How useful randomness for cryptography can emerge from multicore-implemented complex networks of chaotic maps

Jean-Pierre Lozi, René Lozi, Oleg Garasym

▶ **To cite this version:**

Jean-Pierre Lozi, René Lozi, Oleg Garasym. How useful randomness for cryptography can emerge from multicore-implemented complex networks of chaotic maps. Journal of Difference Equations and Applications, 2017, 23 (5), pp.821-859. 10.1080/10236198.2017.1287176 . hal-01742568

HAL Id: hal-01742568
https://hal.science/hal-01742568

Submitted on 25 Mar 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Published in Journal of Difference Equations and Applications,
pp. 821-859, Vol. 23, Issue 5, 2017
http://dx.doi.org/10.1080/10236198.2017.1287176

Personal file

How useful randomness for cryptography can emerge from multicore-implemented complex networks of chaotic maps

Oleg Garasym[1], Jean-Pierre Lozi [2] and René Lozi*[3]

[1] SOC, IBM, Wroclaw, Poland; [2] Laboratoire I3S, UMR CNRS 7271, University Côte d' Azur, Nice, France;
[3] Laboratoire J. A. Dieudonné, UMR CNRS 7351, University Côte d' Azur, Nice, France

Received 09 Nov 20166, Accepted 16 January 2017
Published on line 15 Feb 2017

ABSTRACT

This article is devoted to the study of several topologies of complex networks of chaotic maps, in order to designe new Chaotic Pseudo Random Number Generators for cryptographic purpose in a bottom up approach of doing mathematics.
We examine topologies of special 2-D maps which are built combining piecewise linear and logistic maps. We present also a family of p-Dimensional networks whose we study numerically a particular realization, up to one hundred trillion iterates using multicores computers. From those topologies emerges useful randomness for cryptographic purpose.

KEYWORDS Chaotic map; randomness;ncryptography; PRNG; multicore

AMS SUBJECT CLASSIFICATIONS 65C10; 94A60; 37N30; 37D45

1. Introduction

Nowadays, complex networks exist everywhere. In the recent decade, they have been widely investigated partly due to their wide applications in many fields of science such as neural networks, ecosystems, the Internet, the WWW, electrical power grids, communication systems, etc., and partly due to their broad scientific progress in physics, mathematics, engineering, biology, etc.

In a new kind of bottom up demarche in mathematics, opposed to the top down approach of using mathematics for modelling, we introduced recently complex networks of chaotic maps in order to improve the growing research field of cryptography based chaos.

In this article, after recalling basics of chaotic maps and their use when coupled, we explore thoroughly some new topologies of complex networks of chaotic maps, showing up their excellent properties for cryptographic purpose. Cryptography based chaos needs reproducible pseudo random numbers. They are obtained coupling chaotic maps of different nature in special ways. Reproducibility is required in order to synchronize encryption and decryption processes.

Cryptography based chaos needs also secret keys to avoid any recovering of plain text messages by malicious intruders. Both reproducibility and secrecy in randomness are easily obtained by new topologies of complex networks we present in this article.

In Section 2 we recall some recent topologies of systems of piecewise linear chaotic maps and useful tools to assess the randomness of the iterated numbers they generate. In

* corresponding author: René Lozi: rlozi@unice.fr

Section 3 we examine mainly topologies of 2-D coupled map, highlighting good candidates of Chaos Pseudo Random Number Generators (CPRNG). In Section 4 we present a family of p-Dimensional networks whose we study numerically a particular realization, up to one hundred trillion iterations using multicore computers.

2. Piecewise linear chaotic map of order p

Efficient CPRNG have been recently introduced in [7]. The idea of applying discrete chaotic dynamical systems, intrinsically, exploits the property of extreme sensitivity of trajectories to small changes of initial conditions. They use the ultra weak multidimensional coupling of p 1-dimensional dynamical systems which preserve the chaotic properties of the continuous models in numerical experiments. The process of chaotic sampling and mixing of chaotic sequences, which is pivotal for these families, works perfectly in numerical simulation when floating point (or double precision) numbers are handled by a computer.

It is noteworthy that these families of very weakly coupled maps are more powerful than the usual formulas used to generate chaotic sequences mainly because only additions and multiplications are used in the computation process; no division being required. Moreover, the computations are done using floating point or double precision numbers, allowing the use of the powerful Floating Point Unit (FPU) of the modern microprocessors. In addition, a large part of the computations can be parallelized taking advantage of the multicore microprocessors which are of common use in laptop computers.

In this article, we are looking for chaotic systems which satisfy the required parameters properties (Table 1) for CPRNG design.

The alternate system with auto-coupling and ring-coupling (1) has been initially selected because it has sufficient randomness and high chaoticity [8]. It has been only studied in the chaotic range of parameters in the aim of applications to chaotic cryptography. Classical analysis in term of stability of fixed points, periodic points and bifurcations remains an open problem. The system successfully passed statistical and numerical tests such as: auto-correlation; cross-correlation; uniform distribution [5,7].

$$M_p : \begin{cases} x_{n+1}^{(1)} = 1 - 2|x_n^{(1)}| + k^1((1-e_1)x_n^{(2)} + e_1 x_n^{(1)}) \\ x_{n+1}^{(2)} = 1 - 2|x_n^{(2)}| + k^2((1-e_2)x_n^{(3)} + e_2 x_n^{(2)}) \\ \vdots \\ x_{n+1}^{(p)} = 1 - 2|x_n^{(p)}| + k^p((1-e_p)x_n^{(1)} + e_p x_n^{(p)}) \end{cases} \tag{1}$$

where the parameters $k^j = (-1)^{j+1}$, and $e_i \in]0,1[$ are the encryption keys, $x \in R^p$, $T^p = [-1,1]^p$ by the map $M_p = T^p \to T^p$. In addition, at each iteration a point must be checked and fed back to the torus $[-1,1]^p$ (Figure 1) applying the following conditions:

$$\begin{aligned} &\text{if } x_{n+1}^{(j)} < -1 \text{ then add 2} \\ &\text{if } x_{n+1}^{(j)} > 1 \text{ then subtract 2} \end{aligned} \tag{2}$$

The equations make system dynamics to run on the torus $[-1,1]^p$ and impact on the system complexity.

Table 1. Consolidated criteria of robust CPRNG.

No.	Criteria	Succeed characteristic
1	Largest Lyapunov exponent	Positive
2	Attractor in phase space	Dense everywhere
3	Attractor in phase delay	Dense everywhere
4	Topological mixing	Complex and fast
5	Uniform distribution	Decreasing of distribution error with increasing generated points
6	Auto-correlation	Near zero
7	Cross-correlation	Near zero
8	NIST tests	Successfully passed

The mapping M_p is defined in algorithmic way by (1) and (2), in order to highlight how it is built using the symmetric tent map of the interval $[-1, 1]$ together with the very weak coupling between components.

It can be defined in a more traditional mathematical fashion using modulo function:

$$M_p : \begin{cases} x_{n+1}^{(1)} = ((2 - 2|x_n^{(1)}| + k^1((1 - e_1)x_n^{(2)} + e_1 x_n^{(1)})) \bmod 1) - 1 \\ x_{n+1}^{(2)} = ((2 - 2|x_n^{(2)}| + k^2((1 - e_2)x_n^{(3)} + e_2 x_n^{(2)})) \bmod 1) - 1 \\ \vdots \\ x_{n+1}^{(p)} = ((2 - 2|x_n^{(p)}| + k^p((1 - e_p)x_n^{(1)} + e_p x_n^{(p)})) \bmod 1) - 1 \end{cases} \tag{3}$$

However, because this formula introduces unnecessary complexity for the reader, we continue to use the algorithmic way for the next mapping definitions.

2.1. Description of the Piecewise linear (PWL) chaotic map

The selected system with auto-coupling (the j-state impacts itself) and ring-coupling (the j-state impacts the $j + 1$-state) for information encryption (1) exhibits complex non-linear dynamics, has chaotic properties very similar to pseudo-random properties [11] on the p-dimensional torus. The key element of the system is the symmetric tent map $\bar{T}$ that is applied to every state on the torus $X \in [-1, 1]^p$ in Equation (6):

$$\bar{T}(X_n) = \begin{pmatrix} T(x_n^{(1)}) \\ T(x_n^{(2)}) \\ \vdots \\ T(x_n^{(p)}) \end{pmatrix} \tag{4}$$

where

$$T(x) = 1 - 2|x| \tag{5}$$

is the classical symmetric tent map defined from $[-1, 1]$ into $[-1, 1]$.

The p-order function $f: X_{n+1} = f(X_n)$ with $X_n = (x_n^{(1)}, x_n^{(2)}, \ldots, x_n^{(p)})$ can be rewritten from control point of view with output vector $Y_n = (y_n^{(1)}, y_n^{(2)}, \ldots, y_n^{(p)})$ as follows:

$$X_{n+1} = f(X_n) = A\bar{T}(X_n) \tag{6}$$

where $Y_n = CX_n$, C is the $1 \times p$ vector line and A is a $p \times p$ matrix defined as follows:

$$A = \begin{pmatrix} e_{1,1} = 1 - \sum_{j=2}^{j=p} e_{1,j} & e_{1,2} & \cdots & e_{1,p-1} & e_{1,p} \\ e_{2,1} & e_{2,2} = 1 - \sum_{j=1,j\neq 2}^{j=p} e_{2,j} & \cdots & e_{2,p-1} & e_{2,p} \\ \vdots & \ddots & & \vdots & \vdots \\ \vdots & & \ddots & \vdots & \vdots \\ e_{p,1} & \vdots & \vdots & e_{p,p-1} & e_{p,p} = 1 - \sum_{j=1}^{j=p-1} e_{p,j} \end{pmatrix}, \quad (7)$$

with parameters $e_{i,i} = 1 - \sum_{j=1,j\neq i}^{j=p} e_{i,j}$ on the diagonal (the matrix A is stochastic if the coupling parameters verify $e_{i,j} > 0$ for every i and j).

Parameters $e_{i,j}$ generalize coupling relationship of (1) and can be used also as encryption keys for chaotic cryptography.

Remark: In (1) we use a symmetric tent map, however any piecewise linear map with the same single humped shape can be used as well.

The weakly coupled maps are able to generate significantly better pseudo-random sequences than 1-D maps due to the size of the phase space which is in higher dimension avoiding therefore short periodic sequences. The advantages are that only additions and multiplications are used in the map, influencing the speed performance. Moreover, the coupled map exhibits a high number of parameters ($p \times (p-1)$) for p coupled equations) ensuring reliable cypher-keys, when used in chaos-based cryptosystems. The parameters are very sensitive to any changes [8].

2.2. Uniform distribution

An excellent PRNG looks like truly random, when it is unpredictable and there is no correlation between points which have an equal probability. If the generator is capable to produce the sequences uniformly distributed in the phase space and the phase delay, then the system behaviour is like truly random.

There are different tools to analyse points distribution, i.e. histogram, cumulative distribution, etc. However, they give very general information. In order to assess numerical computations more accurately and to study qualitatively the chaotic systems an approximation density function [7] is preferable to the previously mentioned tools. The approximation $P_{M,N}(x)$ of the density is defined by the invariant measure (the probability distribution function) [16] corresponding to the 1-dimensional map f going from the interval $\mathbb{J} \subset \mathbb{R}$ into itself ($\mathbb{J} = [-1, 1]$), when computed with floating numbers. The regular partition into M small intervals (boxes) r_i of $\mathbb{J}$ is defined by

$$\mathbb{J} = \bigcup_{i=0}^{M-1} r_i \quad (8)$$

where

$$r_i = [s_i, s_{i+1}[\quad i = 0,\ M-2 \quad \text{and} \quad r_{M-1} = [s_{M-1}, 1] \tag{9}$$

and

$$s_i = -1 + \frac{2i}{M}, \quad i = 0, \ldots, M \tag{10}$$

One can remark that each box has a length which is equal to $\frac{2}{M}$. The first iterates of the transient regime are discarded, the following are collected. At the end of the computation when N iterates are computed, the value $\sharp r_i$ is the number of iterates belonging to the interval r_i. The numbers $P_N(s_i)$ represent the ratio of the number of iterates with $\frac{N}{M}$ for each box r_i. The approximated $P_N(s_i)$ is a M steps function. As M is parameter we write it

$$P_{M,N}(s_i) = \frac{M}{N}(\sharp r_i) \tag{11}$$

$P_{M,N}(x)$ is normalized to 2 on the interval $\mathbb{J} = [-1, 1]$.

$$P_{M,N}(x) = P_{M,N}(s_i), \quad \forall x \in r_i \tag{12}$$

In the case of a multidimensional p-coupled system it is important to check the distribution of iterates for each component $x^1, x^2, \ldots, x^p$ of $X \subset \mathbb{J}^p$. The approximated probability distribution function, $P_{M,N}(x^j)$ is associated to one among the components x^j of the analysed system. We denote M_{disc} instead of M and N_{iter} instead of N, as often as they are more meaningful.

The discrepancies E_1 (in norm L_1), E_2 (in norm L_2) and E_∞ (in norm L_∞) between $P_{M_{disc},N_{iter}}(x^j)$ (where M_{disc} is the number of boxes, N_{iter} is the number of iterations) and the Lebesgue measure, which is the invariant measure of the uniform randomly distributed process are defined by

$$E_{1,M_{disc},N_{iter}}(x^j) = \|P_{M_{disc},N_{iter}}(x^j) - 0.5\|_{L_1} \tag{13}$$

$$E_{2,M_{disc},N_{iter}}(x^j) = \|P_{M_{disc},N_{iter}}(x^j) - 0.5\|_{L_2} \tag{14}$$

$$E_{\infty,M_{disc},N_{iter}}(x^j) = \|P_{M_{disc},N_{iter}}(x^j) - 0.5\|_{L_\infty} \tag{15}$$

The numerical calculation of the uniform distribution allows us to judge about the system unpredictability. We know that $E_1 < E_2 < E_\infty$, however the computation of errors with three norms gives a more precise view of uniform distribution of iterates. It allows also to check that computations are not flawed.

In the same way an approximation of the correlation distribution function $C_{M,N}(x, y)$ is obtained numerically, building a regular partition of M^2 small squares (boxes) of J^2

embedded in the phase subspace (x^l, x^m)

$$s_i = -1 + \frac{2i}{M}, \quad t_j = -1 + \frac{2i}{M}, \quad i,j = 0, M \tag{16}$$

$$r_{i,j} = [s_i, s_{i+1}[\times [t_j, t_{j+1}[, \quad i,j = 0, M-2 \tag{17}$$

$$r_{M-i,j} = [s_{M-1}, 1] \times [t_j, t_{j+1}[, \quad j = 0, M-2 \tag{18}$$

$$r_{i,M-1} = [s_i, s_{i+1}[\times [t_{M-1}, 1], \quad j = 0, M-2 \tag{19}$$

$$r_{M-1,M-1} = [s_{M-1}, 1] \times [t_{M-1}, 1] \tag{20}$$

the measure of the area of each box is

$$(s_{i+1} - s_i) \times (t_{i+1} - t_i) = \left(\frac{2}{M}\right)^2 \tag{21}$$

Once $N + Q$ iterated points (x_n^l, x_n^m) belonging to these boxes are collected, the number of iterates divided by $\frac{N}{M^2}$ collected in each box $r_{i,j}$ is the value $C_N(s_i, t_j)$ The approximated probability distribution function $C_N(x, y)$ defined here is then a 2-dimensional step function, with M^2 steps. As M can take several values in the next sections, we define

$$C_{M,N}(S_i, t_j) = \frac{1}{4}\frac{M^2}{N}(\#r_{i,j}) \tag{22}$$

where $\#r_{i,j}$ is the number of iterates belonging to the square $r_{i,j}$ and the constant 1/4 allows the normalization of $C_{M,N}(x, y)$ on the square J^2 .

$$C_{M,N}(x, y) = C_{M,N}(s, t) \quad \forall (x, y) \in r_{i,j} \tag{23}$$

The discrepancies E_{C_1} (in norm L_1), E_{C_2} (in norm L_2) and E_{C_∞} (in norm L_∞) between $C_{M_{disc},N_{iter}}(x, y)$ and the uniform distribution on the square, are defined by

$$E_{C_1,M_{disc},N_{iter}}(x, y) = \|C_{M_{disc},N_{iter}}(x, y) - 0.25\|_{L_1} \tag{24}$$

$$E_{C_2,M_{disc},N_{iter}}(x, y) = \|C_{M_{disc},N_{iter}}(x, y) - 0.25\|_{L_2} \tag{25}$$

$$E_{C_\infty,M_{disc},N_{iter}}(x, y) = \|C_{M_{disc},N_{iter}}(x, y) - 0.25\|_{L_\infty} \tag{26}$$

Finally let $AC_{M_{disc},N_{iter}}(x, y)$ be the autocorrelation distribution function which is the correlation function $C_{M_{disc},N_{iter}}(x, y)$ of (23) defined in the phase space (x_n^l, x_{n+1}^l) instead of the phase space (x^l, x^m) . We define the corresponding discrepancies $E_{AC_1}, E_{AC_2}, E_{AC_\infty}$. We make the same remark as previously done for the use of thee different norms instead of one only.

2.3. Injection mechanism for system on torus

As mentioned before, from the initial condition on the torus $[-1, 1]^p$ the trajectory will quickly leave the torus if the function (1) is applied alone. To keep system trajectories on

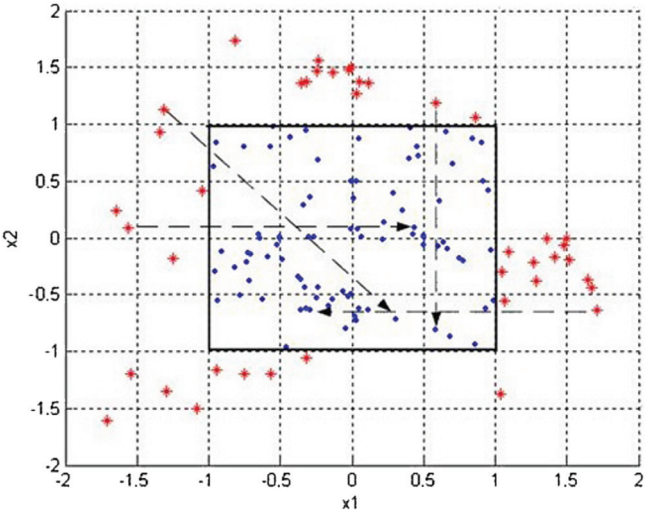

Figure 1. Application of the injection mechanism (27) in order to move the points initially belonging to the torus $[-2,2]^2$ towards the torus $[-1,1]^2$. Four examples of motions indicated by the displayed arrows.

the torus $[-1,1]^p$, we apply the injection mechanism (27):

$$\begin{aligned} &\text{if } x_{n+1}^{(j)} < -1 \quad \text{then add } 2 \\ &\text{if } x_{n+1}^{(j)} > 1 \quad \text{then subtract } 2 \end{aligned} \tag{27}$$

Let us now show how the mechanism works if some points go out of the torus (Figure 1). The unstable system trajectories (in red), which are escaping, are forced to go back to the torus $T^p = [-1,1]^p$ (in blue). By this operation, the system dynamics has been maintained within $[-1,1]^p$. In addition, by this operation the resulting system exhibits more complex dynamics with additional nonlinearity, which is advantageous for chaotic encryption (since it improves the security).

The confinement from torus $[-2,2]^p$ to torus $[-1,1]^p$ of the dynamics is shown in (Figure 2): dynamics crosses from the negative region (blue colored) to the positive one, and conversely, if the points stand in the positive regions (red colored). The maximal torus where points mapped by (1) could belong is $[-2,2]^p$.

Auto and ring-coupling between states (Figure 3) of the system and injection mechanism influence the system dynamics making it attractive to cryptography.

The system behaviour (1) becomes more complex when increasing its order. The analysis of the 2-D order system showed a potential weakness. The careful distribution analysis has been performed using an approximation $P_{M,N}(x)$ (11) that is defined by the invariant measure. The resulting picture (Figure 4) displays regions where the density is lower, and others where it is more concentrated. Thus, points distribution is not uniform. The space $[-1,1]^2$ has been divided into 200×200 boxes and in each of them the points probability is calculated. On the graph (Figure 4) the boxes with low probability are blue colored and with high one are red colored, other colors indicate the probability in-between.

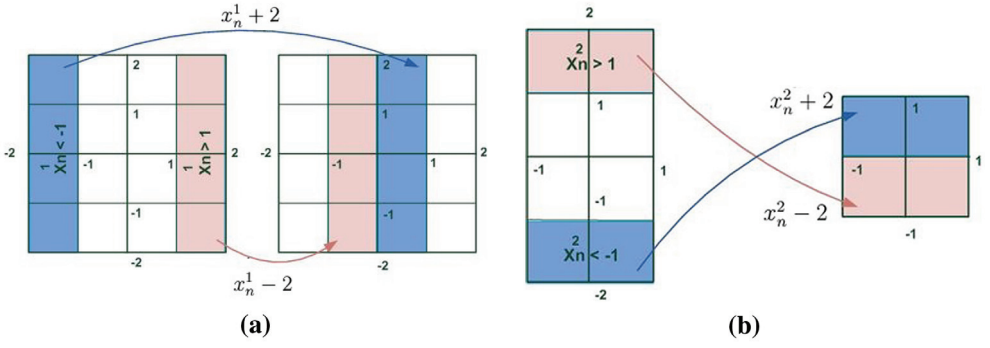

Figure 2. Injection mechanism of the trajectories feed back to the torus ($[-2,2]^2 \Rightarrow [-1,1]^2$) (a) if $x_n^{(1)} > 1$ then $x_n^{(1)} - 2$ or if $x_n^{(1)} < -1$ then $x_n^{(1)} + 2$ (b) if $x_n^{(2)} > 1$ then $x_n^{(2)} - 2$ or if $x_n^{(2)} < -1$ then $x_n^{(2)} + 2$.

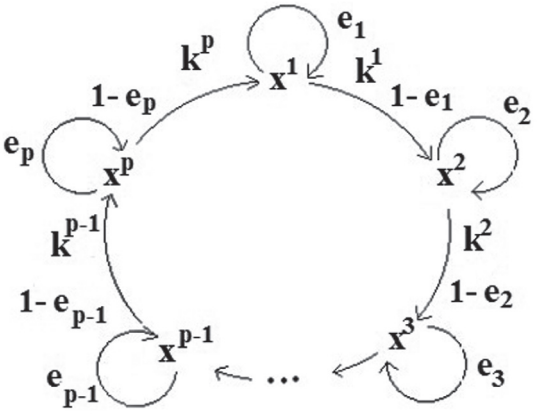

Figure 3. Auto and ring-coupling between states of system (1).

Therefore, only 3-D or 4-D realizations of system (1) are useful. In the next section we explore 2-D topologies which are more efficient than (1).

The geometrical shape of the region with different density of iterates [11] can be easily deduced when using critical lines CL [14]. The critical lines are singularities of dimension 1, they were introduced by C. Mira fifty years ago. They are a very efficient tool for the analysis of non invertible maps.

Note that small (i.e. weak coupling) parameters e_j in (1) are required to guarantee satisfactory topological mixing.

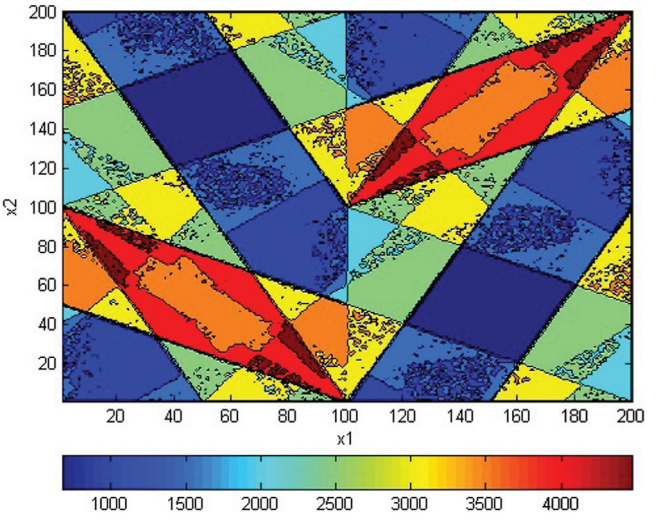

Figure 4. Approximate density function of (1).

3. Exploring topologies of coupled chaotic maps

Both logistic (28) and tent (29) maps have been explored for many years in the hope of generating pseudo-random numbers [13,18].

$$f_\mu(x_n) \equiv L_\mu(x) = 1 - \mu x^2 \tag{28}$$

$$f_\mu(x_n) \equiv T_\mu(x) = 1 - \mu|x| \tag{29}$$

Remark: for the sake of simplicity we have defined $T_2(x)$ as $T(x)$ in Section 2.1 (5).

Some authors proposed to use 1-D chaotic dynamical systems as a base of a cryptosystem [2,3].

Both mapping are topologically conjugate for some parameters [1], therefore their topological properties (distribution, chaoticity, etc.) are in some sense similar however due to the structure of numbers of the computers, their numerically behaviour differs dramatically from the theoretical expectations when the iterates are obtained using a computer. The symmetric tent map is drastically numerical unstable: Sharkovskii's theorem applies for it [17]. When $\mu = 2$ there exists a period three orbit and therefore an infinity of periodic orbits exist too. Nevertheless the orbit of almost every initial point of the interval $[1, -1]$ of the descretized tent map eventually wind up to the stable fixed point $x = -1$, and these is no numerical attracting periodic orbit. This behaviour is called the collapse of iterates [9]. The numerical behaviour of the iterates is not at all chaotic. It is worse than the numerical behaviour of the logistic map when chaos is assessed. Many papers have been published on collapsing effect [4].

Some interesting theoretical studies of chaotic dynamical systems on finite sets have also been published [10]. Therefore, the original idea we introduce in this article, is to combine properties of the tent map (T_μ) with properties of the logistic one (L_μ) to build

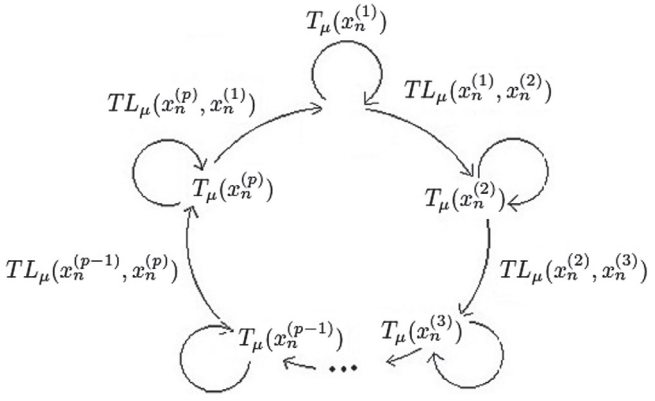

Figure 5. Auto and ring-coupling between states of the T_μ with TL_μ maps.

a new map showing enhanced properties, through combination in several topologies of the network.

Thus, our proposition has the form:

$$f_\mu(x) \equiv TL_\mu(x) \equiv L_\mu(x) - T_\mu(x) = \mu|x| - \mu x^2 = \mu(|x| - x^2) \tag{30}$$

which presents the advantage to inverse the shape of the graph of the tent map T_μ combined with the graph of the logistic map L_μ.

When some maps as logistic and tent are used alone in cryptography, due to the collapsing effect [6,20] they show very weak security. Thus, multidimensional maps are often applied to construct PRNG [15,19]. The system (1) provides a new method to enhance chaotic properties of the tent map thanks to the coupling and under sampling. In another way, we propose to couple T_μ map with TL_μ map (30) (Figure 5). When used in such context, the TL_μ map can be seen as a map of two variables:

$$TL_\mu(x^{(p)}, x^{(q)}) = \mu\left(|x^{(p)}| - (x^{(q)})^2\right) \tag{31}$$

Because two variables are involved in the definition of TL_μ, it is therefore convenient and easy to define a new mapping $M_{\mu,p}$ from $[-1,1]^p \to [-1,1]^p$

$$M_{\mu,p}\begin{pmatrix} x_n^{(1)} \\ x_n^{(2)} \\ \vdots \\ x_n^{(p)} \end{pmatrix} = \begin{pmatrix} x_{n+1}^{(1)} \\ x_{n+1}^{(2)} \\ \vdots \\ x_{n+1}^{(p)} \end{pmatrix} = \begin{cases} T_\mu(x_n^{(1)}) + TL_\mu(x_n^{(1)}, x_n^{(2)}) \\ T_\mu(x_n^{(2)}) + TL_\mu(x_n^{(2)}, x_n^{(3)}) \\ \vdots \\ T_\mu(x_n^{(p)}) + TL_\mu(x_n^{(p)}, x_n^{(1)}) \end{cases} \tag{32}$$

Due to its structure this multidimensional mapping is unstable and the trajectories are mapped outside the initial torus. Therefore, in order to keep bounded in $[-1,1]^p$ the

iterates, we use the quasi similar to (27) injection mechanism:

$$\begin{aligned} &\text{if } x_{n+1}^{(i)} < -1 \\ &\quad \text{then add } 2 \\ &\text{if } x_{n+1}^{(i)} > 1 \\ &\quad \text{then subtract } 2 \end{aligned} \tag{33}$$

which allows for $1 \leq i \leq p$, points to be sent from $[-3,3]^p$ to $[-1,1]^p$.

The combined use of T_μ and TL_μ mapping entangles system states. The resulting mapping possesses an attractor because it performs in the same time contraction and stretching of the distance between states. Moreover, this forced motion improves its mixing properties. Therefore, TL_μ function is a powerful tool to modify dynamics.

The coupling of those simple maps is good for achieving complexity, because:

- Simple states interact with the whole dynamics.
- The states interaction has a global mixing effect.

Therefore, if we use TL_μ instead of simple tent or logistic maps in order to improve their properties of complexity we obtain an excellent effect on chaoticity. Randomness could be achieved in this way. The proposed function improves the complexity of a simple map. The question is how to study the obtained system. Poincaré was one of the first who used graphical analysis of the complex systems. We will use as well graphical approach to study the new chaotic systems, but also, theoretical functions involved in our study.

Note that the system (32) can be seen in the scope of a more general point of view, introducing constants k^i which generalize considered topologies. It is called alternate if $k^i = (-1)^i$, $1 \leq i \leq p$, or non-alternate if $k^i = +1$ or if $k^i = (-1)^{i+1}$, $1 \leq i \leq p$; or $k^i = -1$, $1 \leq i \leq p$. It can be a mix of alternate and non-alternate if $k^i = +1$ or -1 randomly. Let $\underline{k} = (k^1, k^2, \ldots, k^p)$

$$M_{\mu,p}^{\underline{k}} \begin{pmatrix} x_n^{(1)} \\ x_n^{(2)} \\ \vdots \\ x_n^{(p)} \end{pmatrix} = \begin{pmatrix} x_{n+1}^{(1)} \\ x_{n+1}^{(2)} \\ \vdots \\ x_{n+1}^{(p)} \end{pmatrix} = \begin{cases} T_\mu(x_n^{(1)}) + k^1 \times TL_\mu(x_n^{(i)}, x_n^{(j)}), & i,j = (1,2) \text{ or } (2,1) \\ T_\mu(x_n^{(2)}) + k^2 \times TL_\mu(x_n^{(i)}, x_n^{(j)}), & i,j = (2,3) \text{ or } (3,2) \\ \vdots \\ T_\mu(x_n^{(p)}) + k^p \times TL_\mu(x_n^{(i)}, x_n^{(j)}), & i,j = (p,1) \text{ or } (1,p) \end{cases} \tag{34}$$

In this article we will consider only multidimensional mapping possessing the best properties for CPRNG design.

3.1. 2-D topologies

One of the first aim assigned to our new CPRNG design is to obtain near perfect uniform distribution, the other goals are to obtain a CPRNG.

The general form of the new considered 2-D map is as follows:

$$M_{\mu,2}^{\underline{k}} \begin{pmatrix} x_n^{(1)} \\ x_n^{(2)} \end{pmatrix} = \begin{pmatrix} x_{n+1}^{(1)} \\ x_{n+1}^{(2)} \end{pmatrix} = \begin{cases} T_\mu(x_n^{(1)}) + k^1 \times TL_\mu((x^{(i)}, x^{(j)})) \\ T_\mu(x_n^{(2)}) + k^2 \times TL_\mu((x^{(i')}, x^{(j')})) \end{cases} \tag{35}$$

Table 2. The sixteen maps defined by Equation (35).

Case	k^1	k^2	i	j	i'	j'
#1	+1	+1	1	2	1	2
#2	+1	−1	1	2	1	2
#3	−1	+1	1	2	1	2
#4	−1	−1	1	2	1	2
#5	+1	+1	2	1	2	1
#6	+1	−1	2	1	2	1
#7	−1	+1	2	1	2	1
#8	−1	−1	2	1	2	1
#9	+1	+1	1	2	2	1
#10	+1	−1	1	2	2	1
#11	−1	+1	1	2	2	1
#12	−1	−1	1	2	2	1
#13	+1	+1	2	1	1	2
#14	+1	−1	2	1	1	2
#15	−1	+1	2	1	1	2
#16	−1	−1	2	1	1	2

with $i, j, i', j' = 1$ or 2, $i \neq j$ and $i' \neq j'$. Considering the above conditions, it is possible to define 16 different maps (see Table 2):

Therefore, we will consider only two 2-D systems: $TTL_\mu^{RC}(x_n^{(2)}, x_n^{(1)})$ **non-alternate** (case #13):

$$TTL_\mu^{RC}: \begin{cases} x_{n+1}^{(1)} = 1 - \mu|x_n^{(1)}| + \mu(|x_n^{(2)}| - (x_n^{(1)})^2) = T_\mu(x_n^{(1)}) + TL_\mu(x_n^{(2)}, x_n^{(1)}) \\ x_{n+1}^{(2)} = 1 - \mu|x_n^{(2)}| + \mu(|x_n^{(1)}| - (x_n^{(2)})^2) = T_\mu(x_n^{(2)}) + TL_\mu(x_n^{(1)}, x_n^{(2)}) \end{cases} \tag{36}$$

and $TTL_\mu^{SC}(x_n^{(1)}, x_n^{(2)})$ **alternate** (case #3):

$$TTL_\mu^{SC}: \begin{cases} x_{n+1}^{(1)} = 1 - \mu|x_n^{(1)}| - \mu(|x_n^{(1)}| - (x_n^{(2)})^2) = T_\mu(x_n^{(1)}) - TL_\mu(x_n^{(1)}, x_n^{(2)}) \\ x_{n+1}^{(2)} = 1 - \mu|x_n^{(2)}| + \mu(|x_n^{(1)}| - (x_n^{(2)})^2) = T_\mu(x_n^{(2)}) + TL_\mu(x_n^{(1)}, x_n^{(2)}) \end{cases} \tag{37}$$

We have chosen systems #3 and #13 because when assessed with respect to the uniform distribution of iterates they give the best results due to balanced processes between contraction and extension.

3.2. Study of randomness properties of both TTL_μ^{RC} and TTL_μ^{SC}

Using numerical computations we assess the randomness properties of both 2-dimensional maps. When all requirements 1–8 on Figure 6 are satisfied the dynamical systems associated to those mapping can be considered as pseudo-random and their application to cryptosystems is possible.

Whenever one among the eight criteria is not satisfied those mapping generate a less randomly behaviour.

The chaotic behaviour of each map depends essentially on the 'control' parameter μ. When the phase portrait is globally observed, the analyse does not depend on the initial guess x_0. Therefore, a bifurcation diagram is an appropriate tool to study the dependency of parameter μ.

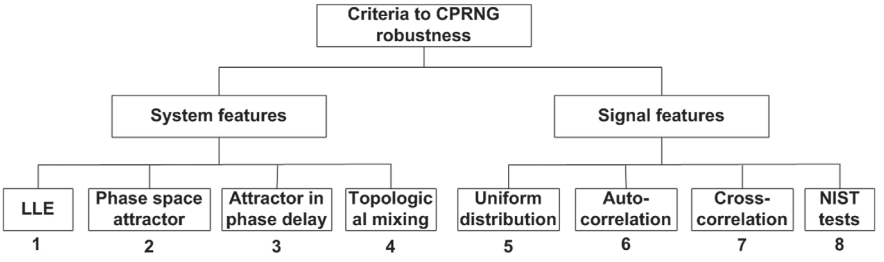

Figure 6. The main criteria for PRNG robustness.

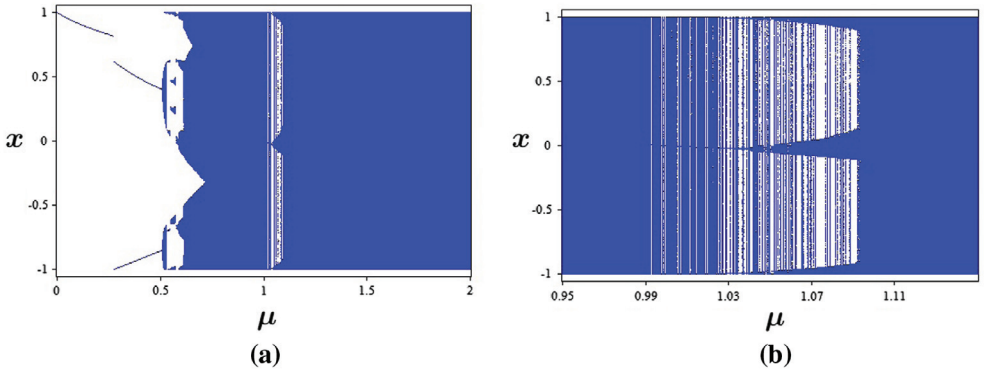

Figure 7. (a) Bifurcation diagram of 2-D new map: TTL_{μ}^{RC} non-alternate (36) (b) Magnification of the bifurcation diagram of TTL_{μ}^{RC} non-alternate.

In order to compute this diagram, for every value of μ, an initial value x_0 is arbitrary selected. The map is iterated many times, however the first iterates are discarded, in order to avoid the plotting of a transient regime. The next points are plotted. Afterwards, the process is repeated incrementing slightly μ.

The bifurcation diagram of both 2-dimensional mappings TTL_{μ}^{RC} non-alternate (Figure 7) and TTL_{μ}^{SC} alternate (Figure 8) is built computing 10,000 points for each initial guess, the first 1000 of which are discarded. Hence, for every value of the parameter μ, we plot 9000 points. We observe more or less the same patterns in both graphs of components $x^{(1)}$ and $x^{(2)}$.

For μ belonging to the interval $[0, 0.25]$ a fixed point (i.e. a period 1) is observed. After the greatest value of the parameter interval, the steady state fixed point is transformed in period 2 orbit via a pitchfork bifurcation. This bifurcation is followed by a cascade of bifurcation. When μ increases the dynamical system is eventually chaotic, even if some periodic windows exist in the neighbourhood of $\mu = 1.1$ (Figure 7). The subsequent interval of μ shows chaotic dynamics.

Diagrams of bifurcation are often used for studying the global behaviour of nonlinear maps because they show in a glimpse the dynamics. They are very useful when used with the graph of the Lypunov exponent, it is very simple to determine if a dynamics is chaotic or not.

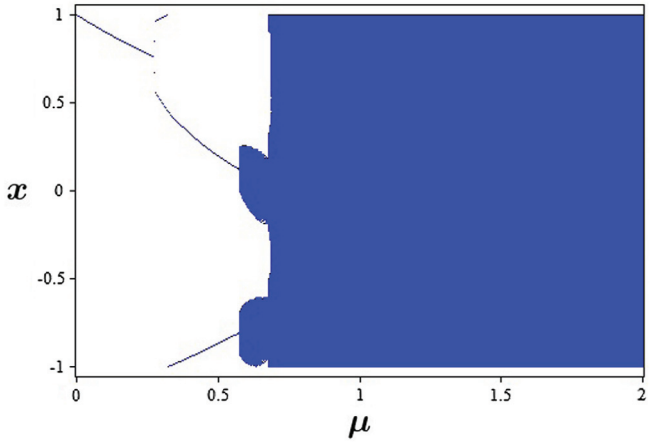

Figure 8. Bifurcation diagram of 2-D new map: TTL_{μ}^{SC} alternate (37).

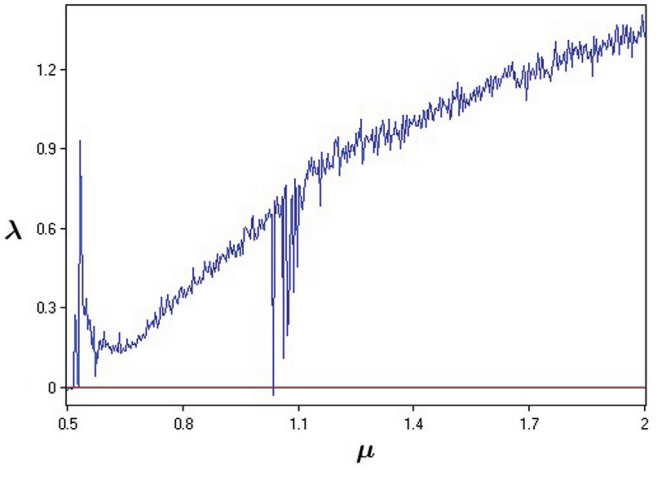

Figure 9. Largest Lyapunov exponent for 2-D TTL_{μ}^{RC} non-alternate map (36).

On Figures 9 and 10 we display respectively the Lyapunov exponent of TTL_{μ}^{RC} non-alternate, and TTL_{μ}^{SC} alternate. We use 10,000 iterations in order to compute such exponents. The range of parameters μ belongs between $\mu = 0.5$ to $\mu = 2$. The value of μ is plotted on the horizontal axis meanwhile the Lyapunov exponent λ is plotted with respect to the scale of the vertical axis.

The Lyapunov exponent plotted on those figures correspond exactly with the expected dynamics given by the bifurcation diagram. The positive value of the Lyapunov exponent indicates chaotic dynamics which increases with the increase of μ demonstrating the strongest chaos for $\mu = 2$.

The study highlights that both TTL_{μ}^{RC} non-alternate (Figure 9) and TTL_{μ}^{SC} alternate (Figure 10) maps exhibit a strong chaotic behaviour when $\mu = 2$, therefore we will only consider in the following this value for the parameter. We can see, checking the graphs

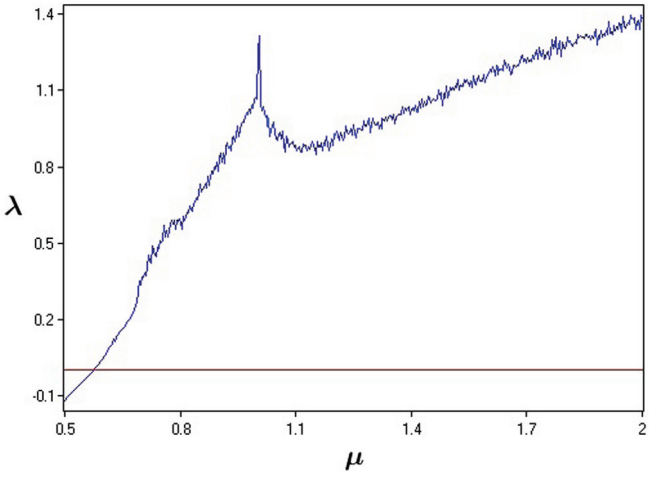

Figure 10. Largest Lyapunov exponent for 2-D TTL_{μ}^{SC} alternate map (37).

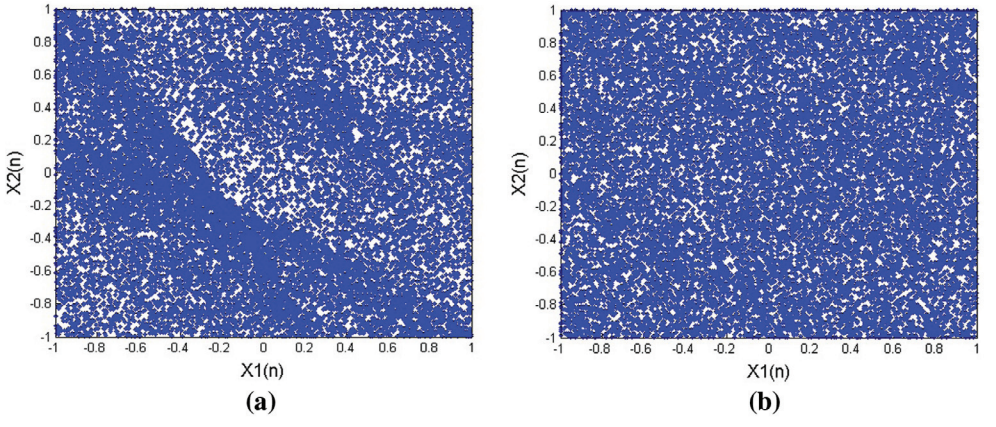

Figure 11. Phase space attractor of 2-D new maps, 2×10^4 points are generated (a) TTL_2^{RC} non-alternate (36) (b) TTL_2^{SC} alternate (37).

that for each initial point chosen, the trajectories starting from such points look chaotic. Hence, we can study an attractor in the phase space and the phase delay.

In the phase space we plot the iterates in the system $x_n^{(1)}$ vs. $x_n^{(2)}$ of coordinates in order to analyze the density of the points distribution. Depending on such analysis it is possible to assess the complexity of the behaviour of dynamics, noticing any weakness or inferring on the randomness nature of it.

The plot of the sequence of iterates is done using 2×10^4 points obtained by computing 3×10^4 points and deleting the first 10^4 points to avoid any plot of the transient regime.

The graphs of the attractor in phase space for TTL_2^{RC} non-alternate (Figure 11(a)) and TTL_2^{SC} alternate (Figure 11(b)) maps are quite different. The first one has well-scattered

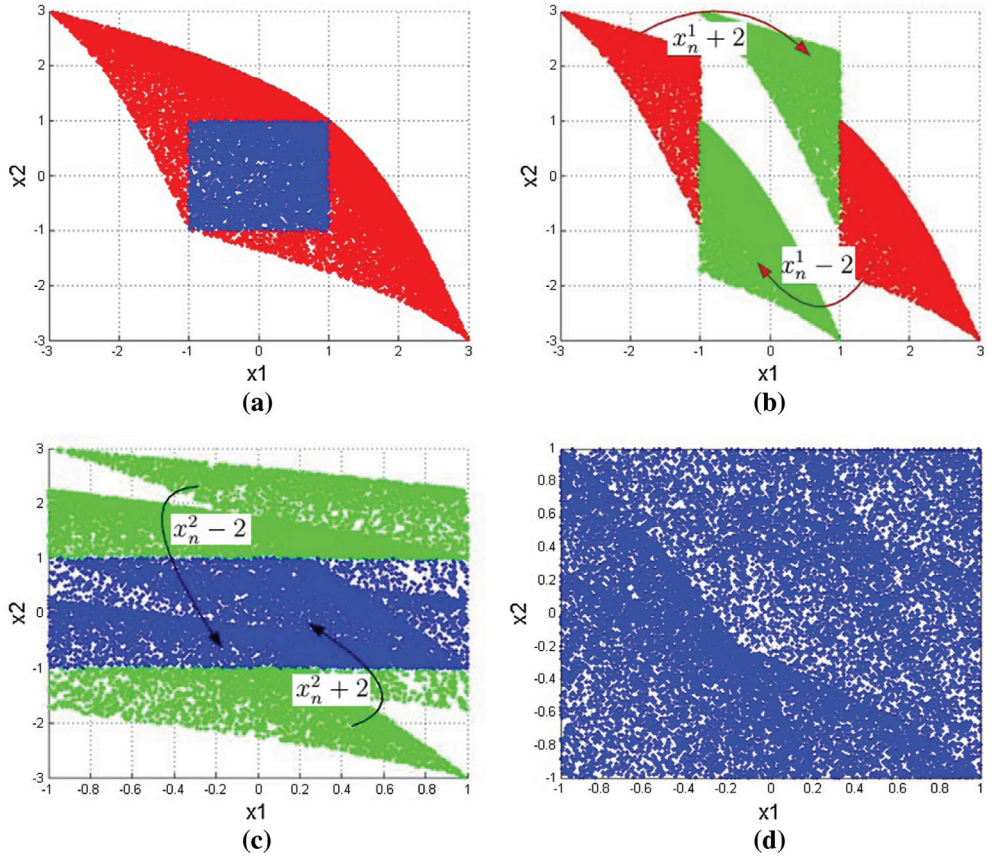

Figure 12. Injection mechanism $[-3,3]^2 \Rightarrow [-1,1]^2$ for TTL_2^{RC} non-alternate map (a) 2-D chaotic map without addition/subtraction (b) injection $x_n^{(1)}$ to the torus $[-1,1]^2$ (c) injection $x_n^{(2)}$ to the torus $[-1,1]^2$ (d) results after passing injection mechanism.

points on the whole pattern, but there are some more 'concentrated' regions forming curves on the graph. We will search an answer to the questions: 'Why there are more concentrated regions? From where curves are created?', by considering the injection mechanism.

The dynamics send the iterates between a mixed values equal to $+3$ and a minimal value equal -3. Equation (33) allows to send those points back to the initial square, however their influence to the dynamics works differently as in the system (1) [7]. Among 2×10^4 points generated on those square, 77 % are scattered out of the square $[-1,1]^2$. The multidimensional mechanism consists of p-steps. In each step, when the dynamics is going outbounds the value 2 is added or subtracted to the variables (33).

For the first step 69 % of the points are re-injected to the interval $[-1,1]^2$ (Figure 12(b)). After the second step (Figure 12(c)) all points are displaced from the extended range $[-3,3]^2$ into the torus $[-1,1]^2$ (Figure 12(d)). Therefore, the injection mechanism improves the non-linearity to the behaviour of the iterates, rendering the system more complex which can be seen as an advantage in case of cryptographic usage, from the security point of view.

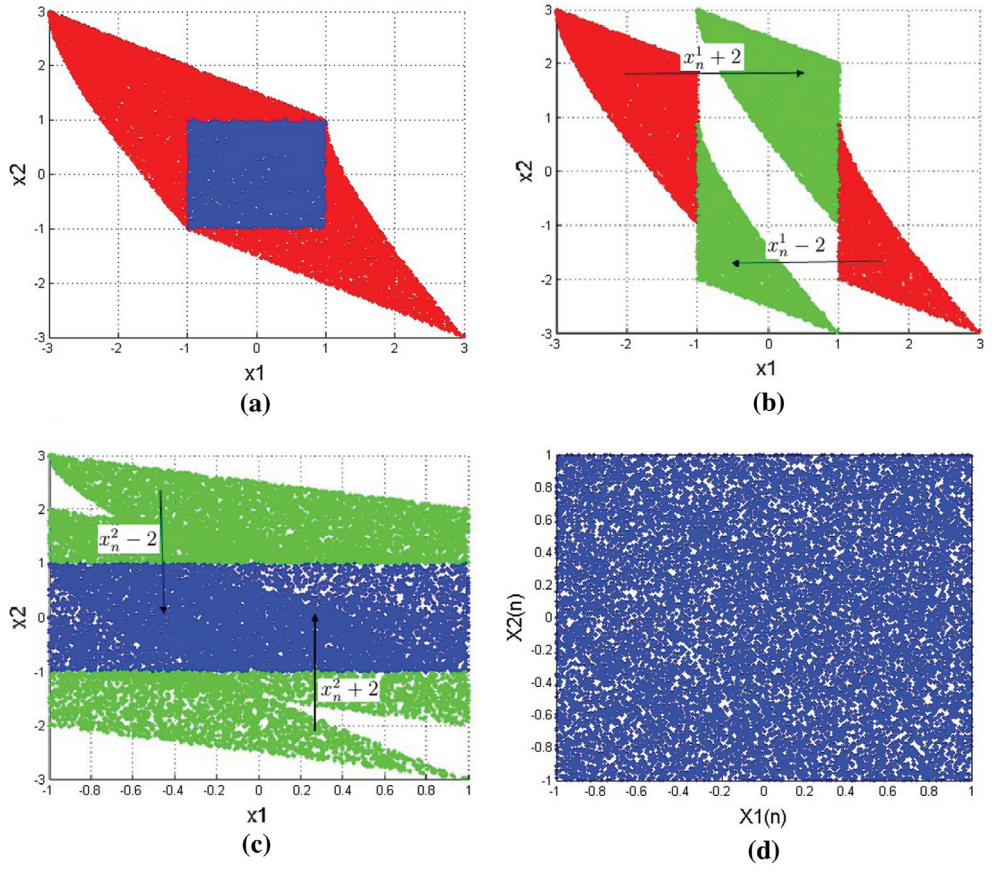

Figure 13. Injection mechanism $[-3,3]^2 \Rightarrow [-1,1]^2$ for TTL_2^{SC} alternate map (a) 2-D chaotic map without addition/subtraction (b) injection $x_n^{(1)}$ to the torus $[-1,1]^2$ (c) injection $x_n^{(2)}$ to the torus $[-1,1]^2$ (d) results after passing injection mechanism.

One can note that in the phase space, the attractor of TTL_2^{SC} alternate map may look uniformly distributed on the whole square, without any concentration of points in special region (11(b)). The injection mechanism works for uniformising the points distribution as shown in Figure 13.

When those PRNG are used for cryptographic purpose, the quality of the cryptosystem mostly rely to the randomness quality of the PRNG. Thus, the uniform generation by its dynamics is one of the most important criterion in order to build a robust PRNG. (Criterion 5, Figure 6).

An approximated invariant measure (11) assesses better then a visual examination of the picture of iterates: the density probability. It is hence used in the aim of providing a precise floating points distribution study. Using such a tool, a better uniform distribution can be reached.

The graph of the function value allows to compare the distribution of points between regions of the graph. The length of each side box is measured by the variable *step*. In other

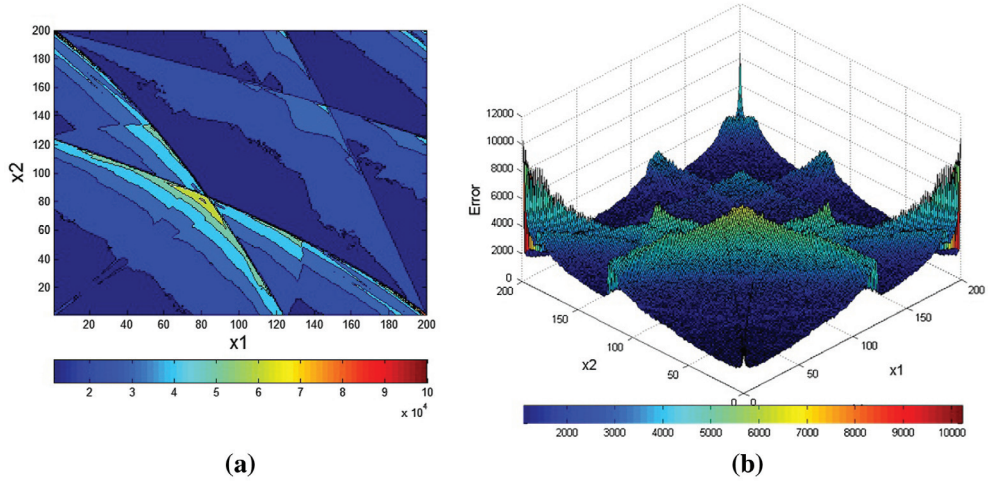

Figure 14. Approximate density function of TTL_2^{RC} non-alternate mapping, where *step* variable is fixed to 0.01. The plotting displays 10^9 points.

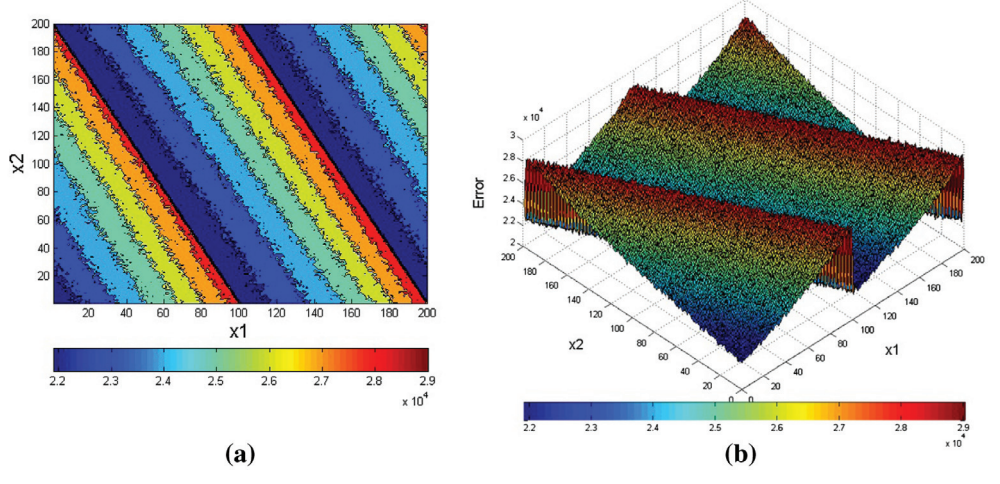

Figure 15. (a) Plot of the projection on the plane (x_1, x_2) of the approximate function of density of TTL_2^{SC} alternate map, where 10^9 points are computed. The variable *step* is fixed to 0, 01. (b) Plot of the approximate function.

words, the space is divided into a grid of $boxes[i, j]$ of area size equal to $step^2$. Then during the generation of iterates, the number of points which are sent to the $box[i, j]$ is counted.

In order to compute the approximate density function the square was divided using a grid of 200×200 boxes, i.e. *step* is fixed to 0.01. Then 10^9 iterates were computed. This number of iterates is the maximum possible to compute the approximate function using only a laptop computer in a reasonable time. The details of the points distribution are displayed on Figures 14 and 15. It is obvious that both multidimensional mapping do not give satisfactory uniform distribution in the phase space. However, it is noticed that some

parts of the graph (13(b)) are perfectly joint regions. This leads to a new idea in order to improve the regularity of the density modifying slightly (37).

3.3. A new 2-D chaotic PRNG

Taking into account the results of Section 3.2. the randomness result observed for 2-D topology could be enhanced changing slightly the coupling. In Figure 13(b) both particular regions: top-green and right-red ones have special shape of boundary that can be matched in a pretty way. In this aim we write the mapping TTL_{μ}^{SC} alternate (37) where $\mu = 2$ as follows:

$$TTL_2^{SC}(x_n^{(1)}, x_n^{(2)}) = \begin{cases} x_{n+1}^{(1)} = 1 + 2(x_n^{(2)})^2 - 4|x_n^{(1)}| \\ x_{n+1}^{(2)} = 1 - 2(x_n^{(2)})^2 + 2(|x_n^{(1)}| - |x_n^{(2)}|) \end{cases} \tag{38}$$

The first problem is that the top green colored region occurs after injection is applied. Thus, we develop the system (38) in such way that the green colored region 'stays' in such a position without the injection mechanism. Secondly, we need to reduce the width of the region. Evidently, it is possible to achieve this need by reducing the impact of the state $x^{(1)}$, with the new following map:

$$MTTL_2^{SC}(x_n^{(1)}, x_n^{(2)}) = \begin{cases} x_{n+1}^{(1)} = 1 + 2(x_n^{(2)})^2 - 2|x_n^{(1)}| \\ x_{n+1}^{(2)} = 1 - 2(x_n^{(2)})^2 + 2(|x_n^{(1)}| - |x_n^{(2)}|) \end{cases} \tag{39}$$

and the injection mechanism (33) is used as well, but restricted to 3 phases:

$$\begin{aligned} &\text{if } x_{n+1}^{(1)} > 1 \text{ then subtract } 2 \\ &\text{if } x_{n+1}^{(2)} < -1 \text{ then add } 2 \\ &\text{if } x_{n+1}^{(2)} > 1 \text{ then subtract } 2 \end{aligned} \tag{40}$$

Figures 16–18 show the result of the modifications highlighting that the 3-phase mechanism (Figure 16) of injection works very well. The techniques we have introduced in this section improve greatly the uniformity of the density of iterates in the phase space (Figures 17, 18).

Moreover, using a multicore computer, we were able to compute this distribution up to 10^{14} points (Figure 19, Table 3). Those computation results show excellent decreasing distribution errors with respect to the number of iterates. The Lyapunov exponent worths 0.5905 which is the characteristic of a perfect pseudo-random behaviour.

Concerning the distribution of iterates in the phase delay, Figures 20, 21 show that it is very good as well. In those figures we plotted as much as 10^9 points. The only concern is in Figure (22(b)) where the tent pattern is recognizable for $x^{(1)}$ variable. However, for many applications, only one output stream of pseudo-random numbers is required, which can be obtained by using $x^{(2)}$ variable. Both variables have a strong coupling impact on themselves and for the global dynamics.

The techniques make uniform iterates distribution on the torus implying strong chaoticity.

$MTTL_2^{SC}$ alternate map is built using ring-coupling and auto-coupling mechanisms. Since one variable is partially creating the dynamics of the other, we have to check the dependency and repeatability of auto-correlation as well of cross-correlation. Figures 23

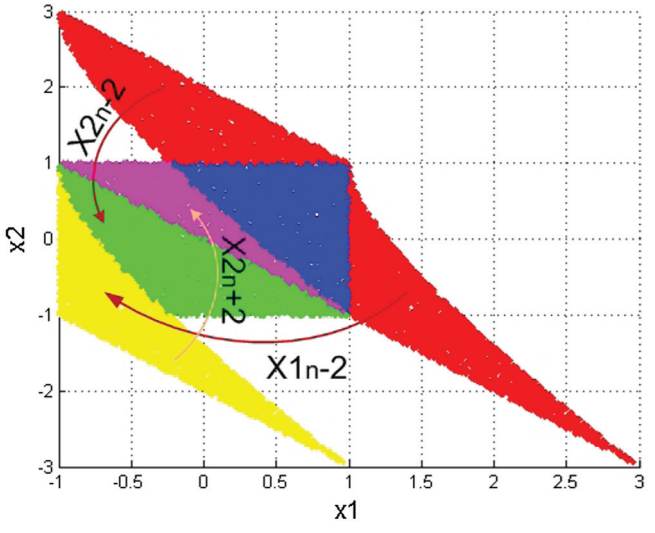

Figure 16. Injection mechanism (40) of $MTTL_2^{SC}$ alternate map (39).

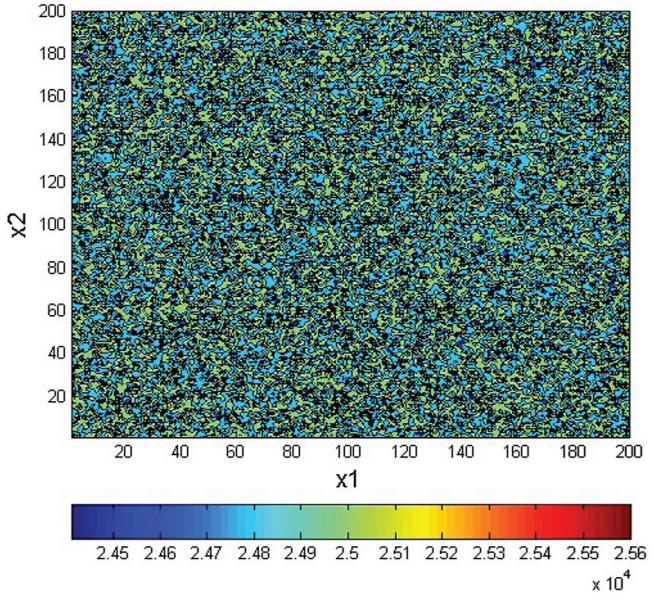

Figure 17. Approximate density function of $MTTL_2^{SC}$ alternate map (39), where $step = 0.01$, 10^9 points are generated.

and 24 display the results of this 2-dimensional mapping. The same excellent results are presented in Figure 23 for auto-correlation, and in Figure 24 for cross-correlation. One can see that graphs lie near the zero axis.

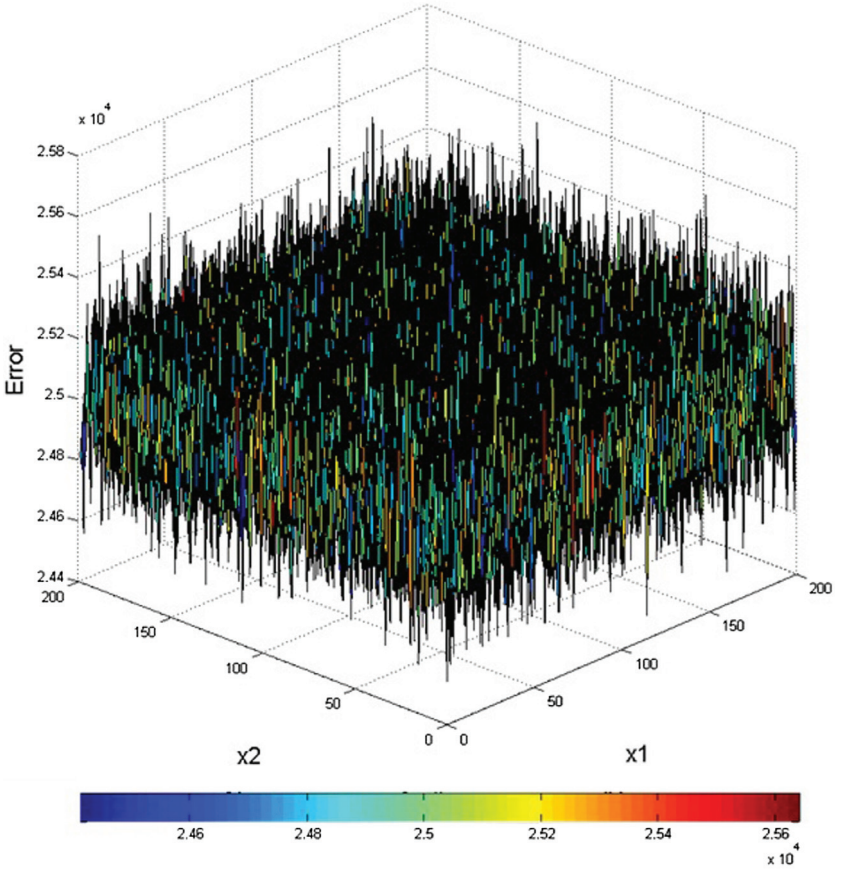

Figure 18. Approximate density function in 3D of $MTTL_2^{SC}$ alternate map, where $step = 0.01$, 10^9 points are generated.

Table 3. Approximate distribution errors (13, 14, 15), for the system (39) in the phase space.

N_{iter}	$x^{(i)}x^{(j)}$	$E_{AC_1,200}$, $N_{iter}(x^{(i)},x^{(j)})$	$E_{AC_2,200}$, $N_{iter}(x^{(i)},x^{(j)})$	$E_{AC_\infty,200}$, $N_{iter}(x^{(i)},x^{(j)})$
10^4	$x^{(1)}x^{(2)}$	1.5532	1.97707	15
10^5	$x^{(1)}x^{(2)}$	0.51531	0.634136	5
10^6	$x^{(1)}x^{(2)}$	0.159092	0.200498	1.16
10^7	$x^{(1)}x^{(2)}$	0.0503666	0.063169	0.252
10^8	$x^{(1)}x^{(2)}$	0.0159087	0.0199153	0.0792
10^9	$x^{(1)}x^{(2)}$	0.00503929	0.00630626	0.02748
10^{10}	$x^{(1)}x^{(2)}$	0.00160207	0.00200759	0.008996
10^{11}	$x^{(1)}x^{(2)}$	0.000505848	0.000631793	0.0029196
10^{12}	$x^{(1)}x^{(2)}$	0.000159674	0.000199849	0.00087284
10^{13}	$x^{(1)}x^{(2)}$	5.04686e−05	6.30986e−05	0.000281112
10^{14}	$x^{(1)}x^{(2)}$	1.59962e−05	2.00927e−05	8.68908e−05

Topological mixing means the system capability to progress over a short period of time. The system from any given initial region or open set of its phase space will be ultimately mixed up with any other region so that it is impossible to predict system evolution.

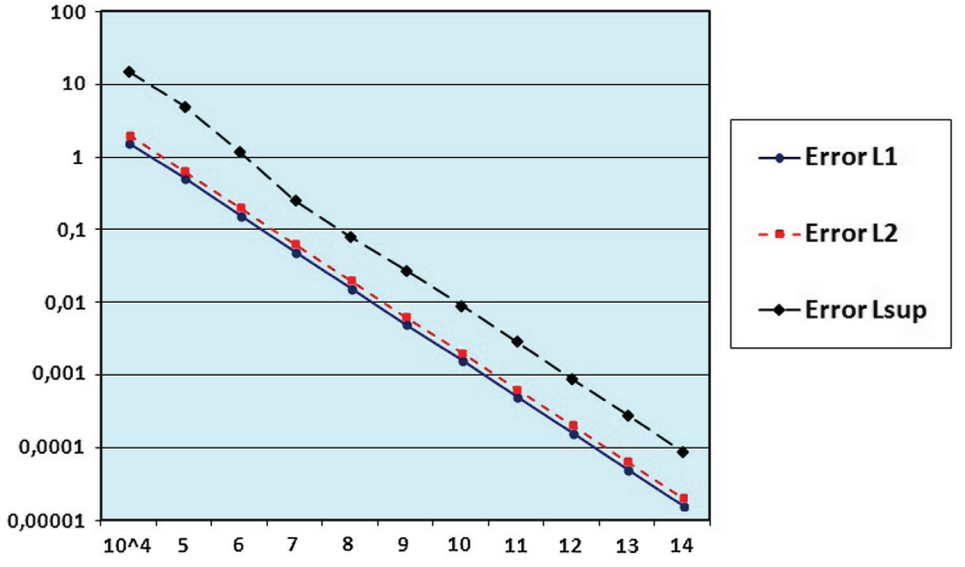

Figure 19. Approximate distribution errors (13), for the system (39).

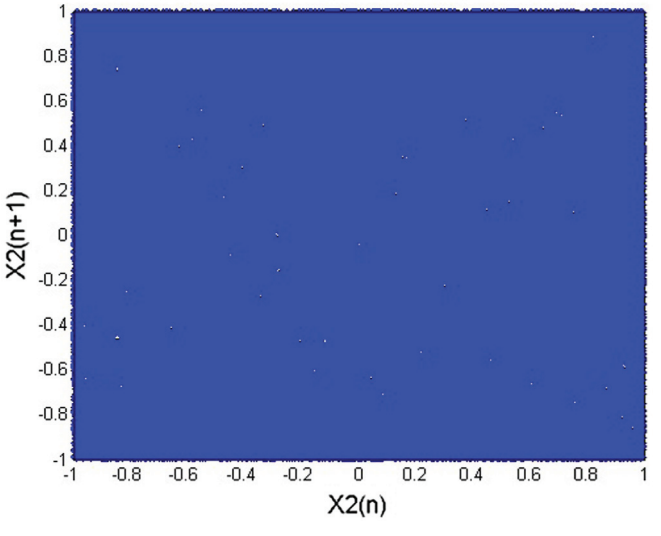

Figure 20. Attractor in the phase delay $(x_n^{(2)}, x_{n+1}^{(2)})$, 10^9 points are generated, for the system (39).

This mapping is spanning through any region or open neighbourhood of a given point in phase space to a larger region of the $[-1,1]^2$ square. Therefore, the image of those regions overlap many other leading to a strong mixing property. It is why it is impossible to predict the evolution of the dynamics.

Here we represent the graphical analysis of the 2-D $MTTL_2^{SC}$ alternate map showing its topological mixing ability. We divide the square $[-1,1]^2$ into 4 orthants and we split each of them in a grid of boxes $(A2, B2, C2, \ldots, O2)$. Up to 5×10^3 iterates are collected in each

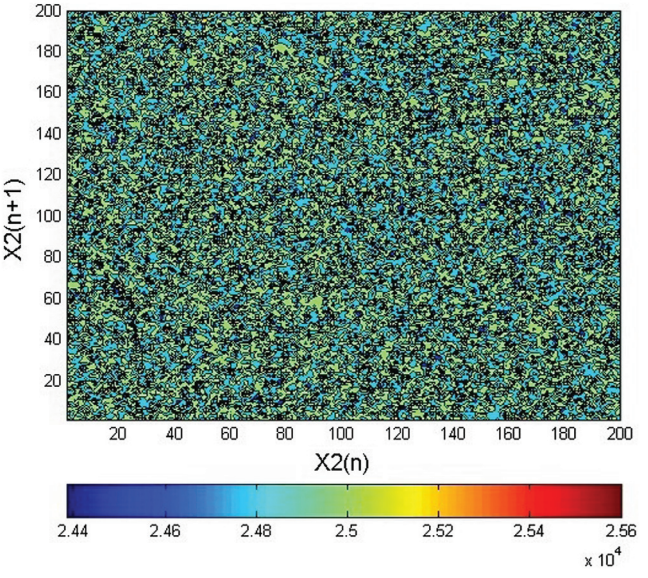

Figure 21. Attractor in the phase delay $(x_n^{(2)}, x_{n+1}^{(2)})$, box-method, 10^9 points are generated, for the system (39) .

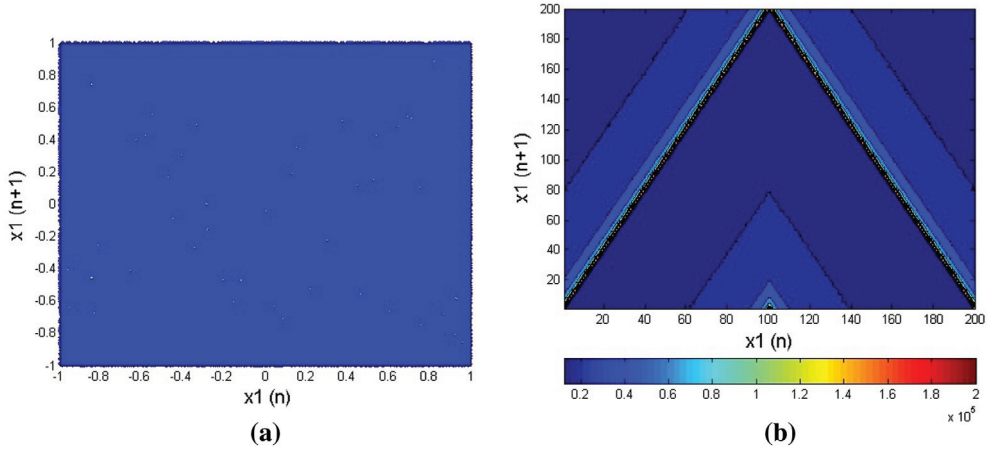

Figure 22. Density of iterates for the phase delay space where 10^9 points are computed (a) plot of 10^9 points following the initial guess $(x_n^{(1)}, x_{n+1}^{(1)})$ (b) $(x_n^{(1)}, x_{n+1}^{(1)})$ using the box-method.

box of the grid (Figure 25). In the next Figures (Figure 26(a)–(e)) we display the image of those initial boxes $(A1, B1, C1, \ldots, O1)$ of the first quadrant by the $MTTL_2^{SC}$ alternate mapping.

Topological mixing can be seen in Figure 26: the points are distributed everywhere over the square. It is near impossible to predict where the iterates will be mapped without the very exact knowledge of the equation of the mapping, the parameters values which can be used as encryption keys when cryptography is involved, or the retrieve of the previous

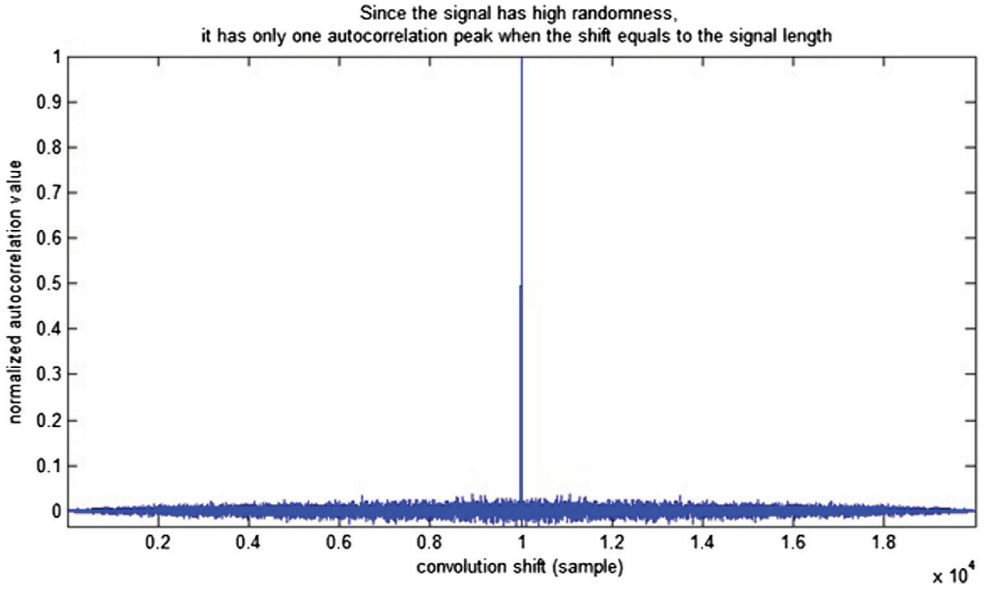

Figure 23. State auto-correlation analysis of the $MTTL_2^{SC}$ alternate map

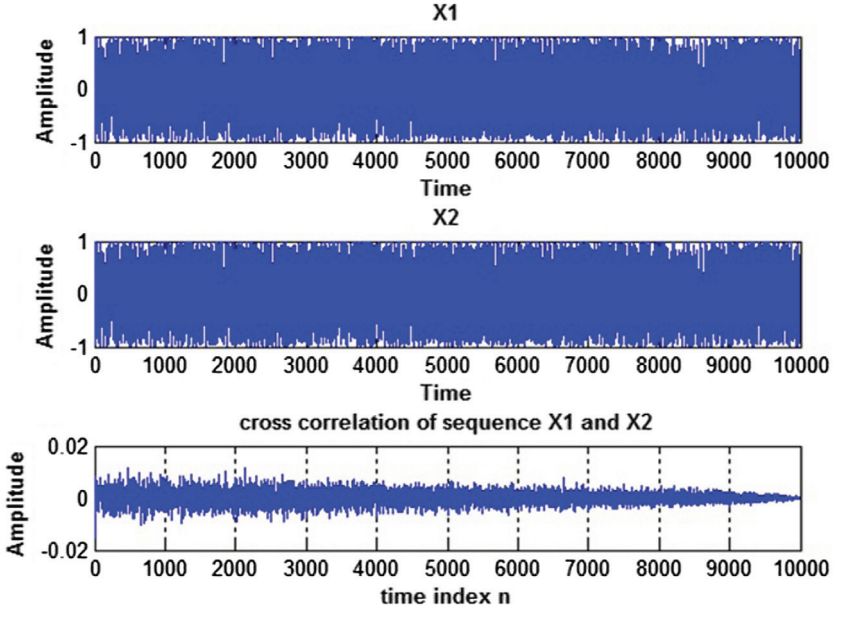

Figure 24. Correlation between states of the $MTTL_2^{SC}$ alternate map.

iterates as well. Due to sensitivity to initial condition, a small error in the knowledge of any initial condition can lead to very different behaviour of the iterates.

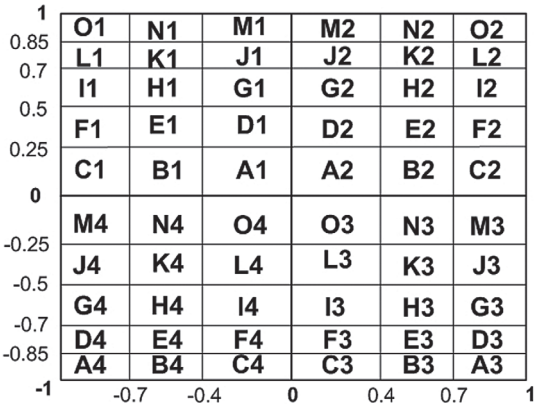

Figure 25. Grid of the boxes $(A, B, C, \ldots, O)$ of the $(x_n^{(1)}, x_n^{(2)})$ square of the phase space.

The system has strong mixing property because the images of the regions overlap on many other. As an example taking several points of the box $A2$ (Figure 25) some are mapped into the same box $A2$ (Figure 26(a)). However, for some other their behaviour is somewhat different, they are mapped through the boxes $O1, I1, P1, C1, B1, E1, H1, M4, N4$ (Figure 27), that means their successor belong somewhere in those boxes (Figure 27). Continuing the mapping process, their iterates mix more complexly, the behaviour becomes unpredictable and eventually looks like scattered points everywhere across the space. Colors and letters overlapping on the graphs vividly display that arbitrarily close points in some periods of time will have vastly different behaviours, which means mixing. This phenomenon is quantified through the value of the Largest Lyapunov exponent. The arbitrarily taken points which are far alone will ultimately approach looking nearly the same only for several iterations means mixing as well. Since the new map implies strong chaos, the phase space is thoroughly mixed together after quite a short time. In a forthcoming paper, we will quantify this mixing, building a corresponding Markov transition matrix as in [11].

NIST tests are used to verify randomness and system capability to resist main attacks. As it was discussed earlier the advantage of the binary sequences has to be approximately the same as of the truly random number generator. NIST tests fully cover the statistical tests. For a long time, the tests are used to check the robustness of the PRNG. In order to assess $MTTL_2^{SC}$ using such tests we have computed up to 4×10^6 points, discarding the 5×10^5 first ones. Because only binary sequences are accepted as input in NIST tests, we have converted the floating numbers obtained from this 2-dimensional mapping, to binary form. The IEEE-754 standard (32 bit single precision floats) is the standard we have used.

We have assessed successfully both variables $x_n^{(1)}$ and $x_n^{(2)}$ of this map, proving therefore strong pseudo-randomness of both streams of numbers. This implies robustness against many statistical attacks when those streams of numbers are used in cryptography (Figure 28). Those chaos based generated sequences of iterates can be considered as good as truly random numbers. Thus, if an intruder try to find some clues to generate such sequences, it will be very difficult to find any, those numbers being not distinguishable from true random numbers.

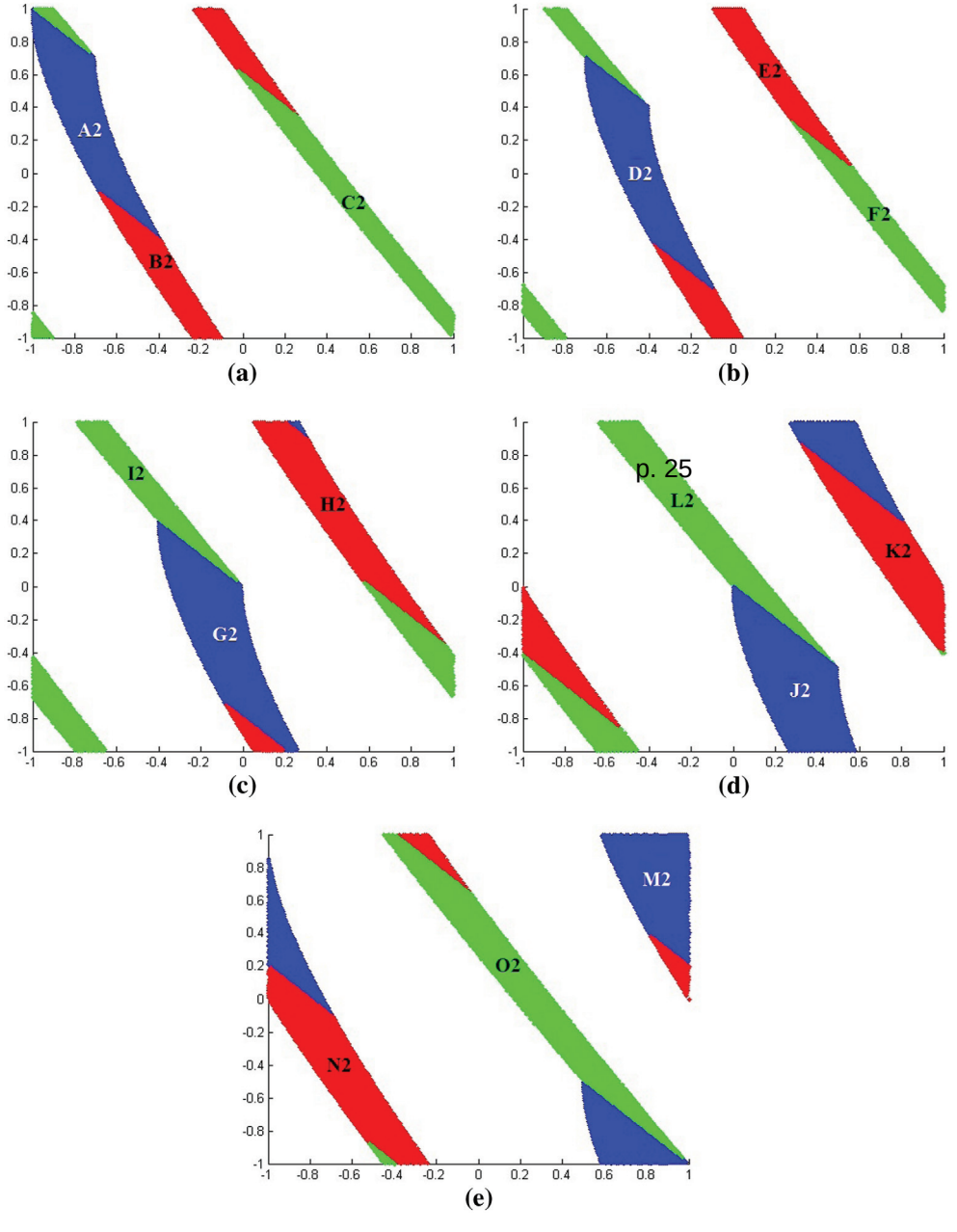

Figure 26. Topological mixing.

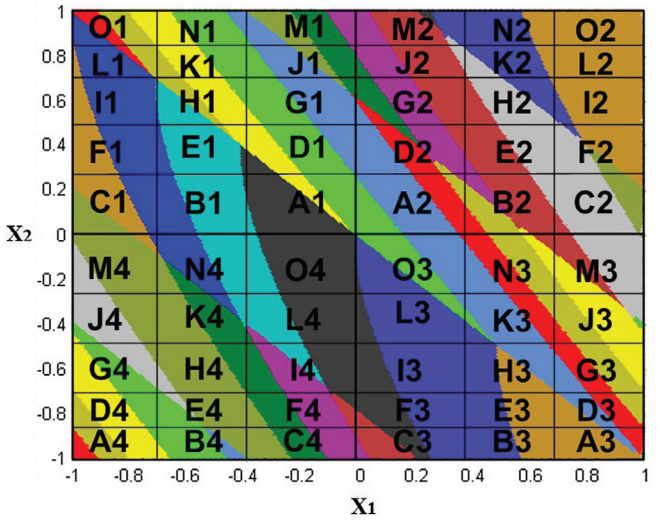

Figure 27. Grid of boxes (*A* . . . *O*) of the four quadrants 1,2,3,4 overlapped by the colored images of the boxes of the first quadrant in the phase space $(x_n^{(1)}, x_n^{(2)})$.

We consider now another topology of network of chaotic maps introducing another structurally simple 2-D mapping. The map is described as follows:

$$NTTL_2^{SC}(x_n^{(1)}, x_n^{(2)}) = \begin{cases} x_{n+1}^{(1)} = 1 - 2|x_n^{(2)}| = T_2(x_n^{(2)}) \\ x_{n+1}^{(2)} = 1 - 2(x_n^{(2)})^2 - 2(|x_n^{(2)}| - |x_n^{(1)}|) \\ \qquad\quad = L_2(x_n^{(2)}) + T_2(x_n^{(2)}) - T_2(x_n^{(1)}) \end{cases} \tag{41}$$

Note, that the first equation corresponds to the tent map. In addition, the injection mechanism (Figure 29) should be applied to hold the dynamics in $[-1, 1]^2$:

$$\begin{aligned} &\text{if } x_{n+1}^{(2)} < -1 \text{ then add } 2 \\ &\text{if } x_{n+1}^{(2)} > 1 \text{ then subtract } 2 \end{aligned} \tag{42}$$

The $NTTL_2$ exhibits an excellent density in phase delay for both states (Figure 30), being very promising in real application.

The $NTTL_2$ map has a complex dynamics which allow it to resist to cryptographic statistical attacks proved by its good results in NIST tests (Figure 31).

In the next section we investigate topologies in dimension of system more than 2. We expect that the complexity of the dynamics should be more complicated and therefore more similar to pseudo-random dynamics [12].

4. A new higher-dimensional map

The use of dimensional systems in higher dimensions are expected more effective to simulate pseudo-random properties because there are more interactions between states with more mixing properties. We have observed that dimensions greater or equal to 3 are

RESULTS FOR THE UNIFORMITY OF P-VALUES AND THE PROPORTION OF PASSING SEQUENCES

generator is <data/Modified TL_{\mu}^{SC} alternative map_x1.txt>

C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	P-VALUE	PROPORTION	STATISTICAL TEST
8	8	11	9	10	8	11	15	11	9	0.897763	100/100	Frequency
13	13	12	7	11	10	12	9	5	8	0.678686	99/100	BlockFrequency
6	7	5	12	16	12	12	9	14	7	0.191687	100/100	CumulativeSums
8	10	12	6	14	12	9	6	12	11	0.678686	100/100	Runs
14	11	12	10	15	5	6	13	8	6	0.236810	99/100	LongestRun
9	6	13	10	7	10	11	11	12	11	0.897763	97/100	Rank
11	12	6	19	4	11	11	13	8	5	0.037566	97/100	FFT
7	9	13	14	12	9	9	11	7	9	0.816537	100/100	NonOverlappingTemplate
10	11	15	10	11	9	12	6	11	5	0.595549	98/100	OverlappingTemplate
11	10	5	7	5	13	16	5	13	15	0.058984	100/100	Universal
14	6	11	10	7	9	13	12	8	10	0.739918	98/100	ApproximateEntropy
2	9	7	8	5	7	5	5	8	7	0.689019	63/63	RandomExcursions
5	8	4	4	6	4	4	11	6	11	0.222869	63/63	RandomExcursionsVariant
12	10	12	13	7	8	7	7	6	18	0.171867	99/100	Serial
9	13	11	12	7	9	7	16	7	9	0.534146	99/100	LinearComplexity

(a)

RESULTS FOR THE UNIFORMITY OF P-VALUES AND THE PROPORTION OF PASSING SEQUENCES

generator is <data/Modified TL_{\mu}^{SC} alternative map_x2.txt>

C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	P-VALUE	PROPORTION	STATISTICAL TEST
18	6	8	12	9	6	7	10	11	13	0.191687	98/100	Frequency
12	7	12	7	3	11	13	10	13	12	0.366918	98/100	BlockFrequency
15	14	8	6	8	13	7	10	9	10	0.494392	98/100	CumulativeSums
12	15	11	8	7	12	9	5	8	13	0.474986	98/100	Runs
9	12	13	13	9	14	9	6	8	7	0.637119	100/100	LongestRun
8	12	8	10	13	15	10	6	7	11	0.616305	98/100	Rank
8	12	9	15	9	8	17	9	9	4	0.181557	99/100	FFT
7	12	7	12	6	9	15	12	7	13	0.437274	100/100	NonOverlappingTemplate
9	12	11	3	16	8	10	13	10	8	0.289667	99/100	OverlappingTemplate
9	13	10	6	8	8	11	10	11	14	0.816537	99/100	Universal
7	24	9	7	7	8	8	17	7	6	0.000347	98/100	ApproximateEntropy
2	4	2	5	5	7	2	13	4	8	0.011791	52/52	RandomExcursions
5	4	8	5	2	1	8	6	4	9	0.191687	52/52	RandomExcursionsVariant
6	10	8	7	15	15	15	8	8	8	0.236810	100/100	Serial
7	9	11	11	6	15	7	11	8	15	0.419021	99/100	LinearComplexity

(b)

Figure 28. Successfull results of the NIST tests for mapping $MTTL_2^{SC}$ alternate (a) $x^{(1)}$ (b) $x^{(2)}$.

more powerful in order to generate generic pseudo-random sequences. Fortunately, the kind of systems we study in this article are able to increase easily the number of components (i.e. the number of states).

Since the specific $MTTL_2^{SC}$ alternate map cannot be extended with more variables, we describe how to achieve such a task in the goal of improving chaoticity, having a more regular distribution of points, a more complex dynamics than $TTL_2^{RC}(x^{(2)}, x^{(1)})$ alternate map (36).

A good manner to get randomness using chaotic maps is to do some coupling between states using auto-coupling or ring-coupling [8]. In this section we consider the special realization of $M_{\mu,p}^{k}$ (34) for which we choose $k^i = +1$ for $1 \leqslant i \leqslant p$ and $(i,j) = (2,1); (3,2); \ldots (1,p)$:

$$p-D, TTL_2^{RC} : \begin{cases} x_{n+1}^{(1)} = 1 - 2|x_n^{(1)}| + 2(|x_n^{(2)}| - (x_n^{(1)})^2) \\ x_{n+1}^{(2)} = 1 - 2|x_n^{(2)}| + 2(|x_n^{(3)}| - (x_n^{(2)})^2) \\ \vdots \\ x_{n+1}^{(p)} = 1 - 2|x_n^{(p)}| + 2(|x_n^{(1)}| - (x_n^{(p)})^2) \end{cases} \tag{43}$$

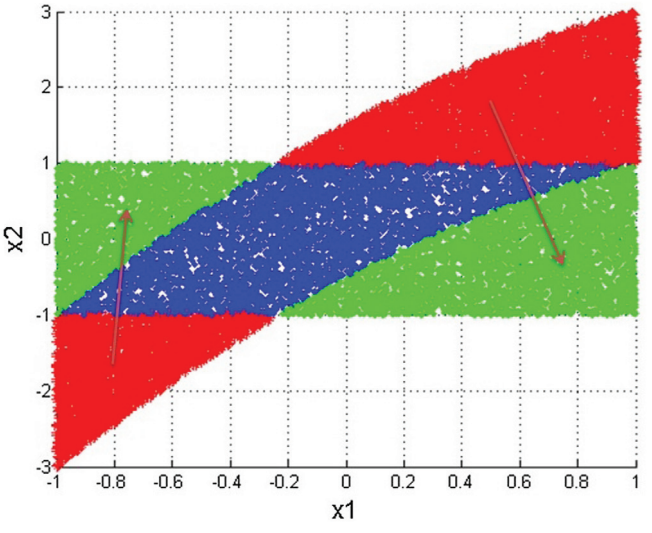

Figure 29. Injection mechanism (40) of $NTTL_2$ alternate map.

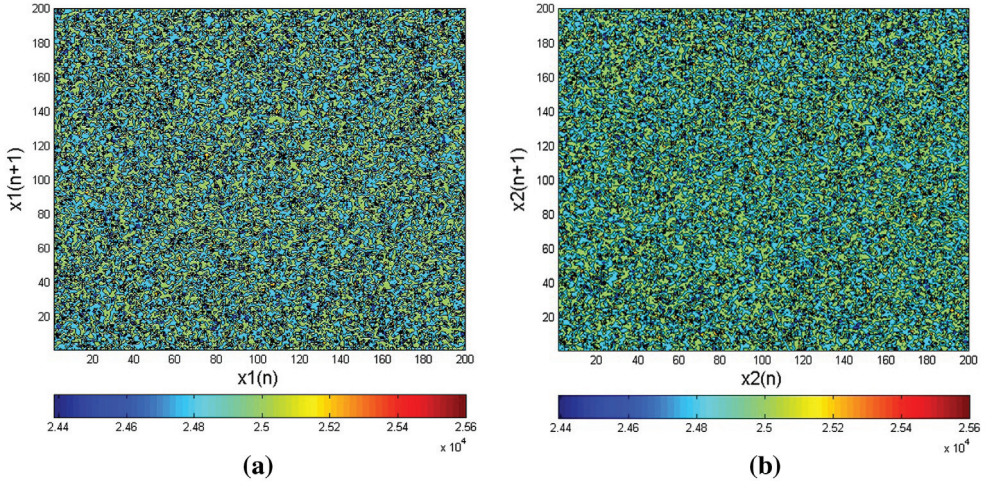

Figure 30. Density of iterates of $NTTL_2$ in the phase delay representation (a) $(x_n^{(1)}, x_{n+1}^{(1)})$ (b) $(x_n^{(2)}, x_{n+1}^{(2)})$.

The injection mechanism (33) is involved at each step.

Note, that each of the states has to satisfy the requirements for chaos and randomness. Therefore, the 3-D, 4-D and 5-D systems were studied for criteria 1–8 (Figure 6) independently for each of the state and in a correlation between them. All of the tests have been successfully passed with improving results when the dimension is increased. In this article we provide only the most significant and important tests.

As for system $MTTL_2^{SC}$, all 3-D, 4-D and 5-D systems (43) show a very good points distribution. The errors $E_{AC_1,200,N}$; $E_{AC_2,200,N}$; $E_{AC_\infty,200,N}$ decrease steadily with respect to the number N of generated points and with respect to the dimension of the system (see

RESULTS FOR THE UNIFORMITY OF P-VALUES AND THE PROPORTION OF PASSING SEQUENCES

generator is <data/x2.txt>

C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	P-VALUE	PROPORTION	STATISTICAL TEST
5	12	10	5	12	12	15	7	14	8	0.236810	99/100	Frequency
8	9	14	8	6	12	12	10	10	11	0.834308	100/100	BlockFrequency
4	12	14	13	10	8	7	7	17	8	0.122325	100/100	CumulativeSums
10	9	10	15	9	12	9	8	9	9	0.924076	100/100	Runs
9	14	7	8	11	12	15	10	7	7	0.554420	100/100	LongestRun
10	11	11	4	14	13	8	13	11	5	0.334538	100/100	Rank
14	9	13	7	11	7	11	14	8	6	0.514124	99/100	FFT
6	10	10	11	5	18	12	3	9	16	0.020548	100/100	NonOverlappingTemplate
12	13	14	11	8	7	9	10	6	10	0.739918	100/100	OverlappingTemplate
7	12	13	16	11	13	13	5	5	5	0.085587	99/100	Universal
12	12	15	4	11	7	10	8	6	15	0.191687	100/100	ApproximateEntropy
3	9	7	7	10	7	6	3	6	6	0.568055	64/64	RandomExcursions
1	6	5	8	8	5	6	7	8	10	0.407091	64/64	RandomExcursionsVariant
14	8	11	10	11	14	9	2	9	12	0.289667	100/100	Serial
9	10	10	5	16	8	5	12	13	12	0.289667	100/100	LinearComplexity

Figure 31. $NTTL_2$ map successfully passed NIST tests.

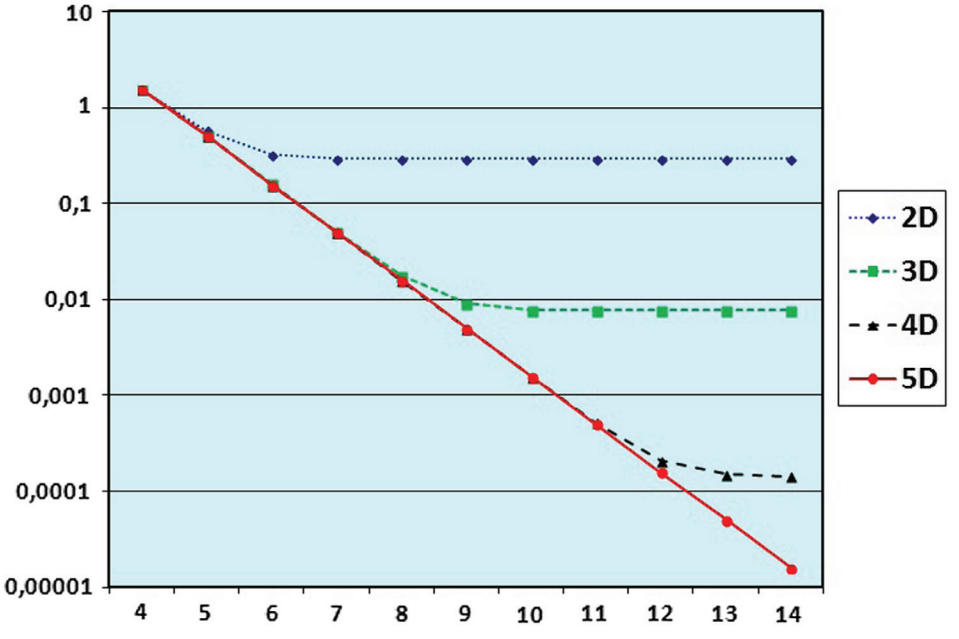

Figure 32. Systems 3-D, 4-D and 5-D, $E_{AC_1,200,N}(x_n^{(1)}, x_n^{(2)})$.

Tables 4–6). In order to display evolution of decreasing error, we have computed from 10^4 up to 10^{14} points, using multicore computer. In those tables the errors are the errors in any phase space: x_n^i vs. $x_n^j, 1 \leqslant i, j \leqslant$ dimension of the system. Figure 32 summarizes those results for the phase space $(x^{(1)}, x^{(2)})$ and the error in norm $E_{AC_1,200,N}$.

Results for the other norms are similar. One can only remark that if for systems 3-D and 5-D the errors are completely similar in magnitude for any phase space $(x^{(i)}, x^{(j)})$, it is not exactly the case for 4-D system. For this system the results are less good for $(x^{(1)}, x^{(3)})$ and $(x^{(2)}, x^{(4)})$ phase spaces than for the other, albeit good enough. The ratio of magnitude

Table 4. Distribution of iterates errors computed for 3-D $TTL_2^{RC}(x^{(i)}, x^{(j)})$ alternate map with 200×200 boxes.

N_{iter}	$x^{(i)}x^{(j)}$	$E_{AC_1,200}$, $N_{iter}(x^{(i)}, x^{(j)})$	$E_{AC_2,200}$, $N_{iter}(x^{(i)}, x^{(j)})$	$E_{AC_\infty,200}$, $N_{iter}(x^{(i)}, x^{(j)})$
10^4	x(1) x(2)	1.5568	2.0006	15
	x(1) x(3)	1.55615	1.99218	15
	x(2) x(3)	1.56015	2.01236	15
10^5	x(1) x(2)	0.5163	0.63752	3.8
	x(1) x(3)	0.51246	0.632626	3.4
	x(2) x(3)	0.51364	0.633454	3.4
10^6	x(1) x(2)	0.160282	0.201211	0.88
	x(1) x(3)	0.160272	0.201644	0.92
	x(2) x(3)	0.159704	0.200846	0.92
10^7	x(1) x(2)	0.0509584	0.0637174	0.264
	x(1) x(3)	0.0510062	0.0639212	0.328
	x(2) x(3)	0.0508942	0.0637151	0.026
10^8	x(1) x(2)	0.0176344	0.0220618	0.0856
	x(1) x(3)	0.0176219	0.02207	0.0948
	x(2) x(3)	0.0175315	0.0219646	0.0916
10^9	x(1) x(2)	0.00911485	0.0112042	0.0408
	x(1) x(3)	0.00907769	0.0111668	0.04576
	x(2) x(3)	0.00906877	0.0111685	0.04204
10^{10}	x(1) x(2)	0.00783204	0.00942348	0.0293
	x(1) x(3)	0.00782634	0.00942161	0.028612
	x(2) x(3)	0.00784682	0.00943308	0.031216
10^{11}	x(1) x(2)	0.00771201	0.0092317	0.0273972
	x(1) x(3)	0.00771015	0.0092324	0.0270468
	x(2) x(3)	0.00771428	0.00923352	0.0262988
10^{12}	x(1) x(2)	0.00769998	0.0092118	0.0261484
	x(1) x(3)	0.00770078	0.00921384	0.0261491
	x(2) x(3)	0.00769833	0.00921204	0.0261546
10^{13}	x(1) x(2)	0.00769867	0.00921041	0.0260495
	x(1) x(3)	0.00769895	0.00921083	0.0259947
	x(2) x(3)	0.00769905	0.00921094	0.0259662
10^{14}	x(1) x(2)	0.00769874	0.00921054	0.02608
	x(1) x(3)	0.00769895	0.00921067	0.026042
	x(2) x(3)	0.00769873	0.00921049	0.0260237

being approximately 8 vs 1 depending on the chosen norm of the error. For systems 2-D, after 10^6 generated points the sequences become repeatable because the errors do not longer decrease. The same situation occurs after 10^9 points for the 3-D system, and after 10^{13} points for the 4-D system. Up to the computation of 10^{14} iterates for the 5-D system (which takes roughly 74 h on a highly-concurrent Sun SPARC Enterprise T5240 server Niagara computer with 128 logical cores) no such repeatable sequences are observed.

Table 5. Distribution of iterates error computed for 4-D $TTL_2^{RC}(x^{(i)}, x^{(j)})$ alternate map with 200 × 200 boxes.

N_{iter}	$(x_n^{(i)}, x_{n+1}^{(i)})$	$E_{AC_1,200}$, $N_{iter}(x^{(i)}, x^{(j)})$	$E_{AC_2,200}$, $N_{iter}(x^{(i)}, x^{(j)})$	$E_{AC_\infty,200}$, $N_{iter}(x^{(i)}, x^{(j)})$
10^4	x(1) x(2)	1.55725	1.9964	15
	x(1) x(3)	1.55695	1.9976	15
	x(1) x(4)	1.55625	1.99539	15
	x(2) x(3)	1.55985	2.01097	15
	x(2) x(4)	1.5541	1.98494	15
	x(3) x(4)	1.55615	1.99319	15
10^5	x(1) x(2)	0.51083	0.631607	4.2
	x(1) x(3)	0.51184	0.631645	3.4
	x(1) x(4)	0.51172	0.631082	3.8
	x(2) x(3)	0.51413	0.63353	3.4
	x(2) x(4)	0.51618	0.636735	3.4
	x(3) x(4)	0.51506	0.634999	3.4
10^6	x(1) x(2)	0.158256	0.199192	1.04
	x(1) x(3)	0.158918	0.199753	1
	x(1) x(4)	0.15942	0.200724	0.92
	x(2) x(3)	0.158746	0.200455	0.92
	x(2) x(4)	0.16029	0.201252	0.92
	x(3) x(4)	0.158228	0.199217	0.84
10^7	x(1) x(2)	0.0504002	0.0632736	0.312
	x(1) x(3)	0.0501062	0.0629123	0.372
	x(1) x(4)	0.050657	0.0634455	0.252
	x(2) x(3)	0.050618	0.0633568	0.272
	x(2) x(4)	0.0506104	0.0635347	0.26
	x(3) x(4)	0.050197	0.0630166	0.288
10^8	x(1) x(2)	0.0157924	0.0197939	0.0848
	x(1) x(3)	0.0160034	0.0200479	0.086
	x(1) x(4)	0.0159352	0.0199486	0.0884
	x(2) x(3)	0.0160063	0.0200269	0.0932
	x(2) x(4)	0.0160009	0.0200713	0.0972
	x(3) x(4)	0.0159821	0.0200349	0.0828
10^9	x(1) x(2)	0.00506758	0.00634845	0.02916
	x(1) x(3)	0.0051405	0.00646279	0.02872
	x(1) x(4)	0.00507518	0.00638122	0.03116
	x(2) x(3)	0.00508724	0.00635539	0.02548
	x(2) x(4)	0.0051651	0.0064819	0.02904
	x(3) x(4)	0.00506246	0.00634861	0.02752
10^{10}	x(1) x(2)	0.00159046	0.00199353	0.008152
	x(1) x(3)	0.00190656	0.00240003	0.010208
	x(1) x(4)	0.00159748	0.00200329	0.008604
	x(2) x(3)	0.0016072	0.00201414	0.008136
	x(2) x(4)	0.00190768	0.00239212	0.010296
	x(3) x(4)	0.00159866	0.00200415	0.008992
10^{11}	x(1) x(2)	0.000521561	0.000653761	0.0030604
	x(1) x(3)	0.00113195	0.00144655	0.0057648
	x(1) x(4)	0.000520847	0.000652876	0.0028436
	x(2) x(3)	0.000521564	0.000654063	0.0029228
	x(2) x(4)	0.00113732	0.00144898	0.0059528
	x(3) x(4)	0.000521442	0.000651506	0.0027404

(*Continued*)

Table 5. (*Continued*).

10^{12}	x(1) x(2)	0.000209109	0.000260625	0.00098868
	x(1) x(3)	0.0010181	0.00131377	0.0047824
	x(1) x(4)	0.000209131	0.000260484	0.00106452
	x(2) x(3)	0.000209507	0.00026041	0.00105064
	x(2) x(4)	0.00101941	0.00131403	0.00464148
	x(3) x(4)	0.000208125	0.000259515	0.00092748
10^{13}	x(1) x(2)	0.000150031	0.000179686	0.000583912
	x(1) x(3)	0.00100668	0.0013004	0.00431795
	x(1) x(4)	0.000150505	0.000180148	0.000604532
	x(2) x(3)	0.00014943	0.000178778	0.000675604
	x(2) x(4)	0.00100677	0.00130022	0.00432338
	x(3) x(4)	0.000149962	0.000179475	0.000645028
10^{14}	x(1) x(2)	0.000144162	0.000169327	0.000502641
	x(1) x(3)	0.00100504	0.00129871	0.00426248
	x(1) x(4)	0.000144398	0.0001695	0.000490893
	x(2) x(3)	0.000144101	0.000169222	0.000491529
	x(2) x(4)	0.00100495	0.00129875	0.00427791
	x(3) x(4)	0.000144307	0.00016943	0.000504066

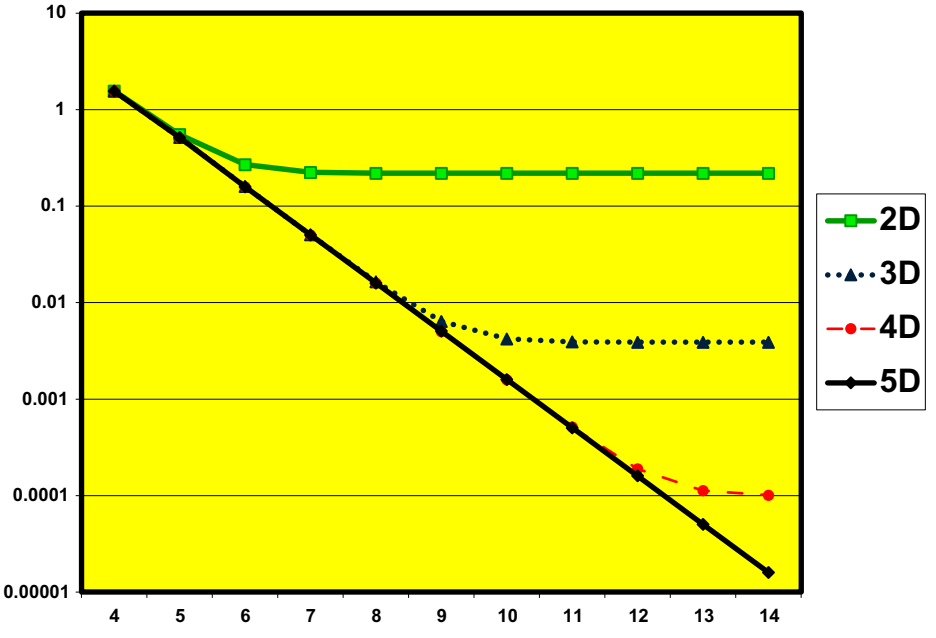

Figure 33. Systems 3-D, 4-D and 5-D, Errors $E_{C_1,200,N}(x_n^{(1)}, x_{n+1}^{(1)})$.

Remark: We use Sun SPARC Enterprise T5240 server with two 1.2 Ghz, 16 core UltraSPARC T2+ processors. Each core has 8 hardware contexts, which means the server is capable of running up 128 threads in 'true' (hardware) concurrency.

In the phase-delay space $(x_n^{(i)}, x_{n+1}^{(i)})$ we obtain also very good results of uniform repartition of iterates for all 3-D and 5-D systems (see Tables 7–9). Figure 33 summarizes such results.

Table 6. Distribution of iterates error computed for 5-D $TTL_2^{RC}(x^{(i)}, x^{(j)})$ alternate map with 200 × 200 boxes.

Points	$(x_n^{(i)}, x_{n+1}^{(i)})$	$E_{AC_1,200}$, $N_{iter}(x^{(i)}, x^{(j)})$	$E_{AC_2,200}$, $N_{iter}(x^{(i)}, x^{(j)})$	$E_{AC_\infty,200}$, $N_{iter}(x^{(i)}, x^{(j)})$
10^4	x(1) x(2)	1.55915	2.00818	15
	x(1) x(3)	1.5561	1.99539	19
	x(1) x(4)	1.55935	2.00559	15
	x(1) x(5)	1.5568	1.99539	15
	x(2) x(3)	1.55785	2.0018	15
	x(2) x(4)	1.55575	1.98655	15
	x(2) x(5)	1.5584	2.00499	19
	x(3) x(4)	1.55655	1.9962	15
	x(3) x(5)	1.55675	1.99359	19
	x(4) x(5)	1.55625	1.99098	15
10^5	x(1) x(2)	0.514	0.633448	3.4
	x(1) x(3)	0.5123	0.631481	3.4
	x(1) x(4)	0.51109	0.631335	3.8
	x(1) x(5)	0.51285	0.633132	3.4
	x(2) x(3)	0.5118	0.632158	3.8
	x(2) x(4)	0.51545	0.634173	3.8
	x(2) x(5)	0.51376	0.633599	3.8
	x(3) x(4)	0.51271	0.631303	3.4
	x(3) x(5)	0.51183	0.63046	3.8
	x(4) x(5)	0.5129	0.632822	3.4
10^6	x(1) x(2)	0.158058	0.198943	0.96
	x(1) x(3)	0.158956	0.199411	0.92
	x(1) x(4)	0.15943	0.200825	0.88
	x(1) x(5)	0.159074	0.200251	0.92
	x(2) x(3)	0.15825	0.199351	0.92
	x(2) x(4)	0.159248	0.200233	1
	x(2) x(5)	0.15889	0.199995	0.84
	x(3) x(4)	0.159136	0.199863	1
	x(3) x(5)	0.159216	0.200545	0.96
	x(4) x(5)	0.158918	0.199639	0.88
10^7	x(1) x(2)	0.0505508	0.0634574	0.308
	x(1) x(3)	0.0504804	0.0632541	0.272
	x(1) x(4)	0.0501244	0.0628956	0.268
	x(1) x(5)	0.0503472	0.063055	0.296
	x(2) x(3)	0.0503194	0.0632425	0.268
	x(2) x(4)	0.050569	0.0634408	0.296
	x(2) x(5)	0.0506322	0.0635636	0.284
	x(3) x(4)	0.0503476	0.063059	0.256
	x(3) x(5)	0.0504622	0.063331	0.276
	x(4) x(5)	0.0505216	0.0633772	0.288
10^8	x(1) x(2)	0.0160114	0.0200538	0.0852
	x(1) x(3)	0.0159261	0.0199328	0.0892
	x(1) x(4)	0.0160321	0.0200284	0.0844
	x(1) x(5)	0.0158962	0.019966	0.0844
	x(2) x(3)	0.0159754	0.020018	0.094
	x(2) x(4)	0.0159668	0.020047	0.0808
	x(2) x(5)	0.0160116	0.0200677	0.0904
	x(3) x(4)	0.0158826	0.01993	0.0924
	x(3) x(5)	0.0159341	0.0199285	0.084
	x(4) x(5)	0.0160516	0.0200876	0.0936

(*Continued*)

Table 6. (*Continued*).

10^9	x(1) x(2)	0.00507915	0.0063595	0.02716
	x(1) x(3)	0.00504164	0.00631924	0.02888
	x(1) x(4)	0.00503177	0.00631229	0.02452
	x(1) x(5)	0.00504183	0.00630869	0.02744
	x(2) x(3)	0.00504652	0.00632572	0.02768
	x(2) x(4)	0.00505682	0.00633798	0.02468
	x(2) x(5)	0.00505273	0.00634782	0.02728
	x(3) x(4)	0.00502485	0.00630083	0.03192
	x(3) x(5)	0.00504935	0.00633202	0.0268
	x(4) x(5)	0.00501553	0.00628623	0.02588
10^{10}	x(1) x(2)	0.0015927	0.00199644	0.008128
	x(1) x(3)	0.00159456	0.00199969	0.008364
	x(1) x(4)	0.00160091	0.0020046	0.009144
	x(1) x(5)	0.00160204	0.00200558	0.008756
	x(2) x(3)	0.00159442	0.00199577	0.008104
	x(2) x(4)	0.00159961	0.00200365	0.007988
	x(2) x(5)	0.0015934	0.00199718	0.00916
	x(3) x(4)	0.00158123	0.00198712	0.008132
	x(3) x(5)	0.00161268	0.00201818	0.008176
	x(4) x(5)	0.0016008	0.00200208	0.008832
10^{11}	x(1) x(2)	0.000506086	0.000633916	0.0025712
	x(1) x(3)	0.000506032	0.000634157	0.002604
	x(1) x(4)	0.000506226	0.000634534	0.0031212
	x(1) x(5)	0.000507563	0.000635621	0.0027628
	x(2) x(3)	0.000508303	0.000636715	0.002912
	x(2) x(4)	0.000505896	0.000632921	0.0025112
	x(2) x(5)	0.000508998	0.000637142	0.0027688
	x(3) x(4)	0.000505468	0.000631842	0.0025664
	x(3) x(5)	0.000505627	0.000633985	0.002762
	x(4) x(5)	0.000503823	0.000632957	0.0025036
10^{12}	x(1) x(2)	0.000158795	0.000199203	0.00089288
	x(1) x(3)	0.000159326	0.000199796	0.00086472
	x(1) x(4)	0.000160038	0.000200669	0.00082136
	x(1) x(5)	0.000159048	0.000199636	0.0008704
	x(2) x(3)	0.000160659	0.000201643	0.00090456
	x(2) x(4)	0.000160313	0.000201294	0.00091648
	x(2) x(5)	0.000160462	0.00020094	0.00082616
	x(3) x(4)	0.000158758	0.000198643	0.00091512
	x(3) x(5)	0.000159079	0.000199344	0.00087596
	x(4) x(5)	0.000159907	0.000200293	0.00085868
10^{13}	x(1) x(2)	5.03666e−05	6.30356e−05	0.000270156
	x(1) x(3)	5.09066e−05	6.38229e−05	0.000298932
	x(1) x(4)	5.09599e−05	6.39809e−05	0.00026382
	x(1) x(5)	5.00546e−05	6.27638e−05	0.00028508
	x(2) x(3)	5.03313e−05	6.31806e−05	0.000250588
	x(2) x(4)	5.12567e−05	6.43641e−05	0.00030346
	x(2) x(5)	5.10924e−05	6.41551e−05	0.000276092
	x(3) x(4)	5.05484e−05	6.33096e−05	0.000259888
	x(3) x(5)	5.06863e−05	6.36835e−05	0.000292484
	x(4) x(5)	5.03483e−05	6.31095e−05	0.000284756

(*Continued*)

Table 6. (*Continued*).

10^{14}	x(1) x(2)	1.60489e−05	2.00692e−05	8.53124e−05
	x(1) x(3)	1.73852e−05	2.18348e−05	9.88376e−05
	x(1) x(4)	1.74599e−05	2.18483e−05	9.66572e−05
	x(1) x(5)	1.59133e−05	1.99122e−05	8.96988e−05
	x(2) x(3)	1.60419e−05	2.01421e−05	9.01576e−05
	x(2) x(4)	1.73507e−05	2.17665e−05	9.4832e−05
	x(2) x(5)	1.73496e−05	2.17415e−05	9.1582e−05
	x(3) x(4)	1.59451e−05	1.9985e−05	8.67056e−05
	x(3) x(5)	1.75013e−05	2.19225e−05	9.8746e−05
	x(4) x(5)	1.59445e−05	2.0002e−05	8.79312e−05

Table 7. Distribution of iterates errors computed for 3-D *TTL* in phase delay $(x_n^{(i)}, x_{n+1}^{(i)})$ with 200 × 200 boxes.

Points	$(x_n^{(i)}, x_{n+1}^{(i)})$	$E_{C_1,200}$	$E_{C_2,200}$	$E_{C_\infty,200}$
10^4	Identical values for x(1) x(2) and x(3)	1.55575	1.99058	15
10^5	Identical values for x(1) x(2) and x(3)	0.51516	0.634375	3.4
10^6	Identical values for x(1) x(2) and x(3)	0.160148	0.201038	0.92
10^7	Identical values for x(1) x(2) and x(3)	0.0505148	0.0633416	0.26
10^8	Identical values for x(1) x(2) and x(3)	0.0164343	0.0205923	0.0888
10^9	Identical values for x(1) x(2) and x(3)	0.00640451	0.00804826	0.03748
10^{10}	Identical values for x(1) x(2) and x(3)	0.00420824	0.00533879	0.02388
10^{11}	Identical values for x(1) x(2) and x(3)	0.00392619	0.00499949	0.0231904
10^{12}	Identical values for x(1) x(2) and x(3)	0.00388937	0.00496257	0.0219244
10^{13}	Identical values for x(1) x(2) and x(3)	0.00388768	0.0049599	0.0222608
10^{14}	Identical values for x(1) x(2) and x(3)	0.003887	0.00495925	0.0222616

Table 8. Distribution of iterates errors computed for 4-D *TTL* in phase delay $(x_n^{(i)}, x_{n+1}^{(i)})$ with 200 × 200 boxes.

Points	$(x_n^{(i)}, x_{n+1}^{(i)})$	$E_{C_1,200}$	$E_{C_2,200}$	$E_{C_\infty,200}$
10^4	Identical values for x(1) to x(4)	1.5571	1.996	15
10^5	Identical values for x(1) to x(4)	0.51115	0.631113	3.4
10^6	Identical values for x(1) to x(4)	0.158472	0.198974	0.88
10^7	Identical values for x(1) to x(4)	0.0503522	0.0632225	0.256
10^8	Identical values for x(1) to x(4)	0.0159245	0.0199042	0.084
10^9	Identical values for x(1) to x(4)	0.00502109	0.00629915	0.02732
10^{10}	Identical values for x(1) to x(4)	0.00159193	0.0019977	0.00866
10^{11}	Identical values for x(1) to x(4)	0.00051438	0.000643966	0.0028136
10^{12}	Identical values for x(1) to x(4)	0.000189418	0.000238006	0.00098772
10^{13}	Identical values for x(1) to x(4)	0.000112771	0.000143508	0.000675228
10^{14}	Identical values for x(1) to x(4)	0.000101139	0.00013067	0.000474188

Table 9. Distribution of iterates errors computed for 5-D *TTL* in phase delay $(x_n^{(i)}, x_{n+1}^{(i)})$ with 200 × 200 boxes.

Points	$(x_n^{(i)}, x_{n+1}^{(i)})$	$E_{C_1,200}$	$E_{C_2,200}$	$E_{C_\infty,200}$
10^4	Identical values for x(1) to x(5)	1.5577	2.0012	19
10^5	Identical values for x(1) to x(5)	0.51372	0.633959	3.8
10^6	Identical values for x(1) to x(5)	0.15872	0.199793	0.88
10^7	Identical values for x(1) to x(5)	0.0503658	0.0631425	0.26
10^8	Identical values for x(1) to x(5)	0.0159765	0.0200503	0.084
10^9	Identical values for x(1) to x(5)	0.00509015	0.00636626	0.02528
10^{10}	Identical values for x(1) to x(5)	0.00159581	0.00199936	0.008604
10^{11}	Identical values for x(1) to x(5)	0.000505068	0.000633088	0.0025432
10^{12}	Identical values for x(1) to x(5)	0.000160547	0.000201102	0.0008602
10^{13}	Identical values for x(1) to x(5)	5.0394e−05	6.31756e−05	0.000280168
10^{14}	Identical values for x(1) to x(5)	1.59929e−05	2.00533e−05	9.89792e−05

C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	P-VALUE	PROPORTION	STATISTICAL TEST
8	14	8	9	10	9	11	12	6	13	0.779188	100/100	Frequency
11	9	9	8	6	15	7	13	9	13	0.574903	100/100	BlockFrequency
14	6	13	7	11	5	10	11	9	14	0.401199	100/100	CumulativeSums
12	10	7	7	16	8	13	7	13	7	0.366918	99/100	CumulativeSums
16	9	7	11	14	12	6	13	7	5	0.181557	100/100	Runs
13	9	14	11	11	8	9	12	5	8	0.678686	100/100	LongestRun
14	9	7	8	9	16	9	12	6	10	0.455937	100/100	Rank
13	4	9	11	7	4	10	12	19	11	0.037566	100/100	FFT
14	8	8	9	8	15	11	11	8	8	0.699313	100/100	NonOverlappingTemplate
14	15	12	10	6	9	13	7	3	11	0.162606	99/100	OverlappingTemplate
8	7	11	16	9	12	10	9	7	11	0.678686	100/100	Universal
13	11	10	12	6	12	12	14	6	4	0.304126	97/100	ApproximateEntropy
5	5	6	9	2	7	5	8	9	6	0.637119	62/62	RandomExcursions
6	2	4	9	6	11	6	5	6	7	0.407091	62/62	RandomExcursionsVariant
13	8	15	8	12	9	7	15	8	5	0.275709	99/100	Serial
13	6	15	12	11	6	15	8	8	6	0.213309	99/100	Serial
9	6	8	13	8	11	10	11	12	12	0.883171	99/100	LinearComplexity

(a)

C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	P-VALUE	PROPORTION	STATISTICAL TEST
7	5	12	14	10	9	12	16	8	7	0.289667	99/100	Frequency
7	7	9	10	6	10	14	8	10	19	0.137282	99/100	BlockFrequency
8	2	9	16	13	9	13	9	7	14	0.090936	99/100	CumulativeSums
5	8	14	11	11	11	14	5	10	11	0.437274	99/100	CumulativeSums
6	16	13	11	9	10	8	7	11	9	0.554420	100/100	Runs
9	13	6	9	14	10	8	11	12	8	0.779188	99/100	LongestRun
9	8	14	6	12	12	8	10	8	13	0.719747	100/100	Rank
10	10	17	5	9	13	14	10	6	6	0.153763	99/100	FFT
9	7	9	13	9	10	10	14	6	13	0.719747	100/100	NonOverlappingTemplate
5	9	12	7	7	12	12	13	12	11	0.637119	99/100	OverlappingTemplate
12	16	8	7	9	10	7	12	8	11	0.616305	99/100	Universal
8	16	6	12	11	13	5	7	13	9	0.249284	99/100	ApproximateEntropy
4	8	4	6	8	5	7	8	9	7	0.804337	66/66	RandomExcursions
4	7	7	8	2	8	6	8	7	9	0.602458	66/66	RandomExcursionsVariant
11	10	10	18	6	5	11	12	10	7	0.213309	100/100	Serial
8	11	10	10	12	11	10	9	9	10	0.998821	98/100	Serial
10	7	13	11	8	7	11	14	11	8	0.798139	99/100	LinearComplexity

(b)

Figure 34. NIST tests for (a) 3-D $TTL_2^{RC}(x^{(2)}, x^{(1)})$ alternate map (b) 4-D $TTL_2^{RC}(x^{(2)}, x^{(1)})$ alternate map.

Numerical results provide harmony of the points density between states. In addition, the NIST tests show that the multidimensional map behaves randomly (Figure 34).

5. Conclusion

In this paper we have studied several topologies of complex networks of chaotic maps, especially exploring the idea to mix two well-known chaotic maps: the tent map (or PWL map) and the logistic map, which do not possess all the needed properties for encryption purposes when used separately and coupling such maps in complex networks.

Either in 2-D or in upper dimensions the new coupling of mixed tent-logistic map with injection mechanism changed qualitatively the overall system behaviour, increasing their complexity. In a bottom up deMarche of building such networks, we highlighted that useful randomness for cryptographic purpose can emerge from the studied topologies. The proposed systems exhibit strong nonlinear dynamics, demonstrating great sensitivity to initial conditions. They generate strong chaotic dynamics characterized by positive Lyapunov exponents with values far greater than zero. Moreover, concerning the consolidated criteria for robust CPRNG, they successfully comply all the required tests such as: NIST tests, uniform distribution, either in the phase space or in the phase delay, both cross-correlation and autocorrelation tests.

We have checked their complex behaviour doing an analysis of their dynamics using in this aim bifurcation diagram, study of topological mixing.

In conclusion, we were able to demonstrate the totally unpredictable dynamics of such multidimensional mapping, making those systems good potential candidate for high security applications.

Disclosure statement

No potential conflict of interest was reported by the authors.

References

[1] J.M. Aguirregabiria, *Robust chaos with variable lyapunov exponent in smooth one-dimensional maps*, Chaos Solitons Fractals 42 (2009), pp. 2531–2539.

[2] M. Ariffin and M. Noorani, *Modified Baptista type chaotic cryptosystem via matrix secret key*, Phys. Lett. A 372 (2008), pp. 5427–5430.

[3] M. Baptista, *Cryptography with chaos*, Phys. Lett. A 240 (1998), pp. 50–54.

[4] P. Diamond, P. Kloeden, A. Pokrovskii, and A. Vladimirov, *Collapsing effects in numerical simulation of a class of chaotic dynamical systems and random mappings with a single attracting centre*, Phys. D 86 (1995), pp. 559–571.

[5] A. Espinel and I. Taralova, *Ring-coupled chaotic generator for coherent and non-coherent detection*, in *Adaptation and Learning in Control and Signal Processing, IFAC Proceedings*, Caen, France, Vol. 11, 2013, pp. 718–723.

[6] O.E. Lanford III, *Informal remarks on the orbit structure of discrete approximations to chaotic maps*, Exp. Math. 7 (1998), pp. 317–324.

[7] R. Lozi, *Chaotic pseudo random number generators via ultra weak coupling of chaotic maps and double threshold sampling sequences*, in *Proceedings of ICCSA 2009 The 3rd International Conference on Complex Systems and Applications*, University of Le Havre, France, June 29–July 02 2009, pp. 20–24.

[8] R. Lozi, *Emergence of randomness from chaos*, Int. J. Bifur. Chaos 22 (2012), pp. 1250021-1/15.

[9] R. Lozi, *Can we trust in numerical computations of chaotic solutions of dynamical systems?*, Dyn. Chaos, World Sci. Ser. Nonlinear Sci. Ser. A 84 (2013), pp. 63–98.

[10] R. Lozi and C. Fiol, *Global orbit patterns for one dimensional dynamical systems*, in *Iteration Theory*, Grazer Mathematische Berischte, 2009, pp. 112–144.

[11] R. Lozi and I. Taralova, *From chaos to randomness via geometric undersampling*, ESAIM Proc. Surveys 46 (2014), pp. 177–195.
[12] G. Manjunath, D. Fournier-Prunaret, and A.K. Taha, *A 3-dimensional piecewise affine map used as a chaotic generator*, in *Iteration Theory*, Gvazer Mathematische Berischte, 2009, pp. 145–157.
[13] R.F. Martinez-Gonzalez, J.A. Diaz-Mendez, and R. Vazquez-Medina, *Vhdl implementation for a pseudo random number generator based on tent map*, Comput. Appl. Math. 1 (2015), pp. 12–15.
[14] C. Mira, L. Gardini, A. Barugola, and J.C. Cathala, *Chaotic Dynamics in Two-dimensional Noninvertible Maps*, Vol. 20, World Scientific, 1996.
[15] H. Nejati, A. Beirami, and Y. Massoud, *A realizable modified tent map for true random number generation*, in *51st Midwest Symposium on IEEE Circuits and Systems, 2008. MWSCAS 2008*, 2008, pp. 621–624.
[16] S.M. Ross, *Introduction to Probability Models*, Academic Press, 2014.
[17] A. Sharkovskii, *Coexistence of cycles of a continuous map of the line into itself*, Int. J. Bifur. Chaos 5 (1995), pp. 1263–1273.
[18] Y. Wang, Z. Liu, J. Ma, and H. He, *A pseudorandom number generator based on piecewise logistic map*, Nonlinear Dyn. 83 (2016), pp. 2373–2391.
[19] W.K. Wong, L.P. Lee, and K.W. Wong, *A modified chaotic cryptographic method*, in *Communications and Multimedia Security Issues of the New Century*, Springer, 2001, pp. 123–126.
[20] G. Yuan and J.A. Yorke, *Collapsing of chaos in one dimensional maps*, Phys. D 136 (2000), pp. 18–30.