Nontrivial behavior of the fixed-point version of $2D$-chaotic maps

Luciana De Micco [a,b], Maximiliano Antonelli [a], Hilda A. Larrondo [a,b]

[a] *ICYTE (Instituto de Investigaciones Científicas y Tecnológicas en Electrónica) Facultad de Ingeniería, Universidad Nacional de Mar del Plata Juan B. Justo 4302, Mar del Plata Buenos Aires, Argentina.*

[b] *Fellow of CONICET-Argentina*

Abstract

This paper deals with a family of interesting $2D$-quadratic maps proposed by Sprott, in his seminal paper [1], related to "chaotic art". Only results for the analytical representation of these maps have been published in the open literature. Our main interest in these maps is they may be used to generate a novel encryption system, because they present multiple chaotic attractors depending on the selected point in the parameter's space. Consequently the objective of this paper is to extend the analysis to the digital version, to make possible the hardware implementation in a digital medium, like Field Programmable Gate Arrays (FPGA) in fixed-point arithmetic. Our main contributions are: (a) the study of the domains of attraction in fixed-point arithmetic; (b) the determination of the *threshold* of the bus width that preserves the integrity of the domain of attraction and (c) the comparison between two quantifiers based on respective probability distribution functions and the well known Maximum Lyapunov Exponent (*MLE*) to detect the above mentioned threshold.

VERSION: 5 December 2016

1 Introduction

Chaotic systems have an increasing number of applications and their implementation is specially involved due to the *extreme sensitivity to initial condi-*

Email address: `ldemicco@fi.mdp.edu.ar` (Luciana De Micco).

tions. In general, these systems are used for the generation of controlled noises, these digital Pseudo-Random Noise Generators (PRNGs) can be employed in a large number of electronic applications, such as encryption sequences for privacy, multiplexing techniques, electromagnetic compatibility and so on [2–4]. In computers and digital devices only *pseudo chaotic* attractors may be generated. But discretization may even destroy the *pseudo chaotic* behavior and consequently is a non trivial process.

Among many chaotic systems available in the literature, we are interested in a family of $2D$-maps [1] proposed by Sprott, and modelled by a pair of coupled quadratic equations:

$$\begin{cases} x_{n+1} = a_1 \ + \ a_2\ x_n \ + \ a_3\ x_n^2 \ + \ a_4\ x_n\ y_n \ + \ a_5\ y_n \ + \ a_6\ y_n^2 \\ y_{n+1} = a_7 \ + \ a_8\ x_n \ + \ a_9\ x_n^2 \ + \ a_{10}\ x_n\ y_n \ + \ a_{11}\ y_n \ + \ a_{12}\ y_n^2 \end{cases} \tag{1}$$

where $\{x, y\}$ are the state variables and $\{a_i, i = 1, \ldots, 12\}$ are the parameters. The reasons to study this particular system are two-fold:

(1) using floating-point arithmetic Sprott saw that by automatic swept of parameters a_i a huge number of points in the parameter's space (about 6.10^{16}) having a chaotic permanent regime may be detected. He also found a correlation between the correlation dimension and the Lyapunov exponents of these chaotic attractors, with their *visual appeal*, an interesting issue for automatic *art* generation
(2) it is possible to generate a novel encryption system because they present multiple chaotic attractors depending on the selected point in the parameter's space. For the replacement of the S-box in AES, or for the development of a new encryption algorithm [5,6].

Digital hardware implementation of dynamical systems, requires the use of a finite number of bits to represent the state variables. Only rational numbers may be represented in a computer, in spite of the arithmetics used (fixed-point or floating-point arithmetics). From an engineering point of view, fixed-point arithmetic is more efficient than floating-point because it uses less resources, and each operation requires a lower number of clock cycles. As a consequence, power consumption is also diminished using fixed-point arithmetic. Floating-point architecture, on the other hand, allows one to *recreate* the ideal system's trajectories in $\mathbb{R}^n$.

Only results for the analytical representation of the maps in Eq. 1 have been published in the open literature. The objective of this paper is to extend the analysis to the digital version, to make possible the hardware implementation in fixed-point arithmetic.

Several strategies have been proposed in the literature for a correct selection of the optimal number of bits in hardware implementations. However, most of these procedures are limited to linear systems [7,8]. In digital chaotic systems, a completely different behavior may be obtained by varying the precision. This issue has gained interest recently, and several new schemes have been proposed [9–11].

Grebogi's work [12] showed that the average length T of periodic orbits of a dynamical system implemented in a computer, scales as a function of the computer precision ξ and the correlation dimension of the chaotic attractor, as $T \sim \xi^{-d/2}$. In [13] some findings on a new series of dynamical indicators, which can quantitatively reflect the degradation effects on a digital chaotic map realized with a fixed-point finite precision have been reported, but they are restricted to $1D$ piecewise linear chaotic maps (PWLCM). In [14] the effect of numerical precision on the mean distance and on the mean coalescence time between trajectories of deterministic maps with either multiplicative noise parameter or with an additive noise term was investigated.

In this work we developed a detailed analysis of the *degradation* of the multi-attractor chaotic system modelled by Eqs. 1 as a fixed-point implementation is used. By *degradation* we mean: (a) the appearance of stable iced points and stable periodic orbits with short periods, inside a floating-point domain of attraction without stable orbits; (b) the attractor itself becomes periodic and its statistical characteristics change, making the system more deterministic. The main contributions of this paper are:

- the analysis of the domains of attraction of the chaotic attractors for a given set of parameters as the number of bits (that encode the decimal part of the number) increases; the appearance of stable fixed points and periodic orbits with short periods are specially considered.
- the determination of the consequent *threshold width* for the bus, in order to make the statistical properties of the digital implementation close to those of the floating-point implementation;
- two different probability distribution functions (PDF) are assigned to evaluate the stochasticity of the time series for different bus widths. Each PDF P is measured by the respective *normalized Shannon entropy* $H(P)$. These entropies have abrupt changes at specific bus widths. Period's lengths and MLE are also evaluated and results are compared with Hs.

This work is organized as follows: first section 2 comments some preliminary concepts and a few remarks on the problem that concern us. Section 3 gives a brief description of the chaotic maps analyzed. Section 4 describes the quantifiers and the method used to study the degradation of the attractors. Section 5 describes our proposed method in detail and emulate fixed-point representation. Then we give experimental results in Section 6. Finally, the conclusions are given in section 7.

2 Problem statement

When iterating chaotic maps in $\mathbb{R}^2$, after a transient that depends on the mixing parameter (r_{mix}), the generated sequence limits in a point or a collection of points called attractor. A chaotic map can have one or more attractors. Attractor domain is called to all the initial conditions (ICs) that converge to each attractor. The ergodic sequences of the attractos, generated by the map, have a determined distribution called Invariant Probability Density Function (IPDF). Main characteristics of chaotic maps, IPDF and r_{mix}, can be obtained by calculating the Frobenius-Perron Operator (FPO) which depends on the map's structure. The fixed points of its spectrum are the invariant densities and they correspond to the eigenvectors with eigenvalue equal to one, the mixing constant corresponds to the second largest eigenvalue of the FPO, [15,16].

When using finite precision, this analysis is not valid, all attractors take the form of fixed points or periodic orbits. The FPO of the map no longer describes the sequences' characteristics. Regarding the attractor domain, it will also change when digitalized, each initial value will be part of, or will converge to, a certain fixed point or periodic orbit. Generally, many new periodic orbits appear, and they change when the number of bits employed varies.

With the adequate precision, periodic orbits of really extended periods can be reached. With the purpose of utilizing them in electronic applications it becomes necessary to understand how the attraction domain evolves with the variation of the bits employed. It is mainly important to know which seeds, i. e. ICs, generate random-like outputs of the system, and also their period's length. Particular attention should be given to the *randomness degree* of the sequences, for this reason, some quantifiers were used here.

Using n bits to represent the state variables of a D- dimensional system the maximum theoretical period T_{max} that can be reached is $T_{max} = 2^{D.n}$. But some periodic orbits with period much lower than T_{max}, which are unstable in a floating-point arithmetic, become stable in fixed-point arithmetic, and viceversa. In principle, the modifications appear to be unpredictable. The appearance of these low period stable orbits represents a *degradation* of the domains of attraction in the sense that certain initial conditions do not evolve toward the pseudo chaotic attractor. Then, to assure the desired pseudo chaotic behavior a threshold in n_{min} exists. Consequently the hardware implementation requires the design of a bus with at least this number of bits n_{min}. In this paper we want to emulate the behavior of a digital hardware implementation, making mandatory to exactly replicate the operation of the device. Our interest is to measure how the domains of attraction degrade with a change in the number of bits n employed, as well as to find the threshold value n_{min}.

Fig. 1. Three attractors for three different sets of coefficients.

3 Chaotic system under study

The family of $2D$ quadratic maps studied here is given by the above equation 1.The $12D$ parameters space generated by coefficients $A = \{a_1, ..., a_{12}\}$ is very hard to be explored. But Sprott discovered that this set of equations produce a huge number of chaotic attractors (about 6.10^{16}) in floating-point arithmetic. Three of these chaotic attractors are shown together in Fig. 1. Their parameters sets A_i are:

a) $A_1 = \{-0.7, \ -0.4, \ 0.5, \ -1.0, \ -0.9, \ -0.8, \ 0.5, \ 0.5, \ 0.3, \ 0.9, \ -0.1, \ -0.9\}$,
b) $A_2 = \{-0.6, \ -0.1, \ 1.1, \ 0.2, \ -0.8, \ 0.6, \ -0.7, \ 0.7, \ 0.7, \ 0.3, \ 0.6, \ 0.9\}$,
c) $A_3 = \{-0.1, \ 0.8, \ -0.7, \ -1.1, \ 1.1, \ -0.7, \ -0.4, \ 0.6, \ -0.6, \ -0.3, \ 1.2, \ 0.6\}$.

Figures 2.a to 2.d show the same three attractors A_1 to A_3 and also the attractor with $A_4 = \{-1, 0.9, 0.4, -0.2, -0.6, -0.5, 0.4, 0.7, 0.3, -0.5, 0.7, -0.8\}$, superimposed with their basins of attraction (in grey). The white areas of each Fig. correspond to those initial conditions generating divergent trajectories of the system.

Fig. 2. Four chaotic attractors and their domains of attraction in floating-point arithmetics. The set of parameters are (see text): (a) $\{a_i\}$ =key 3; (b) $\{a_i\}$ =key 5; (c) $\{a_i\}$ =key 9; (d) $\{a_i\}$ =key 2, [1].

4 Analysis tools

The Maximum Lyapunov Exponent and the Normalized Shannon Entropy applied to two different PDFs along with the mean period length are the quantifiers employed here to estimate the system's properties. These quantifiers help us to evaluate the two properties that determine the randomness degree, the equiprobability among all possible values and the statistical independence between consecutive values.

4.1 Period analysis

As said, one of the most widely uses of chaotic systems is their use as PRNGs, in that case the generated sequences pretend to be true-random, so they should not have a repetition period at all, but of course, this is impossible in a digital environment. Nevertheless by varying the number of bits is possible to achieve extended periods. The utilization of a limited number of bits restricts the quantity of available symbols used to represent the state variables. Therefore, finite arithmetic implies that the implemented system will always have a finite repetition period, and it will happen when all the state variables repeat their values. Accordingly, the best scenario will be when all the combinations occur. Then, the maximum period that can be reached is determined by the quantity of bits of all the state variables, and is: $2^{(\sharp state\_variables*n)}$, where n is the total number of bits employed to represent the values and $\sharp state\_variables$ is the system's dimension.

Actually, the periods obtained are much lower than the maximum and are heavily dependent on the IC.

We have developed a C code that emulates an FPGA operation, it will be described in detail in section 5. One task of this code is to analyze the reached period when starting iteration from each initial condition with a certain number of bits. The initial condition could converge to a limit cycle, or it could be one value of the limit cycle itself. Basically, the code iterates every IC and detects when any value of the generated sequence is repeated, then it stores the period the limit cycle has. This procedure was repeated for all the initial conditions to obtain the attraction domain scheme of the system.

With the developed code, we have systematically studied the behavior of the system's output using different precisions in a fixed-point architecture.

4.2 Quantifiers of Randomness

Another important characteristic that varies with the precision employed is the randomness of the generated sequences.

Based on results of previous research [17–19] the normalized Shannon Entropy (H) was adopted as quantifier to characterize determinism and stochasticity of the generated sequences. This quantifier derives from the Information Theory, and it is a functional of the PDF. By a proper selection of the used PDF it is possible to cover the two mentioned properties, namely, (1) the probability of occurrence of each element of the alphabet (PDF based on histograms), and (2) the order of the items in the time series (PDF based on Bandt-Pompe technique). A discussion about the convenience of using these quantifiers is beyond the scope of this chapter but there is an extensive literature [20,17,21].

Once the PDF is determined the entropy is defined by the very well known normalized Shannon expression:

$$H = \frac{\sum_{i=1}^{K} p_i \; log \; p_i}{log(K)}, \tag{2}$$

Where K is the number of elements of the alphabet.

4.2.1 Defining the PDF

From a statistics point of view a chaotic system is the *source* of a symbolic time series with an alphabet of M symbols. Entropy is a basic concept in information theory. To evaluate entropy one needs first to define a probability distribution function (PDF) of the time series. There is not a unique procedure to obtain this PDF and the determination of the best PDF P is a fundamental problem because P and the sample space are inextricably linked. Several methods deserve mention:

(1) frequency counting [22],
(2) procedures based on amplitude statistics [17],
(3) binary symbolic dynamics [23],
(4) Fourier analysis [24] and,
(5) wavelet transform [25], among others.

Their applicability depends on particular characteristics of the data, such as stationarity, time series length, variation of the parameters, level of noise contamination, etc.

Basically one may consider the statistics of individual symbols or the statistics of sequences of several consecutive symbols. In the first case P is *non-causal* because it does not change if the outcomes are mixed up and the number of different possible outcomes is M (the number of symbols in the source alphabet). In the second case, the outcome changes if the output is mixed and then one says that P is *causal*. In this second case the number of different outcomes is equal to n^M and increases rapidly with n. Bandt and Pompe made a proposal in [26] that is computationally efficient but retains causal effects. In previous works devoted to $PRNG$'s, the use of two PDFs was successful for the comparison between different systems. One PDF is the normalized histogram, and its normalized Shannon entropy is denoted here H_{hist}. The other one is the ordering PDF proposed by Bandt & Pompe [26] and its normalized Shannon entropy is here denoted as H_{BP}. Let us summarize how these PDF's are obtained.

4.2.2 PDF based on histograms

To evaluate the probability of occurrence of each element of the alphabet, it is possible to use the normalized histogram of the time series as a PDF.

If Y is the time series being analysed $Y = \{y_i, i = 1, ...M$. The obvious PDF to characterize Y is the normalized histogram of the K words Y; let us call it

PDF_{hist}.

In order to extract a *PDF* via amplitude-statistics, divide first the interval $[0, 1]$ into a finite number *nbin* of non overlapping subintervals A_i: $[0, 1] = \bigcup_{i=1}^{nbin} A_i$ and $A_i \bigcap A_j = \emptyset \ \forall i \neq j$. One then employs the usual histogram-method, based on counting the relative frequencies of the time series values within each subinterval. It should be clear that the resulting *PDF* lacks any information regarding temporal evolution. The only pieces of information we have here are the x_i-values that allow one to assign inclusion within a given bin, ignoring just where they are located (this is, the subindex i.)

The first step is to normalize the state variables in the interval $[0, 1]$ and define a finite number n_{bin} of non overlapping subintervals A_i: $[0, 1] = \bigcup_{i=1}^{n_{bin}} A_i$ and $A_i \bigcap A_j = \emptyset \ \forall i \neq j$. One then employs the usual histogram-method, based on counting the relative frequencies of the time series values within each subinterval. It should be clear that the resulting *PDF* has no information regarding temporal evolution. The only pieces of information we have here are the x_i-values that allow one to assign inclusion within a given bin, ignoring just the position i where they are located.

4.2.3 PDF based on Band and Pompe methodology

Let x be the source output and let x_1 to x_M be a M-length digital time series. To use the Bandt and Pompe [26] methodology for evaluating of probability distribution P one starts by considering a vector of length D given by:

$$(s) \ \mapsto \ \Big(\ x_{s-(D-1)}, \ x_{s-(D-2)}, \ \ldots, \ x_{s-1}, \ x_s \ \Big) \tag{3}$$

which assign to each time s the D-dimensional vector of values at times $s, s-1, \ldots, s-(D-1)$. Clearly, the greater the $D-$value, the more information on the past is incorporated into our vectors. By the "ordinal pattern" related to the time (s) we mean the permutation $\pi = (r_0, r_1, \ldots, r_{D-1})$ of $(0, 1, \ldots, D-1)$ defined by

$$x_{s-r_{D-1}} \ \leq \ x_{s-r_{D-2}} \ \leq \ \ldots \ \leq \ x_{s-r_1} \ \leq \ x_{s-r_0} \ . \tag{4}$$

In order to get a unique result we set $r_i < r_{i-1}$ if $x_{s-r_i} = x_{s-r_{i-1}}$. Thus, for all the $D!$ possible permutations π of order D, the probability distribution $P = \{p(\pi)\}$ is defined by

$$p(\pi) \ = \ \frac{\sharp\{s | s \leq M-D+1; \ (s), \ \texttt{has type} \ \pi\}}{M-D+1} \ . \tag{5}$$

In this expression, the symbol $\sharp$ stands for "number".

The Bandt-Pompe's methodology is not restricted to time series representative of low dimensional dynamical systems but can be applied to any type of time series (regular, chaotic, noisy, or reality based), with a very weak stationary assumption (for $k = D$, the probability for $x_t < x_{t+k}$ should not depend on t [26]). One also assumes that enough data are available for a correct attractor-reconstruction. Of course, the embedding dimension D plays an important role in the evaluation of the appropriate probability distribution because D determines the number of accessible states $D!$. Also, it conditions the minimum acceptable length $M \gg D!$ of the time series that one needs in order to work with a reliable statistics.

4.3 Maximum Lyapunov Exponent

The fourth quantifier employed is the Maximum Lyapunov Exponent that determines the presence of chaos. The Lyapunov exponents are quantifiers that characterize how the separation between two trajectories evolves, [27]. It is well known that chaotic behaviors are characterized mainly by Lyapunov numbers of the dynamic systems. If one or more Lyapunov numbers are greater than zero, then the system behaves chaotically. Otherwise, the system is stable. In this paper, we employ the maximum Lyapunov number as it is one of the most useful indicators of chaos.

The distance between trajectories changes in 2^{MLE} for each iteration, on average. If $MLE < 0$ the trajectories approaches, this may be due to a fixed-point, if $MLE = 0$ the trajectories keep their distance, this may be due to a limit cycle, if $MLE > 0$, the distance between trajectories is growing, and is an indicator of chaos. [27]

There is a non-analytical way to measure it if only the inputs and outputs of the system are accessible. The procedure is the following: The system must be started from two neighbor points in the phase plane, lets call them (x_a, y_a) and (x_b, y_b), as the system is iterated the Euclidean distance between the two trajectories is measured (d_n in the n_{th} sample) (eq. 7), and the b trajectory is relocalized on each iteration (eq. 8), obtaining the points (x_{br}, y_{br}) to feed the system. Then the Lyapunov exponent can be calculated as shown in eq. (6).

$$MLE = \frac{1}{M} \sum_{i=2}^{M} \log_2 \frac{d_{1(i)}}{d_{0(i-1)}} \tag{6}$$

$$
\begin{aligned}
d_{0(i-1)} &= \sqrt{(x_{a(i-1)} - x_{br(i-1)})^2 + (y_{a(i-1)} - y_{br(i-1)})^2} \\
d_{1(i)} &= \sqrt{(x_{a(i)} - x_{b(i)})^2 + (y_{a(i)} - y_{b(i)})^2}
\end{aligned}
\tag{7}
$$

$$
\begin{aligned}
x_{br(i)} &= x_{a(i)} + (x_{b(i)} - x_{a(i)})d_{o(i-1)}/d_{1(i)} \\
y_{br(i)} &= y_{a(i)} + (y_{b(i)} - y_{a(i)})d_{o(i-1)}/d_{1(i)}
\end{aligned}
\tag{8}
$$

5 Hardware Digital Simulation.

Within the available options for representing values using finite precision, floating-point arithmetic is the closest to $\mathbb{R}$. However, from the engineering point of view the usage of floating-point is not efficient when compared to fixed-point operations because the first ones consume lot of system resources and require several clock cycles. It is widely known that when the maximal values to be represented and the precision required are pre-established fixed-point arithmetic would allow getting better results in terms of velocity, usage resources and power consumption.

The analysis in this paper was intended to cover any digital electronic device such as FPGA, CPLD (Complex Programmable Logic Device) or ASIC (Application Specific Integrated Circuit). On this kind of devices, saving resources is a crucial issue, this is why they mostly employ fixed-point arithmetic.

A C code that simulates iterating a nonlinear system, the quadratic map, in any of such devices was developed in order to generate sequences which were then analyzed. The code is totally parametrizable and it allows to access intermediate values. A technique that emulates operating in fixed-point arithmetic was employed, the general idea is to use signed integer arithmetic, although chaotic systems work with fractional numbers. To solve this, an equivalence between fractional fixed-point numbers and signed integers was employed here.

Of course, internally all digital device works with binary numbers, designers interpret these bits based on the architecture they want to work with. Binary numbers can be interpreted as integer numbers or, as in this case, they can be thought in terms of a fractional point located at a certain position. To illustrate this:

$$
Fractional\_fixed\_point_{value} = -b_{n_i-1}.2^{(n_i-1)} \ldots b_0.2^0, b_{-1}.2^{-1} \ldots b_{-n_f}2^{-n_f}
\tag{9}
$$

where we called n_i to the number of bits used to represent the integer part and n_f the fractional, the whole number of bits is $n = n_i + n_f$.

In order to make this conversion, each fractional number must be multiplied by 2^{n_f} to obtain its equivalent Signed Integer number. Where n_f is the quantity of bits used to represent the fractional part of the number. This is equivalent to right-shift n_f positions the fractional point. Resulting in:

$$Signed\_integer_{value} = -b_{n_i-1+n_f}.2^{(n_i-1+n_f)} \ldots b_0.2^0 \tag{10}$$

An example of the equivalence is shown in Table 1. The following considerations must be taken into account when operating with this equivalence:

- Addition, this operation does not need any consideration just to make sure not to exceed the limits of the arithmetic used.
- Multiplication, the result of this operation must be divided by 2^{n_f} to adjust the result to the correct range.
- Division, the result must always be rounded towards minus infinity. This is, 7.28 to 7 , -14.9 to -15.

After each operation, the corresponding adjustment is performed to operate identically as digital devices work.

For generating the data, the system was intended to be working in fractional fixed-point architecture with 4 bits for representing the integer part, $n_i = 4$, in two's complement representation (Ca_2). The code automatically varies the number of bits representing the fractional part of the number, n_f, in order to analyze how the system reacts when the precision changes.

Table 1 shows the equivalence when using $n = 6$ bits, 2 bits for the integer part and 4 bits for the fractional part ($n_i = 2$ and $n_f = 4$).

The developed code iterates the $2D$-quadratic map 10^5 times, in this case coefficients a_0 to a_{11} have the values:
$\{a_i\} = \{-1.0, 0.9, 0.4, -0.2, -0.6, -0.5, 0.4, 0.7, 0.3, -0.5, 0.7, -0.8\}$.

The map has been iterated with ICs x_0 and y_0 from -2 to 2. They have been swept in steps determined by the n_f employed. For example, when using five bits to represent the fractional part of the number ($n_f = 5$), the minimum value (minimum grid) that can be represented is 0.0063. In the case of using six bits the resulting minimum value is 0.0026, in general when using n_f bits the resulting grid will be:

Table 1
Example of equivalences.

Binary	Fractional Fixed point	Signed Integer
01.1111	1.9375	31
01.1110	1.8750	30
01.1101	1.8125	29
$\vdots$	$\vdots$	$\vdots$
00.0000	0.00	0
11.1111	-0.0625	-1
11.1110	-0.1250	-2
$\vdots$	$\vdots$	$\vdots$
10.0000	-2.00	-32

$$\frac{1}{n_f * 2^{n_f}} \tag{11}$$

On each case it was determined whether the systems evolves to a fixed point, diverges or goes towards a periodic cycle.

For every value of precision n_f the code outputs a square matrix of order $4n_f 2^{n_f}$ whose elements correspond to the final state of the system when initialized with each IC. This means each position will contain one of three values:

- -1, if it diverged,
- 0, if it converged to a fixed point,
- the length of the period, at which that IC converged.

The interesting thing about this program is that it is independent of where it runs, and of the arithmetic used by it (float, double, long double, etc.).

From the point of view of a PRNG implementation, the desirable properties for the system will be to present large periods, few fixed points and, of course, that do not diverge.

6 Results

Figure 3 displays some of the results (i.e. for some values of n_f and $n_i = 4$) obtained, these are the set of final states for each IC (fixed point, divergent

Fig. 3. Coexisting areas in attraction domains for: (a) $n_f = 5$, (b) $n_f = 6$, (c) $n_f = 7$, (d) $n_f = 8$, (e) $n_f = 9$, (f) $n_f = 10$, (g) $n_f = 11$, (h) $n_f = 12$, (i) $n_f = 13$, (j) $n_f = 14$, (k) $n_f = 17$, (l) $n_f = 18$.

point or the length of the period when the initial condition converges to a cycle). So, the different domain attractors (including the attractors) that coexist in the system can be seen in gray tones.

The abscissa and ordinate axis correspond to initial values of x and y respectively.

The different gray tones that can be seen correspond to the period's lengths of the cycles where each IC converges. The clearer the tone the longer the period. With the purpose of being able to distinguish the different coexisting areas, a diverse range of gray tones have been used on each figure. It must be taken into account that each figure has its own gray range, this means that, for example, an almost white area in Fig. 3.a ($n_f = 5$) corresponds to a period of 6, while a darker area in a figure with higher n_f may correspond to a period higher than a thousand (Fig. 3.e). These figures allow reflecting the complex

Fig. 4. Enlarged views of sections of the attraction domains for higher values of n_f: (a) Rectangular section of the attraction domain to be zoomed in; (b) $n_f = 14$ zoom; (c) $n_f = 17$ zoom; (d) $n_f = 18$ zoom.

domains of attraction that appear when digitalizing.

It can be seen in Fig. 3 that the smaller the value of n_f the bigger the area of ICs that tends to diverge and to converge to fixed points. As n_f increases, the area of divergent and fixed points decreases. These figures along with Table 2 allows an easy interpretation of the system's behavior. In Table 2 the period's lengths that appear in the attractor domain for every n_f are sorted by the more to the less numerous ICs converging to a cycle of that length. In parentheses it can be seen the percent of occurrence. Indeed, figures with lower values of n_f present irregular, or rough surfaces, pointing out that different lengths cycles coexist there. For example, for $n_f = 5$ there is a prevalence of short periods cycles. In that case, there exist just two limit cycles, the lighter grey zone corresponds to the attraction domain of the limit cycles of length six, that is the less numerous cycle, according to Table 2, and, the darker zone corresponds to the attraction domain of length two cycle.

Although for $n_f \geqslant 13$ (Figures 3.i to 3.l) the attractor appears to be smooth, however, if a zoom in is done to the figures (Fig. 4) it can be seen that there are still cycles with different periods that coexist in the attractor for $n_f = 14$,17 and 18.

Fig. 5. Period's lengths evolution of the attraction domains for: (a) $n_f = 5$, (b) $n_f = 6$, (c) $n_f = 7$, (d) $n_f = 8$, (e) $n_f = 9$, (f) $n_f = 10$, (g) $n_f = 11$, (h) $n_f = 12$, (i) $n_f = 13$, (j) $n_f = 14$, (k) $n_f = 17$, (l) $n_f = 18$.

When we want to make a general comparison of what happens to the periods when the precisions are varied a color scale is required. See Fig. 5. .

Fig. 5 shows that as the value of n_f increases the color of the area smooths and tends to be lighter, indicating that the CIs converge to higher periods cycles. This means that the range of initial values that generate useful sequences increases for higher values of n_f.

Table 2
Lengths of the periods within the attractor domain x and $y \epsilon [-2, 2]$.

n_f	Period's length (Percentage of ICs that converge to this period's length cycle)
5	2 (92.7%);6 (7.3%)
6	88 (41.6%);44 (36.7%);12 (13.8%);16 (6.2%);2 (0.8%);24 (0.6%);26 (0.2%)
7	12 (83.5%);14 (8.9%);24 (5.2%);34 (1.8%);2 (0.6%)
8	68 (91.7%);14 (6.2%);12 (1.8%);17 (0.2%);15 (0.1%)
9	140 (54.5%);123 (25.4%);34 (8.6%);44 (4.3%);38 (3.9%);22 (2.9%);48;2;12;4 (< 0.1%)
10	655 (78.2%);212 (21.1%);143 (0.5%);12 (0.1%);2;36;13;20;10;4 (< 0.1%)
11	153 (78.1%);461 (10.8%);1381 (8.7%);434 (2.3%);18;30;53;32;34;10;2 (< 0.1%)
12	$2,278$ (64.4%);438 (22.4%);598 (7.6%);886 (4.7%);12 (0.7%);87;2;42;23;32;10 (< 0.1%)
13	$11,510$ (98.9%);1052 (1%);12;26;2;10 (< 0.1%)
14	$21,333$ (69.2%);5.804 (16.5%);$4,795$ (7.9%);$1,264$ (5.8%);$2,429$ (0.5%) 46;23;21;10;12;17 (< 0.1%)
15	$10,099$ (58.6%);1.762 (19.4%);$14,887$ (18.3%);$1,598$ (3.4%);750;105;23;14;2;10 (< 0.1%)
16	$54,718$ (87.5%);$5,017$ (4.7%);$> 10^5$ (3.7%);$5,367$ (2.5%);703 (0.9%) $1,159$;$1,802$ (0.2%);377;75;10 (< 0.1%)
17	$37,812$ (53.1%);$38,456$ (24.1%);$> 10^5$ (16.0%);$34,749$ (3.0%);$3,362$;718 (1.5%) $3,006$,$5,222$ (0.1%);15 (< 0.1%)
18	$> 10^5$ (87.4%);$52,069$ (12.5%);$2,471$ (0.1%);146;51 (< 0.1%)
$float$	$> 10^5$ (100%)

This can also be seen in Table 2, where as n_f increases the predominant limit cycle's length increases. In the limit, when using floating-point architecture, that is the closest arithmetic to real numbers, all the limit cycles are higher than 10^5, they converge to the chaotic attractor seen in Fig. 2.d.

In relation to the randomness quantifiers, we realized that the analysis performed up to this point was not enough to fully describe the changes in the dynamic of a digitalized chaotic system. So we decided to further study the data obtained by employing some statistical quantifiers.

As said in Fig. 3.a the two gray zones correspond to the initial conditions that converge to the two coexisting cycles, of period two and six respectively. Then this two cycles will have a determined value of H_{hist} and H_{BP}, $H_{hist} \mid_{n=2}= 0.0625$, $H_{hist} \mid_{n_f=6}= 0.1199$, $H_{BP} \mid_{n_f=2}= 0.1053$ and $H_{BP} \mid_{n_f=6}= 0.2723$. However, the reported value of these quantifiers can not be the average of both, since the frequency of occurrence of cycle two is much greater than that of cycle six (period two appears 92.7% times while period six only 7.3%, see Table 2). Therefore, we have calculated the average weighing each quantifier by its frequency of occurrence.
Figure 8 shows the weighted average of quantifiers H_{hist}, H_{BP} and MLE. In the figure it can be seen that the three quantifiers tend to the value calculated using floating-point arithmetic. While H_{BP} and MLE stabilize for $n_f \sim 12$ or 13, H_{hist} reaches the theoretical value for $n_f \sim 19$, showing that there are properties of the output sequences that only this quantifier can detect.

The H_{hist}-H_{BP} plane, shown in Fig. 8, allows a quick visualization of the behavior in terms of randomness of the system. Here, again, the system seems

Fig. 6. Summary of initial conditions' behavior: (a) number of fixed points; (b) number of divergent points; (c) logarithm of the length's cycles weighted average; (d) initial conditions with period length higher and lower than $1,000$.

to stabilize for n_f higher than 12.

A summary of the observed analysis of these outputs can be seen in Fig. 6.

Fig. 6.a and 6.b show the number of points that diverge and converge to fixed points respectively as the value of n_f increases, in both cases the final value tends to the floating-point case. It is clear from these figures that for $n_f \sim 12$ the system seems have stabilized. Figure 6.c shows that the averaged period of cycles increases at a logarithmic rate.

Finally, Fig. 6.d shows the number of initial conditions that presents periods T higher and lower than $1,000$. Again, a value of 12 for n_f seems to be the limit to obtain a good approximation of the system.

Table 3 shows the calculated *MLE* for some values of n_f. It can be seen that, as expected, while n_f increases the *MLE* tends to its theoretical value.

Fig. 7. Plane H_{hist} - H_{BP} for different number of bits.

Fig. 8. Weighted average of quantifiers H_{BP}, H_{hist} and MLE as functions of the number of bits.

7 Conclusion

In this work, we have developed a detailed analysis of the changes in behaviour of a $2D$-quadratic map fixed-point implementation. We have found a threshold for the required number of bits where the system keeps the properties of the original (real) one. Our goal is to report the rate of degradation for each property, so as to be used by authors at the time of designing their particular applications. Results show that, compared to floating-point, fixed-point arithmetic executed on an integer datapath has a limited impact on the accuracy. This is interesting because in many applications these maps are intended to be used as controlled noise generators. So, to ensure long periods is required.

Table 3
MLE for different values of n_f.

n_f	*MLE*
11	0.049214459144086
12	0.107498218078192
13	0.139472468153184
14	0.135756935006498
15	0.144155039896011
16	0.137514471652835
25	0.142134613438658
27	0.141180317168284
float	0.142275657734227

Acknowledgment

This work was partially supported by the Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), Argentina (PIP 112-201101-00840), ANPCyT (PICT-2013-2066), UNMDP and the International Centre for Theoretical Physics (ICTP) Associateship Scheme.

References

[1] Julien Clinton Sprott. Automatic generation of strange attractors. *Computers & Graphics*, 17(3):325–332, 1993.

[2] L. De Micco, R. A. Petrocelli, and H. A. Larrondo. Constant envelope wideband signals using arbitrary chaotic maps. *Proceedings of XII RPIC*, 2007.

[3] L. De Micco, R. A. Petrocelli, D. O. Carrica, and H. A. Larrondo. Muestreo catico para la adquisicin de seales de baja frecuencia con ruido de alta frecuencia. *Proceedings de la XII Reunin de Trabajo en Procesamiento de la Informacin y Control*, 2007.

[4] L. De Micco, C. M. Arizmendi, and H. A. Larrondo. Zipping characterization of chaotic sequences used in spread spectrum communication systems. *Institute of Physics Conference Proceedings 913*, pages 139–144, 2007.

[5] Musheer Ahmad, Hitesh Chugh, Avish Goel, and Prateek Singla. *A Chaos Based Method for Efficient Cryptographic S-box Design*, pages 130–137. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[6] Iqtadar Hussain, Tariq Shah, Muhammad Asif Gondal, and Hasan Mahmood. Efficient method for designing chaotic s-boxes based on generalized baker's map and tderc chaotic sequence. *Nonlinear Dynamics*, 74(1):271–275, 2013.

[7] G.A. Constantinides, P.Y.K. Cheung, and W. Luk. Optimum wordlength allocation. In *Field-Programmable Custom Computing Machines, 2002. Proceedings. 10th Annual IEEE Symposium on*, pages 219–228, 2002.

[8] George A. Constantinides, Peter Y. K. Cheung, and Wayne Luk. Wordlength optimization for linear digital signal processing. *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, 22:1432–1442, 2003.

[9] Qun Ding, Jing Pang, Jinqing Fang, and Xiyuan Peng. Designing of chaotic system output sequence circuit based on fpga and its applications in network encryption card. *International Journal of Innovative Computing, Information and Control*, 3:1 – 6, 2007.

[10] M. A. Asseri, M. I. Sobhy, and P. Lee. Lorenz chaotic model using field programmable gate array. *The 2002 45th Midwest Symposium on Circuits and Systems, 2002. MWSCAS-2002*, 1:I – 527–30, 2002.

[11] M.S. Azzaz, C. Tanougast, S. Sadoudi, A. Bouridane, and A. Dandache. Fpga implementation of new real-time image encryption based switching chaotic systems. *Signals and Systems Conference (ISSC 2009)*, pages 1 – 6, 2009.

[12] Celso Grebogi, Edward Ott, and James A. Yorke. Roundoff-induced periodicity and the correlation dimension of chaotic attractors. *Phys. Rev. A*, 38:3688–3692, Oct 1988.

[13] Shujun LI, Guanrong Chen, and Xuanqin Mou. On the dynamical degradation of digital piecewise linear chaotic maps. *Int. J. Bifurcation Chaos*, 15, 2005.

[14] S. P. Dias, L. Longa, and E. Curado. Influence of the finite precision on the simulations of discrete dynamical systems. *Communications in Nonlinear Science and Numerical Simulations*, 16:1574–1579, March 2011.

[15] A. Lasota and M. C. Mackey. *Chaos, Fractals, and Noise: Stochastic Aspects of Dynamics.* Applied Mathematical Sciences 97. Springer Verlag, 2nd. edition. edition, 1994.

[16] A. Lasota and J. A. Yorke. On the existence of invariant measure for piecewise monotonic transformations. *Trans. Amer. Math. Soc.*, 186:481–488, 1973.

[17] L. De Micco, C. M. González, H. A. Larrondo, M. T. Martin, A. Plastino, and O. A. Rosso. Randomizing nonlinear maps via symbolic dynamics. *Physica A*, 387:3373–3383, 2008.

[18] M Antonelli, L De Micco, and HA Larrondo. Measuring the jitter of ring oscillators by means of information theory quantifiers. *Communications in Nonlinear Science and Numerical Simulation*, 2016.

[19] L. De Micco, J. G. Fernández, H. A. Larrondo, A. Plastino, and O. A. Rosso. Sampling period, statistical complexity, and chaotic attractors. *Physica A*, 391(8):25642575, 2012.

[20] O. A. Rosso, H. A Larrondo, M. T. Martin, A. Plastino, and M. A. Fuentes. Distinguishing noise from chaos. *Physs. Rev. Lett., pp154102-154106*, 99, 2007.

[21] M. T. Martín and A. Plastino. Generalized statistical complexity measures: Geometrical and analytical properties. *Physica A*, 369:439–462, 2006.

[22] O. A. Rosso, L. De Micco, H. A. Larrondo, M. T. Martin, and A. Plastino. Generalized statistical complexity measure: a new tool for dynamical systems. *International Journal of Bifurcation and Chaos, Vol. , No. 3 (2010) .*, 20(3):775785, 2010.

[23] K. Mischaikow, M. Mrozek, J. Reiss, and A. Szymczak. Construction of symbolic dynamics from experimental time series. *Phys. Rev. Lett. , YEAR = 1999, volume = 82, number = , pages = 1114-1147, month = , note = , abstract = , keywords = , source = .*

[24] G. E. Powell and I. C.. Percival. A spectral entropymethod for distinguishing regular and irregular motion of hamiltonian systems. *J. Phys. A, YEAR = 1979, volume = 12, number = , pages = 2053-2071, month = , note = , abstract = , keywords = , source = .*

[25] O. A. Rosso, S. Blanco, J. Jordanova, V. Kolev, A. Figliola, M. Schürmann, and E. Başar. Wavelet entropy: a new tool for analysis of short duration brain electrical signals. *Journal of Neuroscience Methods*, 105:65–75, 2001.

[26] C. Bandt and B. Pompe. Permutation entropy: a natural complexity measure for time series. *Phys. Rev. Lett.*, 88:174102–1, 2002.

[27] J Sprott. *Chaos and Time-Series Analysis.* Oxford University Press, 2003.