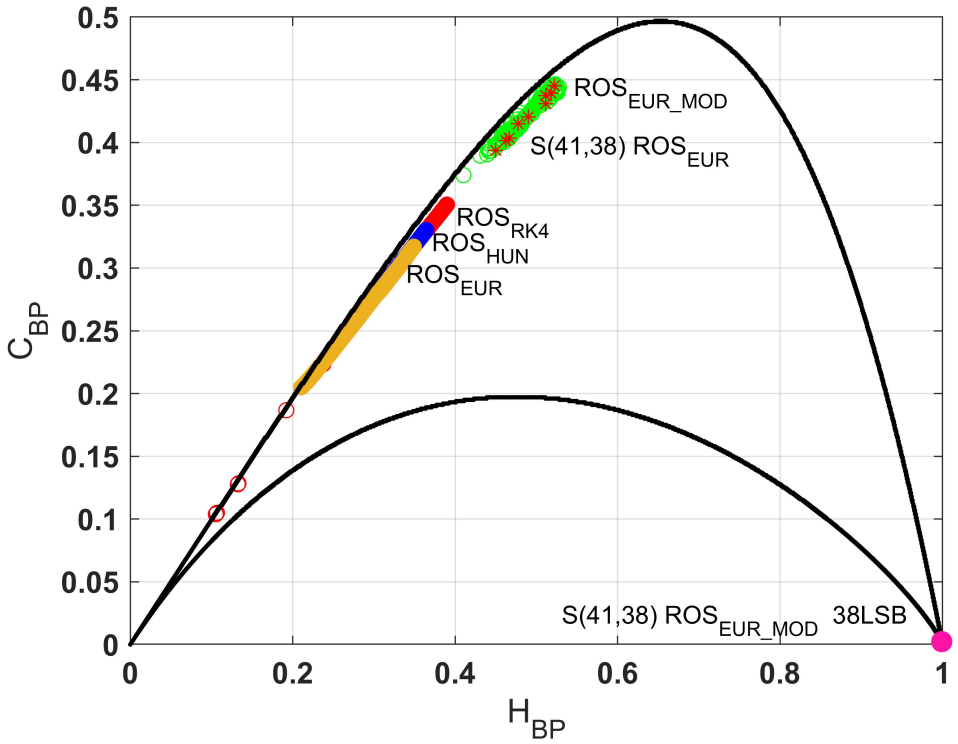


Figure 5. Causal entropy × complexity plane, Rössler system using the three methods of time-discretization for $0.0001 \leq \Delta t$ to the higher Δt that could be reached before the maps diverge. Red points are the ROS_{RK4} map, blue points are the ROS_{HUE} map, and yellow points are the ROS_{EU} map. The ROS_{EUR} and ROS_{EUR_MOD} maps using S(41,38) are the green points and black stars, respectively. Finally, our proposed PRNG (ROS_{EUR_MOD} map S(41,38) considering the 38 least significant bits) are the pink points, and these are the best sequences in terms of randomness (closest to the ideal point $H_{BP} = 1, C_{BP} = 0$).

4.2. Amplitude Digitization Analysis

We iterate the maps using signed fixed-point architecture for analyzing the effect of amplitude digitalization [46]. As said, our goal is to develop a hardware-implemented PRNG, which is why we have selected fixed-point architecture and Euler's discretization method because of the simplicity they mean in terms of hardware design (Equations (14) and (15)).

- ROS_{EUR} map:

$$\begin{cases} x_{t+\Delta t} = x_t + \Delta t(-y_t - z_t) \\ y_{t+\Delta t} = y_t + \Delta t(x_t + a y_t) \\ z_{t+\Delta t} = z_t + \Delta t[b + z_t(x_t - c)] \end{cases} \tag{14}$$

- ROS_{EUR_MOD} map:

$$\begin{cases} x_{t+\Delta t} = x_t + \Delta t(-y_t - z_t) + \frac{z_t}{2}(-1)^{x_t \ mod \ 2} \\ y_{t+\Delta t} = y_t + \Delta t(x_t + a y_t) + \frac{x_t}{2}(-1)^{y_t \ mod \ 2} \\ z_{t+\Delta t} = z_t + \Delta t[b + z_t(x_t - c)] + \frac{y_t}{2}(-1)^{z_t \ mod \ 2} \end{cases} \tag{15}$$

We have employed words of *wl* bits, with *fl* bits to represent the fractional part in two's complement arithmetic; this architecture is represented by S(*wl*,*fl*). Equation (16) outlines the data format used for each state variable.

$$u_t = \overbrace{\underbrace{b_{wl-1}b_{wl-2}\ldots b_{wl-(wl-fl)}}_{integer\ part} \cdot \underbrace{b_{fl-1}b_{fl-2}\ldots b_0}_{fractional\ part\ (fl\ bits)}}^{signed\ word\ (wl\ bits)} \quad (16)$$

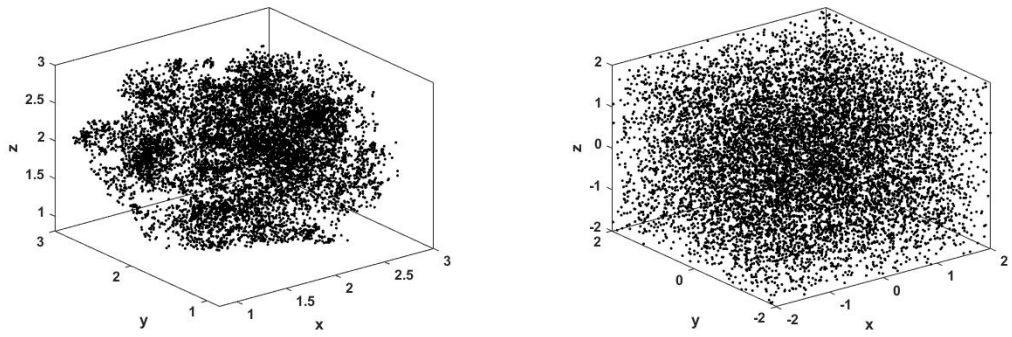
fractional point

4.2.1. Topological Analysis

Figure 6a,b show the 3D phase space of the ROS_{EUR} and ROS_{EUR_MOD} maps, iterated using fixed-point architecture. There, we used $\Delta t = 0.001$. Figure 6a shows the phase space of ROS_{EUR_MOD} map using S(41,38). It can be seen that the phase space does not present significant changes compared to that of the attractor iterated with floating-point arithmetic (Figure 3b). Therefore, the outputs of the proposed PRNG are the k_{LSB} least significant bits of each state variable u_t (see Equation (17)). This is a commonly used procedure in PRNGs [10,47].

$$u_t = b_{wl-1}b_{wl-2}\ldots \underbrace{b_{K_{LSB}-1}b_{K_{LSB}-2}\ldots b_0}_{PRNG\ (K_{LSB}\ bits)} \quad (17)$$

It is desired to find the minimum wl that keeps oscillating the attractor in a non-periodic way, and the largest k_{LSB} that produces the output sequences to pass the Marsaglia and NIST tests. In Figure 6b, it can be seen how the obtained sequence does not present any structure and all the space is equally filled.



(**a**) (**b**)

Figure 6. Phase space of the ROS_{EUR_MOD} system iterated with S(41,38) arithmetic. (**a**) Considering all wl bits. (**b**) Considering the $k_{LSB} = 38$ bits.

4.2.2. Statistical Analysis

Returning to Figure 5 where the causal entropy $\times$ complexity plane was shown, the red stars correspond to the ROS_{EUR_MOD} map using a signed fixed-point architecture with 41 bits, of which 38 are used to represent the fractional part (S(41,38)). It can be seen that the utilization of fixed-point arithmetic does not influence the statistical properties of the proposed system. Finally, when the most significant bits are discarded (pink point), it can be seen that the sequences reach the ideal point in terms of randomness, $H_{BP} = 1$ and $C_{BP} = 0$. Table 1 shows the results obtained when applying the Marsaglia battery of statistical tests to the original system (ROS_{EUR} map) and the modified one (ROS_{EUR_MOD} map). It can be seen that the ROS_{EUR} map needs more bits to pass the tests. There, it is demonstrated that the proposed method keeps the system oscillating and enables it to discard fewer bits of each output. This is due to the fact that the time correlations and internal structures are destroyed. To confirm our proposal's usefulness, we keep k_{LSB} bits of the output of both the original and the modified maps, iterated using S(wl,fl) arithmetic, and test them using Diehard tests.

Table 1 shows that the sequences generated by ROS_{EUR_MOD} pass Marsaglia tests using fewer word bits (wl = 41) and allows to keep more bits per iteration for the PRNG (higher k_{LSB}) than ROS_{EUR} map (wl = 56). Table 2 shows the results of testing the proposed PRNG via the NIST SP 800-22 test suite. In agreement with the values used in the literature [13,47–49], 1000 sequences of length 10^6 bits each have been tested. For a significance level of 0.01 (α = 0.01) and 1000 samples, the minimum pass rate for each statistical test is approximately 980, with the exception of the random excursion (variant) test where it is approximately 597 for a sample size of 611 binary sequences. The proposed PRNG passes all these tests.

Table 1. Results from the Marsaglia Diehard test for ROS_{EUR} and ROS_{EUR_MOD} maps for different precision using S(wl,fl) architecture and considering the 38 least significant bits ($k_{LSB} = 38$) and Δt = 0.001.

wl	fl	ROS_{EUR}	ROS_{EUR_MOD}
40	36	fail	fail
40	38	fail	fail
41	38	fail	success
42	38	fail	success
50	45	fail	success
51	45	fail	success
52	45	fail	success
53	45	fail	success
54	45	fail	success
55	50	fail	success
56	50	success	success

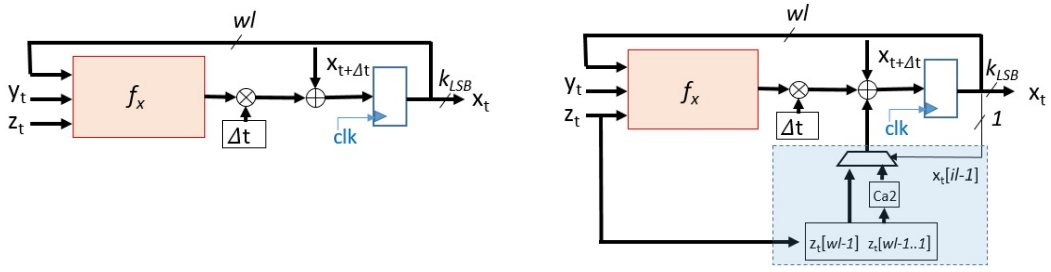
Table 2. Results from the SP 800-22 test for the ROS_{EUR_MOD} map using S(*41*,*38*) architecture and dismissing the 3 most significant bits ($k_{LSB} = 38$).

Statistical Test	p**_Value**	**Proportion**	**Result**
Frequency	0.060875	980/1000	success
BlockFrequency	0.000163	984/1000	success
CumulativeSums	0.008753	981/1000	success
Runs	0.002993	987/1000	success
LongestRun	0.141256	988/1000	success
Rank	0.961869	986/1000	success
FFT	0.424453	990/1000	success
NonOverlappingTemplate	0.697257	989/1000	success
OverlappingTemplate	0.319084	984/1000	success
Universal	0.116065	990/1000	success
ApproximateEntropy	0.894918	991/1000	success
RandomExcursions	0.330947	603/611	success
RandomExcursionsVariant	0.401777	599/611	success
Serial	0.205531	986/1000	success
LinearComplexity	0.971006	988/1000	success

4.3. Hardware Implementation

Here we show that the proposed modification is extremely simple to implement and assures the required randomness. Figure 7 shows a schematic of a hardware implementation of ROS_{EUR} and ROS_{EUR_MOD} maps. In the latter map, the parameters used were $p_1 = p_2 = p_3 = 0.5$.

Figure 7a shows a schematic of the recursive function for x of a general map obtained by the Euler method applied to a continuous-time chaotic system (recursive functions for y and z are analog, and for simplicity are not shown). There, the f_x block receives the three state variables at time t and calculates the next output at time $t+1$. In the case of the Rössler system studied here, this term is $f_x = -y_t - z_t$. Its output is multiplied by Δt and added to x_t in order to generate $x_{t+\Delta t}$. This value is then latched by a register at each clock cycle. Figure 7b shows the proposed modified circuit. It can be seen that it consists of just one extra term. This term makes the product of one state variable (in this case z_t) by 0.5. However, for its implementation, no multiplier is required since this term is a right-shifted version of the state variable. Then the least significant bit of the integer part of x_t is fed back to select if the new term is added or subtracted. The whole term can be implemented by a positive or negative right-shifted version of z_t, as it can be seen in the light blue square of Figure 7b. The figure shows that it requires very few resources.



(**a**) (**b**)

Figure 7. Block diagram of hardware implementation of the maps, in this case, $f_x = -y_t - z_t$. (**a**) ROS_{EUR} map. (**b**) ROS_{EUR_MOD} map.

A comparison of the implementation results of the proposed PRNG with three other PRNGs can be seen in Table 3. ROS_{EUR} refers to a PRNG based on the ROS_{EUR} map, which basically consists of implementing the map and then applying the discarding method. The map requires a word length of 56 bits to enable the extraction of 38 bits (of each variable) on each iteration. This word length is the minimum number of bits for ROS_{EUR} to pass the Marsaglia test (see Table 1). We can see that this PRNG requires more resources and that the maximum frequency and throughput are lower than that of the proposed PRNG, which is due to the need to use a larger word size to ensure the randomness of the output. In the third column, the resources used by an implementation of the well-known PRNG Mersenne Twister (MT19937) implemented in an FPGA are shown [50]. We can see that the resource requirement is slightly higher than that of the proposed PRNG. The maximum operating frequency is lower and the achieved throughput is lower as well. The last column corresponds to a continuos-time, chaotic-based PRNG that employs a linear feedback shift register to obtain the transition between Lorenz-like and Chen-like behaviors. Then, authors keep the eight least significant bits and xor them (the discarding method) [13]. Although this generator does not comply with being generic (it only works for the Lü-like chaotic system because it is capable of exhibiting both Lorenz-like and Chen-like chaotic system behaviors for different parameter values), we include it in order to compare its performance.

Table 3. Summary of resources for ROS_{EUR} and ROS_{EUR_MOD} maps using S(41,38) arithmetic.

Resources	ROS_{EUR_MOD}	ROS_{EUR}	**MT19937 [50]**	**Chaotic-Based [13]**
Platform	Xilinx Zynq-7000	Xilinx Zynq-7000	Xilinx XCV2000E	Altera EP3C16F484C6
LUT	508	604	539	1826
FF	123	200	660	1826
DSP	20	34	0	0
16-Kbit BRAM	0	0	2	0
f_{max} [MHz]	50	40	24.234	30.98
Throughput [bits/sec]	5.7×10^9	4.5×10^9	24.16×10^6	247×10^6

5. Conclusions

In this paper, we showed that the digitalization method and the time step have a significant influence on digitalized systems' dynamics and, therefore, on the sequences generated by them. The chaotic behavior and statistical degree of the sequences were analyzed using tools from nonlinear systems analysis and information theory quantifiers. In that way and with the objective of using these systems as PRNGs, we proposed a modification to the map generated by Euler's method that destroys the time correlations of the output and keeps the chaotic oscillation. We have also analyzed the randomness behavior with the amplitude discretization using different precision and data widths. The $H_{BP} \times C_{BP}$ plane shows that the three methods of digitalization analyzed produce outputs in almost the same area, poor balance of amplitudes, and strong inner correlations. The proposed method produces both floating and fixed-point architectures moving towards the typical area of the chaotic maps.

Our goal was to demonstrate that our proposed modification to the digitalized Euler system generates the most random output, which is located in the optimum point of $H_{BP} \times C_{BP}$ plane (uncorrelated noise). The proposed modification achieves the lower value of wl and higher value of k_{LSB} when passing the Marsaglia Diehard and NIST tests. Our method is general and can be applied to any continuous-time chaotic system.

Regarding portability and reproducibility, which are important PRNG properties, the proposed schematic defines the architecture and precision of the variables and the internal calculations since it is a hardware implementation. Then, identical results will be obtained in any programmable device, and the repeatability of the results will be ensured.

Further study on the basin of attraction of the digitized system would be required to define the set of available seeds for the PRNG.

Finally, we compared the resources needed to implement our and other existing methods to obtain PRNGs. We showed that the proposed PRNG is superior in terms of resources, maximum frequency of operation, and throughput.

Author Contributions: Conceptualization, L.D.M. and M.A.; investigation, L.D.M.; methodology, L.D.M. and M.A.; writing—original draft, L.D.M.; writing—review and editing, L.D.M., M.A. and O.A.R. All authors have read and agreed to the published version of the manuscript

Funding: This research was funded by the Consejo Nacional de Investigaciones Científicas y Técnicas (PIP11220170100553CO), Agencia Nacional de Promoción Científica y Tecnológica (PICT2019-2019-03024), Faculty of Engineering of the National University of Mar del Plata (FI-UNMDP), and by Abdus Salam International Centre for Theoretical Physics (ICTP) Associateship Scheme.

Data Availability Statement: The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Deng, Y.; Hu, H.; Liu, L. Feedback control of digital chaotic systems with application to pseudorandom number generator. *Int. J. Mod. Phys. C* **2015**, *26*, 1550022. [CrossRef]
2. Zhang, Y.; Xiao, D.; Wen, W.; Nan, H.; Su, M. Secure binary arithmetic coding based on digitalized modified logistic map and linear feedback shift register. *Commun. Nonlinear Sci. Numer. Simul.* **2015**, *27*, 22–29. [CrossRef]
3. Wang, Y.; Zhang, Z.; Wang, G.; Liu, D. A pseudorandom number generator based on a 4D piecewise logistic map with coupled parameters. *Int. J. Bifurcat. Chaos* **2019**, *29*, 1950124. [CrossRef]
4. Falcioni, M.; Palatella, L.; Pigolotti, S.; Vulpiani, A. Properties making a chaotic system a good pseudo random number generator. *Phys. Rev. E* **2005**, *72*, 016220. [CrossRef]
5. Zheng, J.; Hu, H.; Ming, H.; Liu, X. Theoretical design and circuit implementation of novel digital chaotic systems via hybrid control. *Chaos Solitons Fractals* **2020**, *138*, 109863. [CrossRef]
6. Senouci, A.; Bouhedjeur, H.; Tourche, K.; Boukabou, A. FPGA based hardware and device-independent implementation of chaotic generators. *AEU-Int. J. Electron. C* **2017**, *82*, 211–220. [CrossRef]
7. Lozi, R. Emergence of randomness from chaos. *Int. J. Bifurcat. Chaos* **2012**, *22*, 1250021. [CrossRef]
8. Muhammad, A.S.; Özkaynak, F. SIEA: Secure Image Encryption Algorithm Based on Chaotic Systems Optimization Algorithms and PUFs. *Symmetry* **2021**, *13*, 824. [CrossRef]
9. Li, S.; Mou, X.; Cai, Y. Pseudo-random bit generator based on couple chaotic systems and its applications in stream-cipher cryptography. In Proceedings of the International Conference on Cryptology in India, Chennai, India, 16–20 December 2001; Rangan, C.P., Ding C., Eds.; Springer: Berlin/Heidelberg, Germany, 2001; pp. 316–329.
10. De Micco, L.; González, C.M.; Larrondo, H.A.; Martín, M.T.; Plastino, A.; Rosso, O.A. Randomizing nonlinear maps via symbolic dynamics. *Phys. A* **2008**, *387*, 3373–3383. [CrossRef]
11. Yuan, F.; Deng, Y.; Li, Y.; Chen, G. A cascading method for constructing new discrete chaotic systems with better randomness. *Chaos Interdiscip. J. Nonlinear. Sci.* **2019**, *29*, 053120. [CrossRef] [PubMed]
12. Setti, G.; Mazzini, G.; Rovatti, R.; Callegari, S. Statistical modeling of discrete-time chaotic processes-basic finite-dimensional tools and applications. *Proc. IEEE* **2002**, *90*, 662–690. [CrossRef]
13. Öztürk, I.; Kılıç, R. A novel method for producing pseudo random numbers from differential equation-based chaotic systems. *Nonlinear Dyn.* **2015**, *80*, 1147–1157. [CrossRef]
14. Li, S.; Mou, X.; Ji, Z.; Zhang, J.; Cai, Y. High-performance multimedia encryption system based on chaos. *Phys. Lett. A* **2003**, *307*, 22. [CrossRef]
15. Hu, H.; Liu, L.; Ding, N. Pseudorandom sequence generator based on the Chen chaotic system. *Comput. Phys. Commun.* **2013**, *184*, 765–768. [CrossRef]
16. Guan, Z.H.; Huang, F.; Guan, W. Chaos-based image encryption algorithm. *Phys. Lett. A* **2005**, *346*, 153–157. [CrossRef]
17. Antonelli, M.; De Micco, L.; Gonzalez, C.; Larrondo, H. Analysis of the digital implementation of a chaotic deterministic-stochastic attractor. In Proceedings of the 2012 Argentine School of Micro-Nanoelectronics, Technology and Applications (EAMTA), Córdoba, Argentina, 9–10 August 2012; pp. 73–78.
18. Lynnyk, V.; Sakamoto, N.; Čelikovskỳ, S. Pseudo random number generator based on the generalized Lorenz chaotic system. *IFAC-PapersOnLine* **2015**, *48*, 257–261. [CrossRef]
19. Rezk, A.A.; Madian, A.H.; Radwan, A.G.; Soliman, A.M. Multiplierless chaotic pseudo random number generators. *AEU-Int. J. Electron. C* **2020**, *113*, 152947. [CrossRef]
20. Machicao, J.; Bruno, O.M. Improving the pseudo-randomness properties of chaotic maps using deep-zoom. *Chaos Interdiscip. J. Nonlinear Sci.* **2017**, *27*, 053116. [CrossRef]
21. Ozkaynak, F. A novel random number generator based on fractional order chaotic Chua system. *Elektron. ir Elektrotechnika* **2020**, *26*, 52–57. [CrossRef]
22. He, S.; Sun, K.; Wu, X. Fractional symbolic network entropy analysis for the fractional-order chaotic systems. *Phys. Scr.* **2020**, *95*, 035220. [CrossRef]
23. Braun, M.; Golubitsky, M. *Differential Equations and Their Applications*; Springer: New York, NY, USA, 1983; Volume 1.
24. Runge, C. Über die numerische Auflösung von Differentialgleichungen. *Math. Ann.* **1895**, *46*, 167–178. [CrossRef]
25. Liu, Y.; Tong, X. Hyperchaotic system-based pseudorandom number generator. *IET Inf. Secur.* **2016**, *10*, 433–441. [CrossRef]
26. Alawida, M.; Samsudin, A.; Teh, J.S. Enhancing unimodal digital chaotic maps through hybridisation. *Nonlinear Dyn.* **2019**, *96*, 601–613. [CrossRef]
27. Murillo-Escobar, M.; Cruz-Hernández, C.; Cardoza-Avendaño, L.; Méndez-Ramírez, R. A novel pseudorandom number generator based on pseudorandomly enhanced logistic map. *Nonlinear Dyn.* **2017**, *87*, 407–425. [CrossRef]
28. Rezk, A.A.; Madian, A.H.; Radwan, A.G.; Soliman, A.M. Reconfigurable chaotic pseudo random number generator based on FPGA. *AEU-Int. J. Electron. C* **2019**, *98*, 174–180. [CrossRef]
29. Sprott, J.C. *Chaos and Time-Series Analysis*; Oxford University Press: Oxford, UK, 2003; Volume 69.
30. Kantz, H.; Schreiber, T. *Nonlinear Time Series Analysis*; Cambridge University Press: Cambridge, UK, 2004; Volume 7.
31. Strogatz, S.H. *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering*; Westview Press: Boulder, CO, USA, 1994.

32. Antonelli, M.; De Micco, L.; Larrondo, H. Measuring the jitter of ring oscillators by means of information theory quantifiers. *Commun. Nonlinear. Sci. Numer. Simul.* **2017**, *43*, 139–150. [CrossRef]
33. Bandt, C.; Pompe, B. Permutation entropy: A natural complexity measure for time series. *Phys. Rev. Lett.* **2002**, *88*, 174102. [CrossRef]
34. Rosso, O.A.; Larrondo, H.A.; Martín, M.T.; Plastino, A.; Fuentes, M.A. Distinguishing noise from chaos. *Phys. Rev. Lett.* **2007**, *99*, 154102. [CrossRef]
35. De Micco, L.; Larrondo, H.A.; Plastino, A.; Rosso, O.A. Quantifiers for randomness of chaotic pseudo-random number generators. *Philos. Trans. R. Soc. A* **2009**, *367*, 3281–3296. [CrossRef]
36. Wackerbauer, R.; Witt, A.; Atmanspacher, H.; Kurths, J.; Scheingraber, H. A comparative classification of complexity measures. *Chaos Solitons Fractals* **1994**, *4*, 133–173. [CrossRef]
37. Lopez-Ruiz, R.; Mancini, H.L.; Calbet, X. A statistical measure of complexity. *Phys. Lett. A* **1995**, *209*, 321–326. [CrossRef]
38. Lamberti, P.W.; Martín, M.T.; Plastino, A.; Rosso, O.A. Intensive entropic non-triviality measure. *Phys. A Stat. Mech. Appl.* **2004**, *334*, 119–131. [CrossRef]
39. Rosso, O.A.; De Micco, L.; Larrondo, H.A.; Martín, M.T.; Plastino, A. Generalized statistical complexity measure. *Int. J. Bifurcat. Chaos* **2010**, *20*, 775–785. [CrossRef]
40. Antonelli, M.; De Micco, L.; Larrondo, H.; Rosso, O.A. Complexity of Simple, Switched and Skipped Chaotic Maps in Finite Precision. *Entropy* **2018**, *20*, 135. [CrossRef]
41. Ribeiro, H.V.; Zunino, L.; Mendes, R.S.; Lenzi, E.K. Complexity–entropy causality plane: A useful approach for distinguishing songs. *Phys. A Stat. Mech. Appl.* **2012**, *391*, 2421–2428. [CrossRef]
42. Zunino, L.; Ribeiro, H.V. Discriminating image textures with the multiscale two-dimensional complexity-entropy causality plane. *Chaos Solitons Fractals* **2016**, *91*, 679–688. [CrossRef]
43. Rukhin, A.; Soto, J.; Nechvatal, J.; Smid, M.; Barker, E. *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications;* NIST: Gaithersburg, MD, USA, 2001.
44. Rössler, O.E. An equation for continuous chaos. *Phy. Lett. A* **1976**, *57*, 397–398. [CrossRef]
45. Sprott, J.C. Numerical Calculations of the Lyapunov Exponent. Available online: http://sprott.physics.wisc.edu/chaos/lyapexp.htm (accessed on 10 March 2020).
46. Micco, L.D.; Antonelli, M.; Larrondo, H. Stochastic degradation of the fixed-point version of 2D-chaotic maps. *Chaos Solitons Fractals* **2017**, *104*, 477–484. [CrossRef]
47. Lambić, D.; Nikolić, M. Pseudo-random number generator based on discrete-space chaotic map. *Nonlinear Dyn.* **2017**, *90*, 223–232. [CrossRef]
48. de la Fraga, L.G.; Torres-Pérez, E.; Tlelo-Cuautle, E.; Mancillas-López, C. Hardware implementation of pseudo-random number generators based on chaotic maps. *Nonlinear Dyn.* **2017**, *90*, 1661–1670. [CrossRef]
49. Teh, J.S.; Samsudin, A.; Al-Mazrooie, M.; Akhavan, A. GPUs and chaos: A new true random number generator. *Nonlinear Dyn.* **2015**, *82*, 1913–1922. [CrossRef]
50. Chandrasekaran, S.; Amira, A. High performance FPGA implementation of the Mersenne Twister. In Proceedings of the 4th IEEE International Symposium on Electronic Design, Test and Applications, Hong Kong, China, 23–25 January 2008; pp. 482–485.