Available online at www.sciencedirect.com
SCIENCE DIRECT®

Physica A 354 (2005) 281–300

PHYSICA A

www.elsevier.com/locate/physa

Statistical complexity measure of pseudorandom bit generators

C.M. González[a], H.A. Larrondo[a], O.A. Rosso[b,*]

[a]Facultad de Ingeniería, Universidad Nacional de Mar del Plata, Juan B. Justo 4302, 7600 Mar del Plata, Argentina

[b]Chaos & Biology Group, Instituto de Cálculo, Facultad de Ciencias Exactas y Naturales, Universidad de Buenos Aires, Pabellón II, Ciudad Universitaria, 1428 Ciudad de Buenos Aires, Argentina

Received 20 October 2004
Available online 18 April 2005

Abstract

Pseudorandom number generators (PRNG) are extensively used in Monte Carlo simulations, gambling machines and cryptography as substitutes of ideal random number generators (RNG). Each application imposes different statistical requirements to PRNGs. As L'Ecuyer clearly states "the main goal for Monte Carlo methods is to reproduce the statistical properties on which these methods are based whereas for gambling machines and cryptology, observing the sequence of output values for some time should provide no practical advantage for predicting the forthcoming numbers better than by just guessing at random". In accordance with different applications several statistical test suites have been developed to analyze the sequences generated by PRNGs. In a recent paper a new statistical complexity measure [Phys. Lett. A 311 (2003) 126] has been defined. Here we propose this measure, as a randomness quantifier of a PRNGs. The test is applied to three very well known and widely tested PRNGs available in the literature. All of them are based on mathematical algorithms. Another PRNGs based on Lorenz 3D chaotic dynamical system is also analyzed. PRNGs based on chaos may be considered as a model for physical noise sources and important new results are recently reported. All the design steps of this PRNG are described, and each stage increase the PRNG randomness using different strategies. It is shown that the MPR statistical

[*]Corresponding author. Tel./fax: +54 11 4576 3375.
E-mail addresses: cmgonzal@fi.mdp.edu.ar (C.M. González), larrondo@fi.mdp.edu.ar (H.A. Larrondo), oarosso@fibertel.com.ar, rosso@ba.net (O.A. Rosso).

doi:10.1016/j.physa.2005.02.054

complexity measure is capable to quantify this randomness improvement. The PRNG based on the chaotic 3D Lorenz dynamical system is also evaluated using traditional digital signal processing tools for comparison.

PACS: 03.67. −a; 89.70. + c; 03.65.Bz

Keywords: Random number generators; Statistical complexity

1. Introduction

Random number generators (RNGs) are essential in statistical studies in several fields. They may be based on physical noise sources or on mathematical algorithms, but in both cases truly random numbers are not obtained because of data acquisition systems in the first case or because machine precision in the second case. Instead, any real implementation actually produces a pseudorandom number generator (PRNG). In spite of this restriction PRNGs have been developed to fulfill many statistical properties required in applications, mainly Monte Carlo simulations, gambling machines and cryptography [1]. Chaotic dynamical systems are models of a lot of physical phenomena [2]. Their sensitivity to initial conditions and their broadband spectrum make them good candidates to generate PRNGs with a behavior very similar to physical noise sources. New important research papers have recently appeared concerning this kind of PRNGs [3–11].

There are several basic properties any good PRNG must fit: long cycle length, randomness, speed, reproducibility and portability. Several test suites [12] are readily available to researchers in academia and industry who wish to analyze their newly developed PRNG. Some general purpose test suites are *Diehard* by George Marsaglia [13], *Crypt-XS* by Helen Gustafson of the Queensland University of Technology [14], the National Institute of Standards and Technology Statistical Test Suite [15]. Additional requirements are imposed in view of the specific application. As L'Ecuyer clearly states "the main goal for Monte Carlo methods is to reproduce the statistical properties on which these methods are based whereas for gambling machines and cryptology, observing the sequence of output values for some time should provide no practical advantage for predicting the forthcoming numbers better than by just guessing at random" [16]. Of course a statistical test can never prove that a sequence generated by a PRNG is random (*because it is not !*), but it helps to detect certain kinds of weaknesses a generator may have. Furthermore none of these tests can prove that a given generator is reliable in all applications. Vattulainen et al. [17], for example, proposed three additional physical tests, to detect deficiencies of several PRNGs used in Monte Carlo simulations.

In the framework of dynamical system theory, the statistical characterization of deterministic sources of apparent randomness was studied by many authors. Tools as metric entropy, Lyapunov exponents, and fractal dimension [18,19] have shed much light into the intricacies of dynamical behavior by describing the unpredictability of dynamical systems. It is thus possible to detect the presence of chaotic behavior and to quantify its degree but, it is not straightforward to characterize the degree of unpredictability and randomness of a system. The statistical complexity measures (SCMs) are recently proposed as quantifiers of the degree of physical structure in a signal [20–22]. Due to its intrinsic characteristics of be null for total random process, the statistical complexity is a natural test for measure the goodness of PRNGs.

In this paper the statistical complexity measure introduced by Martin, Plastino and Rosso (MPR) [22] is used to evaluate three very well-known PRNGs available in the literature. It is shown that MPR statistical complexity measure constitutes a stable quantifier and it tends to a finite limit when the size of the analyzed sequence increases. A PRNG based on the Lorenz 3D chaotic dynamical system is also studied. Contrary to almost all chaos-based cryptographic algorithms available in the literature we prefer here the use of natural numbers. The reason is dynamical systems defined on the set of real numbers are difficult for practical realization and circuit implementation. Moreover the natural number PRNG presented here have been implemented in a medium size field programmable gate arrays (FPGA) [23]. Three different design stages of the PRNG are considered. Each design stage increases the PRNG randomness using different strategies and it is shown that the MPR statistical complexity measure is capable to quantify this randomness improvement. The PRNG is also evaluated using traditional digital signal processing tools (2D and 3D distributions, fast fourier transform, autocorrelation function, frequency test, serial test, poker test, runs test and autocorrelation test).

The paper is organized in the following way: In Section 2 we present the employed statistical complexity measure. We give the chaotic PRNG design stages in Section 3. Sections 4 deals with results for three well known and tested PRNGs available in the literature and also for the Lorenz-based PRNG and the conclusions are given in Section 5. The traditional digital signal processing tools used in the paper are described in the appendix.

2. Statistical complexity measure

Jaynes has long ago established the relevance of Information Theory [24] for theoretical physics. Two essential ingredients of Jaynes' program are (i) Shannon's logarithmic information measure

$$\mathrm{S}[P] = -\sum_{i=1}^{N} p_i \ln[p_i] \,, \tag{1}$$

regarded as the general measure of the uncertainty associated to probabilistic physical processes described by the probability distribution P and (ii) his celebrated maximum entropy principle. Ascertaining the degree of unpredictability and randomness of a system is not automatically tantamount to adequately grasping the correlational structures that may be present, i.e., to be in a position to capture the relationship between the components of the physical system. These structures strongly influence the character of the probability distribution that is able to describe the physics one is interested in. Randomness, on the one hand, and structural correlations on the other one, are not totally independent aspects of this physics. Certainly, the opposite extremes of perfect order and maximal randomness possess no structure to speak of. In between these two special instances a wide range of possible degrees of physical structure exists, degrees that should be reflected in the features of the underlying probability distribution. One would like that they be adequately captured by some functional $\mathcal{F}[P]$ in the fashion that Shannon's S captures randomness. A candidate to this effect has come to be called the *statistical complexity*. $\mathcal{F}[P]$ should, of course, vanish in the two special extreme instances mentioned above.

López-Ruiz, Mancini, and Calbet (LMC) have recently proposed a measure of statistical complexity, based on the notion of "disorder" (H) and "disequilibrium" (Q) [21]. It is rather easy to compute (it is evaluated in terms of ordinary statistical mechanics' concepts). The functional form of statistical complexity measure reads

$$C[P] = Q[P]H[P] . \tag{2}$$

For a given probability distribution $P = \{p_i, i = 1, \dots, N\}$ and its associate information measure S, we define an amount of "disorder" H in the fashion

$$H[P] = \mathrm{S}[P]/\mathrm{S}_{max} , \tag{3}$$

where $\mathrm{S}_{max} = \mathrm{S}[P_e]$ and P_e is the probability distribution that maximize the information measure H and characterizes equilibrium in Gibbs statistical mechanics, i.e., $P_e \equiv \{1/N, \dots, 1/N\}$. Note that $0 \leqslant H \leqslant 1$.

Following LMC one defines, in addition to H, the quantity Q as a "distance" in probability space. It measures "how far" P is located, in this space, from the distribution P_e. In their work LMC adopt the definition of Euclidean distance, $\mathcal{D}_E$, for the evaluation of Q:

$$\begin{aligned} Q[P] \equiv Q_{\mathrm{E}}[P, P_e] &= Q_0^{(\mathrm{E})} \mathcal{D}_E[P, P_e] \\ &= Q_0^{(\mathrm{E})} \sum_{i=1}^{N} \left\{ p_i - \frac{1}{N} \right\}^2 , \end{aligned} \tag{4}$$

with $Q_0^{(\mathrm{E})} = N/(N-1)$ so that $0 \leqslant Q_{\mathrm{E}} \leqslant 1$. The LMC statistical complexity reads

$$C^{(LMC)}[P] = Q_{\mathrm{E}}[P, P_e]H[P] . \tag{5}$$

It has been pointed out in Ref. [25] that the LMC statistical complexity measure is marred by some troublesome characteristics: (i) It is neither an intensive nor an extensive quantity. (ii) It vanishes exponentially in the thermodynamic limit for all one-dimensional, finite-range systems. The authors of Ref. [25] forcefully argue that a reasonable statistical complexity measure should: (iii) be able to distinguish among different degrees of periodicity and (iv) vanish only for periodicity unity. Finally, and with reference to the ability of the LMC measure to adequately capture essential dynamical aspects, some difficulties have also been encountered in Ref. [26]. With the product functional form for the statistical complexity it is impossible to solve the deficiency (ii).

The straightforward definition of distance, $\mathscr{D} \equiv \mathscr{D}_E$, has been criticized by Wootters in an illuminating communication [27]. Essentially, the Euclidean norm ignores the stochastic nature of the probability distribution. Martin, Plastino and Rosso following Wootters' pioneering essay, redefine the notion of disequilibrium and recast it à la Wooters [22].

$$Q[P] \equiv Q_W[P, P_e] = Q_0^{(W)} \mathscr{D}_W[P, P_e] \\ = Q_0^{(W)} \cos^{-1}\left\{ \sum_{i=1}^{N} [p_i]^{1/2} \left[\frac{1}{N}\right]^{1/2} \right\} \tag{6}$$

with $Q_0^{(W)} = 1/\cos^{-1}\{[1/N]^{1/2}\}$ and $0 \leqslant Q_W \leqslant 1$. The Martin, Plastino, and Rosso statistical complexity reads

$$C^{(MPR)}[P] = Q_W[P, P_e] H[P] \,. \tag{7}$$

Application of the ensuing statistical complexity measure to the logistic map shows that important improvements are thereby achieved [22]. The new measure does behave in a manner compatible with that of the Lyapunov exponents, while the original LMC statistical complexity does not. We recognize the fact that not all the objections of Ref. [25] have been overcome for the new statistical complexity. In particular, $C^{(MPR)}$ is neither an intensive nor an extensive quantity. However, for our present propose of test the goodness of PRNG this property is not necessary. That is, a PRNG will be better as well as the $C^{(MPR)}[\{s\}] \to 0$, with $\{s\}$ the number sequence generated by the PRNG. It should be noticed that these statistical complexity measures (Eqs. (5) and (7)) are not a trivial function of the entropy, in the sense that, for a given H-value, there exists a range of complexities between a minimal value C_{min} and a maximal value C_{max} [21,28]. Thus, evaluating the complexity provides one with important additional information regarding the peculiarities of a probability distribution.

In the case of digital PRNGs two different kind of sequences are to be considered: the natural numbers sequence $\{s_n\}$ and the bits sequence $\{s_b\}$ corresponding to its binary representation. The MPR statistical complexity measure is evaluated using the sequence $\{s_n\}$ (words of 8-bits). Some classical tests are applied to $\{s_n\}$ and others to $\{s_b\}$.

3. PRNG based on the Lorenz 3D chaotic system

We propose here a PRNG, based on the classical Lorenz 3D chaotic dynamical system. This PRNG may be implemented on a medium size field programmable gate arrays (FPGA) [23]. The designing process has five steps. After step 3 the output PRNG1 is obtained. Steps 4 and 5 increase the PRNG output randomness giving, respectively, PRNG2 and PRNG3 (the final PRNG).

Step 1: We start considering the classical Lorenz system:

$$\begin{aligned}\frac{\mathrm{d}x}{\mathrm{d}t} &= -\delta(x-y)\,,\\ \frac{\mathrm{d}y}{\mathrm{d}t} &= \Gamma x - y - xz\,,\\ \frac{\mathrm{d}z}{\mathrm{d}t} &= xy - bz\,,\end{aligned} \tag{8}$$

where δ, Γ and b are the available constructive parameters. This is a continuous system having chaotic dynamics for some ranges of the constructive parameters.

Step 2: Eqs. (8) are discretized using the Euler approach:

$$\begin{aligned}\widetilde{X}_{n+1} &= \widetilde{X}_n + k[-\delta(\widetilde{X}_n - \widetilde{Y}_n)]\,,\\ \widetilde{Y}_{n+1} &= \widetilde{Y}_n + k[\Gamma\widetilde{X}_n - \widetilde{Y}_n - \widetilde{X}_n\widetilde{Z}_n]\,,\\ \widetilde{Z}_{n+1} &= \widetilde{Z}_n + k[\widetilde{X}_n\widetilde{Y}_n - b\widetilde{Z}_n]\,,\end{aligned} \tag{9}$$

where k is the time-scaling parameter. Uppercase letters are used in Eq. (9) to name the discrete state variables.

Step 3: Floating point operations are very resource consuming tasks when they are implemented in a FPGA. Then, in order to reduce the hardware, we employ binary arithmetic and we restrict the divisors to powers of two by means of the following biasing and scaling transformations:

$$\begin{aligned}X_n &= (\widetilde{X}_n + B)S\,,\\ Y_n &= (\widetilde{Y}_n + B)S\,,\\ Z_n &= (\widetilde{Z}_n + B)S\,,\end{aligned} \tag{10}$$

where B and S are the biasing and scaling parameters, respectively. Replacing Eq. (10) in Eq. (9) we obtain

$$\begin{aligned}X_{n+1} &= X_n + k\delta(Y_n - X_n)\,,\\ Y_{n+1} &= (1-k)Y_n + k(B+\Gamma)X_n + kBZ_n - \frac{k}{S}X_nZ_n + kBS(1-\Gamma-B)\,,\\ Z_{n+1} &= (1-kb)Z_n - kB(X_n+Y_n) + \frac{k}{S}X_nY_n + kBS(B+b)\,.\end{aligned} \tag{11}$$

The values adopted in this paper are

$$k = \tfrac{1}{64};\, \delta = 8;\, \Gamma = 24;\, b = 2;\, B = 40;\, S = 512\,; \tag{12}$$

and the resulting systems is

$$X_{n+1} = X_n + \mathit{floor}\left[\frac{Y_n}{8}\right] - \mathit{floor}\left[\frac{X_n}{8}\right],$$

$$
\begin{aligned}
Y_{n+1} &= Y_n - floor\left[\frac{Y_n}{64}\right] + X_n + floor\left[\frac{Z_n}{2}\right] \\
&\quad + floor\left[\frac{Z_n}{8}\right] - floor\left[\frac{X_n}{256}\right] floor\left[\frac{Z_n}{128}\right] - 20160\,, \\
Z_{n+1} &= Z_n - floor\left[\frac{Z_n}{32}\right] - floor\left[\frac{(X_n + Y_n)}{2}\right] \\
&\quad - floor\left[\frac{(X_n + Y_n)}{8}\right] + floor\left[\frac{X_n}{256}\right] floor\left[\frac{Y_n}{128}\right] + 13440\,. \qquad (13)
\end{aligned}
$$

Several important remarks:

- The values chosen for the parameters produce an easy implementation. In fact the values adopted in Eqs. (12) make that system given by Eqs. (13) only requires two multipliers. The remaining operations are additions, subtractions and divisions by powers of two, as pointed above.
- The scaling parameter S controls the truncation effect and it also determines the range of the state variables. Consequently it defines the number of bits required for the binary representation.
- The discrete version of any chaotic continuous dynamical system (as it is the case with the Lorenz system) is always periodic and usually several orbits, with different periodicity coexist. The initial conditions must be chosen in order to assure a desired period. The numerical analysis showed us that for the selected value $S = 512$, and the selected initial state $x_0 = 18\,503$, $y_0 = 21\,315$ and $z_0 = 32\,032$, an orbit with 78 782 different states is obtained.
- The floor operation in Eq. (13) is required to reproduce the way a FPGA manages the divisions in binary arithmetic.

The basic structure of PRNG1 is shown in Fig. 1. The signal clk_{lorenz} is the FPGA clock and it determines the time-scaling of the dynamical system measured in seconds. In Fig. 2 a typical 3D trajectory of system given by Eq. (13) is shown. A

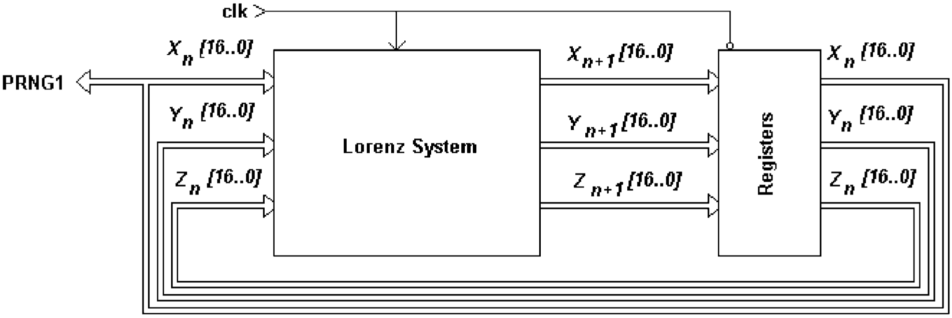

Fig. 1. PRNG Lorenz circuit—PRNG1.

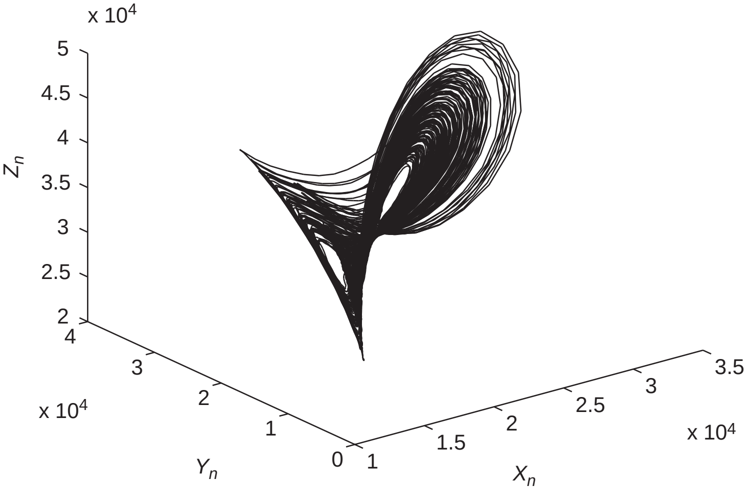

Fig. 2. 3D attractor of the discrete map of Eq. (13).

discrete version of the Lorenz butterfly is apparent in this figure. Note that the ranges of the state variables require at least a 17 bits representation.

Step 4: The period 78 782 pointed above is not large enough for some applications. To increase it an option is to use a higher S. For example with $S = 2^{24}$, 32 bits are required and the period $\gtrsim 2^{34}$. Another option, that does not require more hardware, is to perturb the output. This is the approach adopted here: the least significant byte of X, it means $X[7..0]$, is periodically XORed with the least significant byte of Y, it means $Y[7..0]$ to produce the least significant byte of X' and the most significant byte of X' remains identical to the most significant byte of X. X' is the output of PRNG2. The period of the perturbation is N times the clock cycle. By this method the period of X' (PRNG2) is increased over 2^{22}.

Step 5: As we will show below, the FFT of X has a maximum at low frequencies reflecting that the most significant bits of X' are periodic. If we disregard $X'[16..8]$ and we only use $X'[7..0]$ as pseudorandom output (PRNG3), randomness is highly improved, as will be shown by the evaluation of the MPR statistical complexity, the FFT and the autocorrelation function. Fig. 3 shows the final block-diagram of the PRNG (see in this figure the auxiliary output PRNG2 and PRNG3, the final PRNG).

4. Results

Three algorithmic very well-tested PRNGs [13] were used to evaluate the MPR statistical complexity as randomness quantifier. They are:

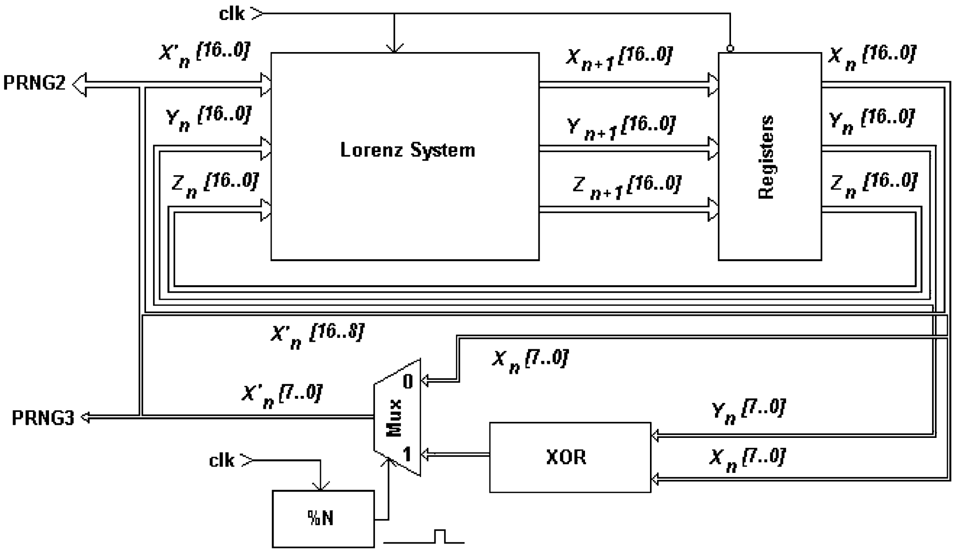

Fig. 3. PRNG Lorenz circuit—PRNG2 and PRNG3.

(a) A multiply-with-carry (*MWC*) PRNG generated by the following expressions:

$$X_n = \mathrm{Mod}\{aX_{n-1} + carry_{n-1}; 2^{32}\}\ ,$$
$$carry_n = floor\left(\frac{X_n}{2^{32}}\right)\ . \qquad (14)$$

The initial condition is random. Here we used $X_0 = 1\,947\,362\,876$ and $a = 2\,131\,995\,753$.

(b) A generator identified as *Mother* given by

$$X_n = \mathrm{Mod}\{2\,111\,111\,111 X_{n-4} + 1492 X_{n-3} + 1776 X_{n-2} + 5115 X_{n-1} + carry_{n-1}; 2^{32}\}\ ,$$
$$carry_n = floor\left(\frac{X_n}{2^{32}}\right)\ . \qquad (15)$$

The initial conditions are random. Here we used: $X_0 = 1\,143\,897\,285$, $X_{-1} = 1\,549\,345\,678$, $X_{-2} = 205\,987\,485$ and $X_{-3} = 164\,987\,491$.

(c) The generator identified as **Combo** which is a combination of two simpler generators:

$$X_n = \mathrm{Mod}\,\{X_{n-1} \cdot X_{n-2}; 2^{32}\}\ ,$$
$$Y_n = \mathrm{Mod}\,\{30903\, Y_{n-1} + carry_{n-1}; 2^{16}\}\ ,$$
$$Z_n = \mathrm{Mod}\,\{X_n + Y_n; 2^{16}\}\ ,$$
$$carry_n = floor\left(\frac{X_n}{2^{16}}\right)\ . \qquad (16)$$

The initial conditions are random. Here we used $X_0 = 356\,819\,112$, $X_{-1} = 455\,997\,113$ and $Y_0 = 158\,644\,912$.

All these PRNG were proposed by Marsaglia [13] and it was checked out that all of them support the Diehard test suite. For each PRNG we generated 2.500.000 32 bits integer. The bits were arranged as 80.000.000 bits long strings. The MPR statistical complexity measure of each PRNG was evaluated in two different ways:

(1) The bits were grouped in 32 bits-integer (the original numbers generated by the PRNGs) and MPR statistical complexity was evaluated using Eq. (7).
(2) The bits were grouped as 8 bits integers and MPR statistical complexity were again evaluated using Eq. (7).

Fig. 4 shows the MPR statistical complexity measure $C^{(MPR)}$ as a function of the number of words for the analyzed sequence, for the three generators. In Fig. 4a calculations were made with 32 bits integers and then the maximum number of words corresponds to 80.000.000 bits. In Fig. 4b calculations were made with 8 bits-integers and consequently the maximum number of words corresponds to 20.000.000 bits. The MPR statistical complexity is stable and it tends to a very small value when the number of bits tends to infinity. In Fig. 5 it is shown that the normalized entropy H, calculated using Eq. (3) tends to 1 as the number of words of the analyzed sequence increases. Fig. 5a (5b) display the results when 32 bits-integers (8 bits-integers) are considered. Consequently the small value of MPR statistical complexity is produced by randomness (for periodic sequences $C^{(MPR)} \sim 0$ and $H \sim 0$).

In the case of the Lorenz-based PRNG it is shown that $C^{(MPR)}$ may be used as a quality parameter because it quantifies the randomness improvement that each design stage produces. We iterate Eq. (13) a million times obtaining a sequence $\{s\}$ consisting of a million 17-bits-length-binary-words. First parameter Γ runs free from 0 to 50 in order to confirm that $\Gamma = 24$ is a good choice (there are other possible values compatible with all the restrictions imposed). The histogram of the output is made using 256 levels and then normalized to obtain the associate probability set $\{p_i, i = 1, \ldots, 256\}$. Using expressions given by Eqs. (3) and (6) the values of MPR statistical complexity, $C^{(MPR)}$ (see Eq. (7)) are obtained.

Figs. 6a–c (Figs. 7a–c) show the results for MPR statistical complexity (normalized entropy H) corresponding to outputs PRNG1, PRNG2 and PRNG3, respectively. Let us stress again that a zero value of $C^{(MPR)}$ indicates a truly random sequence or a perfectly periodic one but in the first case normalized entropy is 1 and in the second case it is zero or almost zero. In the case of pseudorandom outputs it is expected that $C^{(MPR)}$ decreases with increasing randomness. Fig. 6 show that it is clear that $\Gamma = 24$ is a good choice because $C^{(MPR)}$ is minimum around this value. Furthermore, the value of $C^{(MPR)}$ is 0.3 for PRNG1 and PRNG2 but it decreases to 0.05 for PRNG3 showing that the sequence $\{s\}$ is more random.

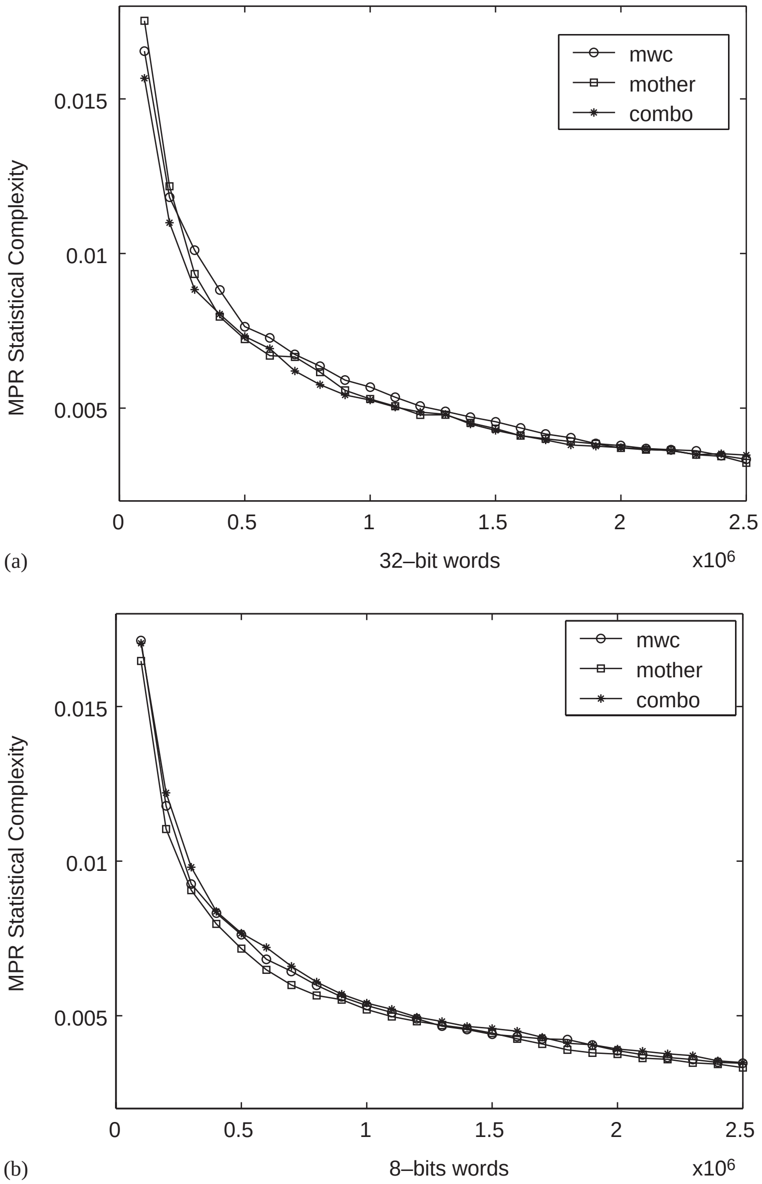

Fig. 4. MPR statistical complexity measure $C^{(MPR)}$ as a function of the number of words for the analyzed sequence generated by three very well-tested PRNGs (see text). (a) 32 bits-word, (b) 8 bits-word.

These observations may be confirmed by means of three typical signal processing tools (see the appendix for a short description of these tools): (a) 2D and 3D graphic representation of $\{s\}$ distributions (see Figs. 8 and 9), (b) the fast fourier transform (FFT) (see Fig. 10) and (c) the autocorrelation function (see Fig. 11). Figs. 8 and 9

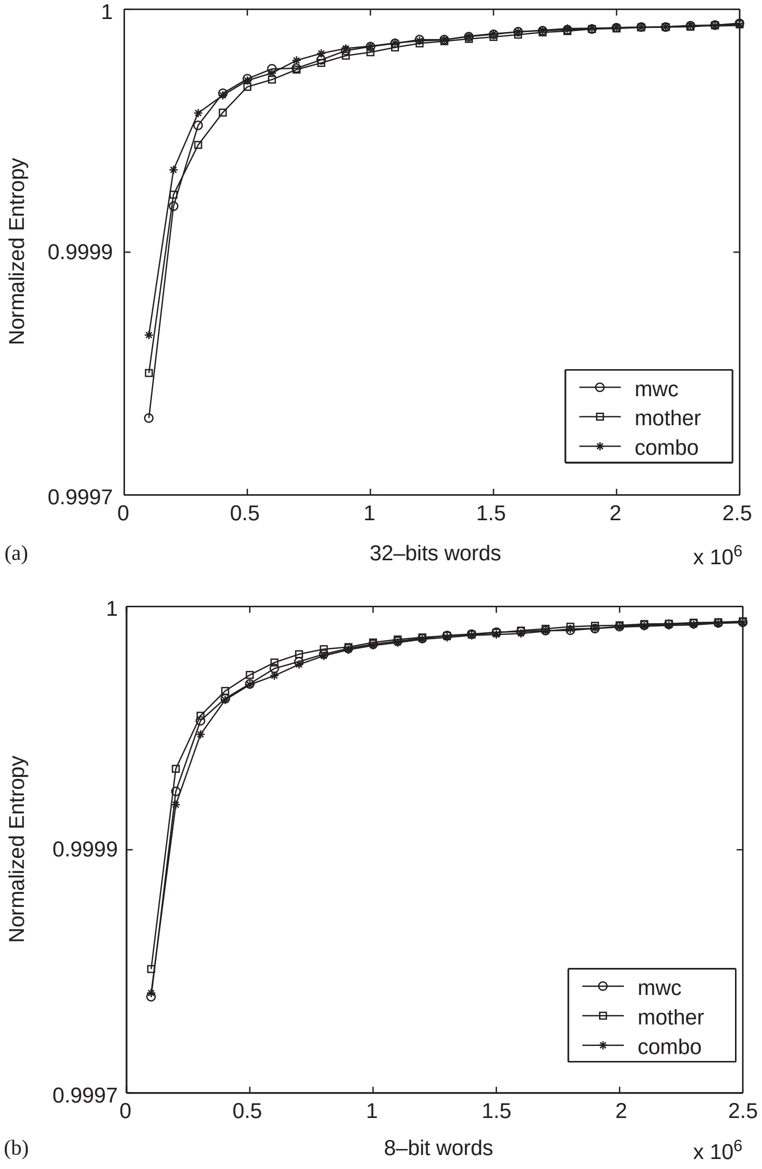

Fig. 5. Normalized entropy H as a function of the number of words for the analyzed sequence generated by three very well-tested PRNGs (see text). (a) 32 bits-word, (b) 8 bits-word.

show that the sequence $\{s\}$ is more random in the case of PRNG3. In fact, only in this case the points cloud present almost uniform distribution that tend to fill the complete 2D and 3D spaces. The power spectra obtained by FFT (see Figs. 10) of $\{s\}$ is very similar for both PRNG1 and PRNG2. They have low-frequency components

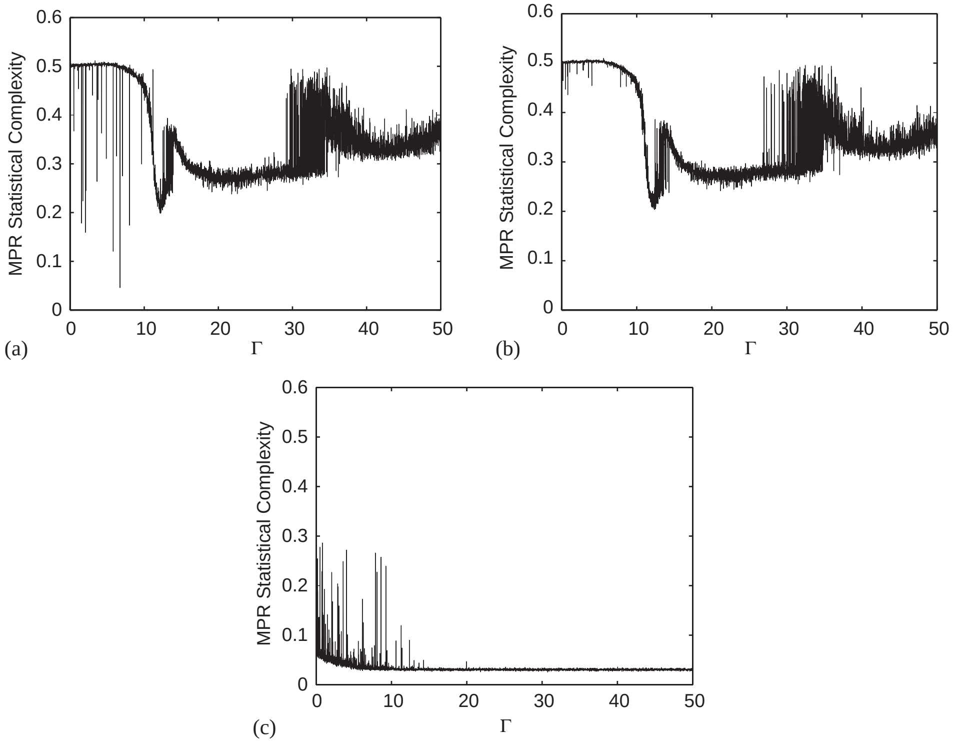

Fig. 6. Statistical complexity $C^{(MPR)}$ for the corresponding sequence generated by Lorenz-based PRNG as a function of the parameter Γ. (a) PRNG1, (b) PRNG2 and (c) PRNG3.

around 0.2×10^{24} samples. This value corresponds to the periodicity of the highest bits of the sequence. As pointed above PRNG3 was obtained removing these bits and its FFT looks like a white-noise spectrum, as expected for a random signal.

Five standard statistical tests (see the appendix) usually employed in PRNGs evaluation were also applied to PRNG1, PRNG2 and PRNG3 for comparative proposes with the new quality quantifier $C^{(MPR)}$. Results are shown in Table 1. Five outcomes of 65 536 bits length each were analyzed using all the tests. The last line includes the MPR statistical complexity measure here proposed as an additional test. As can be seen, all the tests are passed by PRNG3. The $C^{(MPR)}$ is substantially lower for this PRNG.

5. Conclusions

In summary we showed, with the help of three well-known PRNGs, that the MPR statistical complexity is a stable quantifier of the PRNG randomness. Its value tends to zero for long sequences when good PRNGs are tested. The normalized entropy

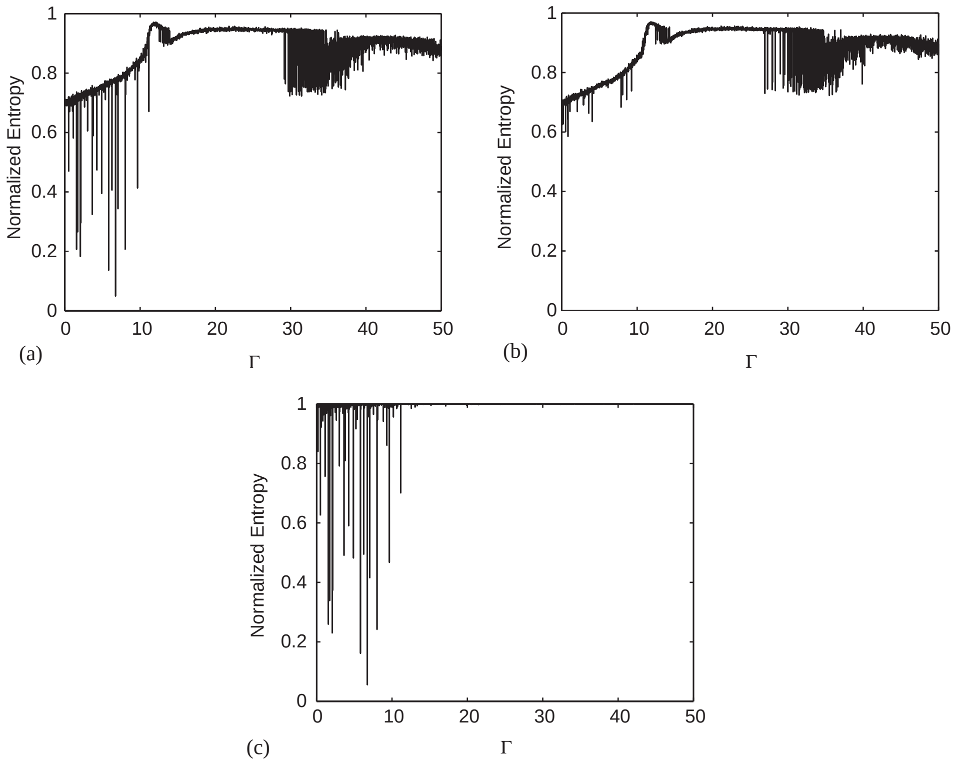

Fig. 7. Normalized entropy H for the corresponding sequence generated by Lorenz-based PRNG as a function of the parameter Γ. (a) PRNG1, (b) PRNG2 and (c) PRNG3.

must be verified to be close to one to assure that the low $C^{(MPR)}$ value corresponds to randomness. The Lorenz-based PRNG presented may be thought as a model of some physical noise source. It is easy to be hardware implemented. Several design strategies were applied to increase its randomness and it was shown that $C^{(MPR)}$ statistical complexity may be used as a quality quantifier of the benefits produced by each design stage.

Acknowledgements

This work was partially supported by Universidad Nacional de Mar del Plata, ANPCyT (PICTO 11-090076); CONICET(PIP 0029/98) (Argentina) and the International Office of BMBF (ARG-4-G0A-6A and ARG01-005), Germany. Two of us (CMG and HAL) acknowledge fruitful discussions with Eduardo Boemo and Carlos Angiano.

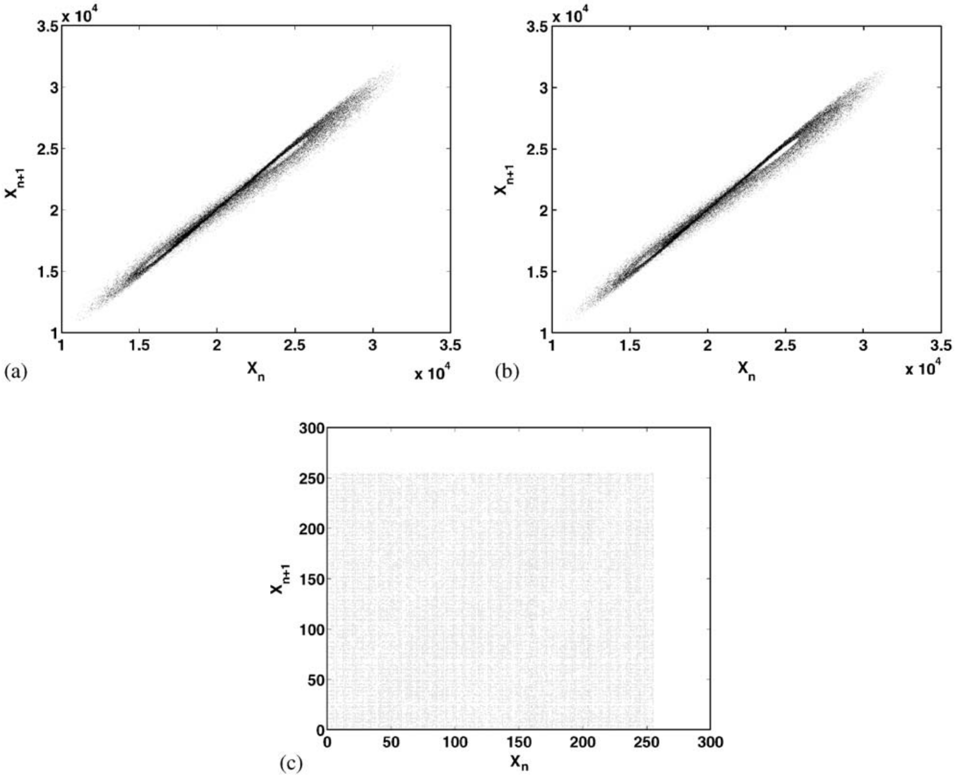

Fig. 8. First-order distribution for the Lorenz-based PRNG sequences: (a) PRNG1, (b) PRNG2 and (c) PRNG3. In all cases $\Gamma = 24$.

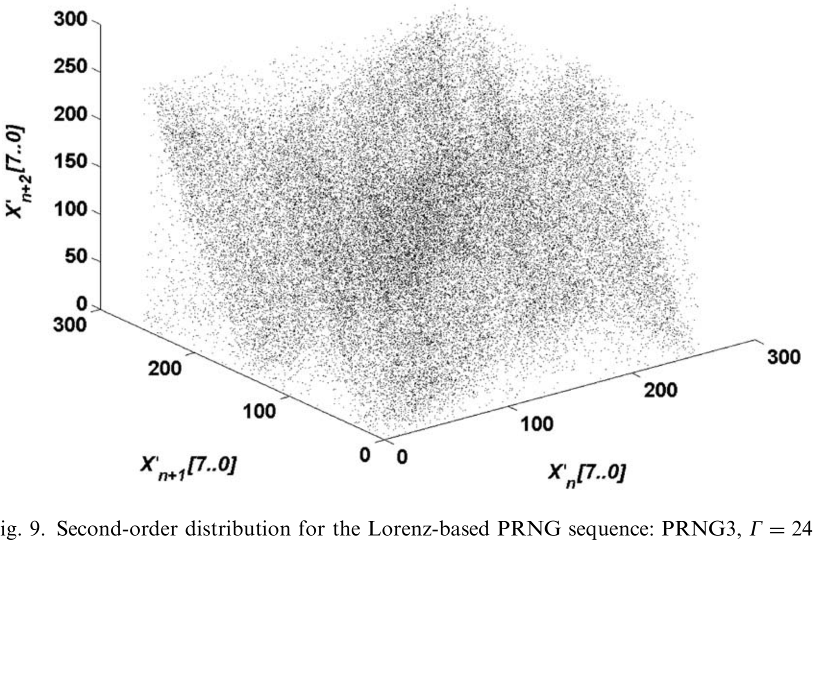

Fig. 9. Second-order distribution for the Lorenz-based PRNG sequence: PRNG3, $\Gamma = 24$.

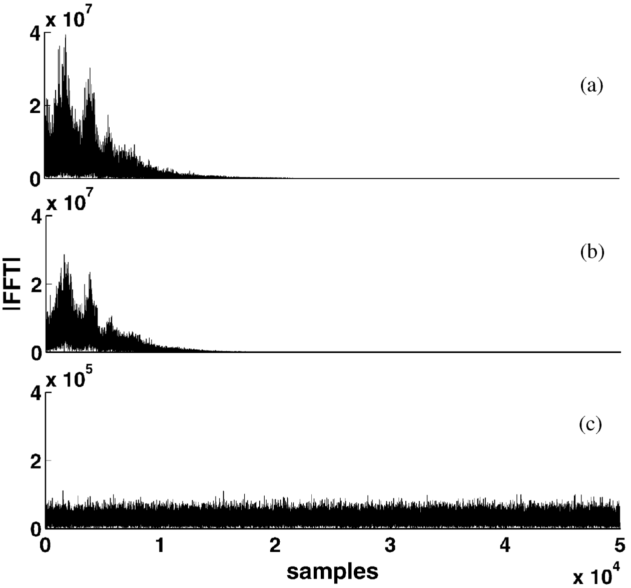

Fig. 10. FFT of the sequences generated by the Lorenz-based PRNG sequences: (a) PRNG1, (b) PRNG2 and (c) PRNG3. In all cases $\Gamma = 24$.

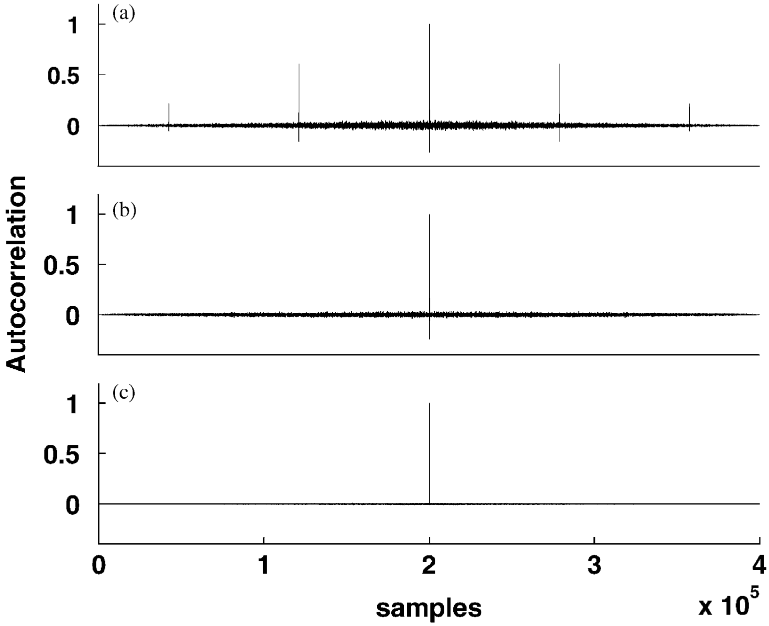

Fig. 11. Autocorrelation of the sequence generated by the Lorenz-based PRNG sequences: (a) PRNG1, (b) PRNG2 and (c) PRNG3. In all cases $\Gamma = 24$.

Table 1
Classical statistical tests and MPR statistical complexity for 5 sequences $\{s\}$ corresponding to the three different PRNG based on Lorenz chaotic dynamical system

Test	*Generator*		
	PRNG1 (X[16..0])	PRNG2 (X'[16..0])	PRNG3 (X'[7..0])
Frequency test	599.4846	652.2437	1.9338
$n = 65536$,	682.5156	719.3292	0.0005
$\alpha = 0.05, v = 1$.	665.8819	636.3795	0.1846
If $X < 3.8415$	607.9307	658.2432	0.1914
$\Rightarrow$ Test is passed	693.9932	629.3043	0.4727
Serial test	685.3358	707.4728	4.5619
$n = 65536$,	733.3969	788.0957	2.7051
$\alpha = 0.05, v = 2$.	710.6938	663.9380	1.7307
If $X < 5.9915$	641.7596	696.3902	0.3863
$\Rightarrow$ Test is passed	727.6159	665.5440	0.6852
Poker test	1445.0000	1469.6250	237.4375
$n = 65536$[a],	1577.9375	1658.6875	282.1875
$\alpha = 0.05, v = 255$.	1467.9375	1361.4375	270.9375
If $X < 293.2478$	1279.0625	1487.9375	266.2500
$\Rightarrow$ Test is passed	1502.0000	1439.5000	255.1875
Runs test	1306.2897	1351.9627	18.6459
$n = 65536$,	1436.6515	1436.7306	22.4999
$\alpha = 0.05, v = 26$.	1302.7890	1254.2268	30.3672
If $X < 31.4104$	1142.0282	1319.7987	13.7675
$\Rightarrow$ Test is passed	1321.1954	1258.6697	11.0715
Autocorrelation test	9.2852	7.4571	1.6211
$n = 65536$, $d = 1$,	7.1602	8.3165	−1.6445
$\alpha = 0.05$.	6.7227	5.2852	−1.2461
If $\lvert X\rvert < 1.6449$	5.8477	6.2071	−0.4414
$\Rightarrow$ Test is passed	5.8321	6.0508	−0.4492
Statistical complexity	0.271720	0.279716	0.021893
$C^{(MPR)}$,	0.274818	0.269337	0.020938
$n = 65536$.	0.273098	0.285062	0.020157
	0.272643	0.266428	0.021197
	0.271272	0.291553	0.020337

[a] It divides the n bits-length sequence s (65 536 bits) into $k = 8192$ non-overlapping subsequences m, 8 bits-length each.

Appendix: Classical statistical tests

In present appendix we describe, in short way, several classical statistical tests applied to the sequences $\{s_n\}$ or $\{s_b\}$ generated by the three Lorenz-based PRNGs [29–31].

- *2D and 3D distributions*: This is a graphic or visual tool and consist in plot two or three successive values of $\{s_n\}$, i.e.: (s_i, s_{i+1}) or (s_i, s_{i+1}, s_{i+2}), into a 2D plane or 3D space, respectively. The sequence $\{s_n\}$ will be more random if the cloud of points tends to fill the complete 2D or 3D space in an uniform way.
- *FFT*: The FFT of the sequence $\{s_n\}$ is performed and the corresponding power spectrum is computed. A complete flat power spectrum, with almost equal frequency contribution for all frequencies, is indicative of a total random series.
- *Autocorrelation Function*: The discrete autocorrelation function of the sequence $\{s_n\}$ is just the discrete correlation of the sequence with itself, i.e., $Corr(s,s)_j = \sum_k s_{nk} s_{nk+j}$, where j is the lag. Almost constant value of autocorrelation, independent of the value of the lag$j$, will be indicative of an uncorrelated random series.
- *Frequency test*: The purpose of this test is to determine whether the number of 0's and 1's, n_0 and n_1, are approximately the same, as would be expected for a random sequence $\{s_b\}$. The statistic used is

  $$X_1 = \frac{(n_0 - n_1)^2}{n} \, . \tag{17}$$

  If $\{s_b\}$ is a truly random sequence and $n \to \infty$, X_1 would have a χ^2 distribution with $v = 1$ degree of freedom. Let α be the significance level and let x_α be the threshold, meaning that for a χ^2 distribution with $v = 1$ degree of freedom, $P(X > x_\alpha) = \alpha$. The test is passed if $X_1 < x_a$. With $\alpha = 0.05$ the test is passed if $X_1 < 3.8415$ [29].
- *Serial test*: Let n_{00}, n_{01}, n_{10} and n_{11} be the number of occurrences of 00, 01, 10 and 11, respectively, in the sequence $\{s_b\}$, when the pairs are allowed to overlap, and let n_0 and n_1 be the number of zero's and one's as above: the following random variable is evaluated:

  $$X_2 = \frac{4}{(n-1)}\left(n_{00}^2 + n_{01}^2 + n_{10}^2 + n_{11}^2\right) - \frac{2}{n}\left(n_0^2 + n_1^2\right) + 1 \, . \tag{18}$$

  If $\{s_b\}$ is a truly random sequence and $n \to \infty$, this variable has a χ^2 distribution with $v = 2$ degrees of freedom. Taking $\alpha = 0.05$ the test is passed if $X_2 < 5.9915$ [29].
- *Poker test*: Divide the sequence $\{s_b\}$ of length n into k non-overlapping parts m bits length each, where m is a positive integer such that $(n/m) \geqslant 5 \times 2^m$ and $k = (n/m)$. Let n_i be the number of occurrences of the ith type of sequence of length m, $1 \leqslant i \leqslant k$. The poker test determines whether these subsequences of length m appear approximately the same number of times in $\{s_b\}$, as would be expected

for a random sequence. Let us define the random variable:

$$X_3 = \frac{2^m}{k}\left(\sum_{i=1}^{2^m} n_i^2\right) - k\,. \tag{19}$$

If $\{s_b\}$ is a truly random sequence and $n \to \infty$, this variable has a χ^2 distribution with $v = 2^m - 1$ degrees of freedom. Taking $\alpha = 0.05$ and $v = 255$ the test is passed if $X_3 < 293.2478$ [29].

- *Runs test*: A run of a sequence $\{s_b\}$ is a subsequence of length i consisting of consecutive 0's (*gap*) or consecutive 1's (*block*), which is neither preceded nor succeeded by the same symbol. In a truly random sequence of length n the expected number of *gaps* (or *blocks*) is

$$e_i = \frac{(n-i+3)}{2^{i+2}}\,, \tag{20}$$

Let k be the highest i $e_k \geqslant 5$. Let B_i and G_i be the number of *blocks* and *gaps* of length i for each i, $1 \leqslant i \leqslant k$. The statistic used is

$$X_4 = \sum_{i=1}^{k} \frac{(B_i - e_i)^2}{e_i} + \sum_{i=1}^{k} \frac{(G_i - e_i)^2}{e_i}\,. \tag{21}$$

If $\{s_b\}$ is a truly random sequence and $n \to \infty$, this variable has a χ^2 distribution with $v = 2k - 2$ degrees of freedom. Taking $\alpha = 0.05$ and $v = 26$ the test is passed if $X_4 < 31.4104$ [29].

- *Autocorrelation test*: The purpose of this test is to check for correlations between the sequence $\{s_b\}$ and shifted versions of it. Let d be a fixed integer and $1 \leqslant d \leqslant (n/2)$. The number of bits in $\{s_b\}$ not equal to their d-shifts is

$$A(d) = \sum_{i=0}^{n-d-1} s_i \oplus s_{i+d}\,, \tag{22}$$

where $\oplus$ denotes the XOR operator. The following random variable is defined:

$$X_5 = \frac{2[A(d) - (n-d)/2]}{\sqrt{n-d}}\,. \tag{23}$$

If $\{s_b\}$ is a truly random sequence and $n \to \infty$, this random variable has a normal distribution $N(0,1)$. Taking $\alpha = 0.05$ the test is passed if $\mid X_5 \mid < 1.6449$ [29].

References

[1] See the excellent web page on random number generation at http://cgm.cs.mcgill.ca/~luc/rng.html
[2] H.G. Schuster, Deterministic Chaos—An introduction, Physik Verlag, Weinheim, 1984.
[3] T. Stojanovski, L. Kocarev, IEEE Trans, Circuits Systems I 48 (2001) 281.
[4] T. Stojanovski, L. Kocarev, IEEE Trans, Circuits Systems I 48 (2001) 382.
[5] L. Kocarev, G. Jakimoski, IEEE Trans, Circuits Systems I 50 (2003) 123.
[6] L. Kocarev, IEEE Circuits Systems Mag. 1 (2001) 6.

[7] M. Jessa, IEEE Trans, Circuits Systems I 49 (2002) 84–89.
[8] Shujun Li, Ph.D Dissertation of Xi an Jiaotong University. Available on-line at http://www.hooklee.com/Thesis/ethesis.pdf
[9] G. Álvarez, F. Montoya, M. Romera, G. Pastor, Phys. Lett. A 306 (2003) 200.
[10] G. Álvarez, F.Montoya, G. Pastor, M. Romera, Breaking a secure communication scheme based on the phase synchronization of chaotic systems, available online at http://arxiv.org/pdf/nlin.CD/0311040
[11] M.S. Baptista, Phys. Lett. A 240 (1998) 50.
[12] J. Soto, Statistical testing of random number generators, available online at http://www.itl.nist.gov/div893/staff/soto/jshome.html
[13] G. Marsaglia, Diehard statistical tests, available online at http://stat.fsu.edu/~geo/diehard.html
[14] H.M. Gustafson, E.P. Dawson, L. Nielsen, W.J. Caelli, J. Comput. Security 13 (1994) 687.
[15] A.L. Rukhin, Testing randomness: a suite of statistical procedures, Theory Probab. Appl. 45 (2000) 111.
[16] P. L'Ecuyer, Random number generation, in: J.E. Gentle, W. Haerdle, Y. Mory (Eds.), Handbook of Computational Statistics, Springer, Berlin, 2004. Available from http://www.iro.umontreal.ca/~lecuyer/myftp/papers/handstat.pdf
[17] I. Vattulainen, T. Ala-Nissila, K. Kankaala, Phys. Rev. Lett. 73 (1994) 2513.
[18] E. Ott, T. Sauer, J.A. Yorke, Copying with Chaos, Wiley, New York, 1994.
[19] E. Ott, Chaos in Dynamical Systems, Cambridge University Press, New York, 1993.
[20] J.S. Shiner, M. Davison, P.T. Landsberg, Phys. Rev. E 59 (1999) 1459.
[21] R. López-Ruiz, H.L. Mancini, X. Calbet, Phys. Lett. A 209 (1995) 321.
[22] M.T. Martin, A. Plastino, O.A. Rosso, Phys. Lett. A 311 (2003) 126.
[23] C. M. González, H. A. Larrondo, C. A. Gayoso, L. J. Arnone, E. I. Boemo, available online at http://arXiv.org/abs/cs.CR/0402056
[24] T.M. Cover, J.A. Thomas, Elements of Information Theory, Wiley, New York, 1991.
[25] D.P. Feldman, J.P. Crutchfield, Phys. Lett. A 238 (1998) 244.
[26] C. Anteneodo, A.R. Plastino, Phys. Lett. A 223 (1996) 348.
[27] W.K. Wootters, Phys. Rev. D 23 (1981) 357.
[28] M. T. Martin, Ph.D. Thesis, Department of Mathematics, Faculty of Sciences University of La Plata, La Plata, Argentina, 2004.
[29] A. Menezes, P. van Oorschot, S. Vanstone, Handbook of Applied Cryptography, CRC Press, Boca Raton, 1997.
[30] A. Papoulis, Probability, Random Variables, and Stochastic Processes, McGraw-Hill, New York, 1991.
[31] S.K. Mitra, Digital Signal Processing, McGraw-Hill, New York, 1998.