

FACULTAD DE INGENIERÍA UNIVERSIDAD NACIONAL  
DE MAR DEL PLATA

**Sistemas Complejos, Ruidos  
Discretos y su implementación en  
FPGA**

TESIS

PARA OBTENER EL TÍTULO DE

DOCTOR EN INGENIERÍA CON ORIENTACIÓN EN ELECTRÓNICA

MAXIMILIANO ANTONELLI



*DEDICATORIA*



# Agradecimientos

¡Muchas gracias a todos!



# Índice general

<b>1. Introducción</b>	<b>1</b>
chapter2. Cuantificadores de Aleatoriedad3	
2.1. Máximo Exponente de Lyapunov . . . . .	3
section2.2.Algoritmo evolutivo para la búsqueda de caos4 subsection2.2.Esto-	
do actual del avance8	
2.3. Cuantificadores de la teoría de la información . . . . .	12
subsection2.3.1.Entropía de Shannon y Complejidad Estadística13	
2.3.2. Determinación de la distribución de probabilidad . . . . .	15
subsection2.3.2.Blanos doble entropía y entropía-complejidad18 section2.4.Cuantificador	
de entropías implementado en FPGA21	
2.4.1. <i>Hardware</i> Implementado . . . . .	22
subsubsectionEtapa de Adquisición23 subsubsectionPresentación24	
2.4.2. <i>Software</i> Implementado . . . . .	25
subsection2.4.2.Resultados26	
2.4.4. Discusión . . . . .	27
subsection2.4.4.Conclusiones y trabajo futuro28	
2.5. Dinámica de los ITQ's con AWGN y banda limitada . . . . .	28
subsection2.5.Filtrado digital28	
2.5.2. Resultados . . . . .	30
subsection2.5.2.Discusión35	
<b>3. Generadores de Números Aleatorios</b>	<b>37</b>
section3.1.Sistemas Caóticos37	
3.2. Caos en redes neuronales . . . . .	38
subsection3.2.1.El modelo de Hopfield39	
3.2.2. Un caso de estudio . . . . .	41
chapter4. El problema de la Aritmética Discreta47	
4.1. Analysis of the digital implementation of a chaotic deterministic-	
stochastic attractor (EAMTA 2012) . . . . .	47
subsection4.2.Complexity of switching chaotic maps47	
4.2.1. Introduction . . . . .	47
subsection4.2.1.Information theory quantifiers49	
4.2.3. Results . . . . .	52
subsection4.2.3.Simple maps.52 subsubsectionPeriod $T$ as a function of $P$ and	
$B$ 55	
4.2.5. Conclusions . . . . .	56
chapter5. Conclusiones63	

**A. Field Programmable Gate Array (FPGA)**

2em

**65**

# Capítulo 1

## Introducción

Aqhora arranco con todo...



## Capítulo 2

# Cuantificadores de Aleatoriedad

### 2.1. Máximo Exponente de Lyapunov

El Máximo Exponente de Lyapunov (MLE) caracteriza que tan rápido se apartan dos trayectorias. Si esta velocidad es exponencial, se dice que el sistema es caótico, por lo que este exponente es conocido como un detector de “caotidad”, [?, ?, ?]. Más adelante, el MLE fue utilizado en diversas aplicaciones de muy distintas áreas. Sólo por mencionar alguna, en [?] el MLE es usado para medir una señal muy débil en un gas ideal utilizando criterios caóticos. En [?], se estudia si es posible predecir un cambio en la probabilidad de caída para un modelo simple de caminante humano a partir del *MLE*.

Los exponentes de Lyapunov son quantificadores que caracterizan como evoluciona la separación entre dos trayectorias [?]. En general es bien conocido que el comportamiento caótico está principalmente caracterizado por los números de Lyapunov de la dinámica del sistema. Si uno o mas números de Lyapunov es mayor que cero, entonces el sistema se comporta caóticamente, de otra forma el sistema es estable.

La distancia entre dos trayectorias cambia en  $2^{MLE}$  por cada iteración, en promedio. Si el  $MLE < 0$  las trayectorias se aproximan, esto puede deberse a un punto fijo. Si el  $MLE = 0$  las trayectorias mantienen su distancia, esto puede deberse a un ciclo límite. Si el  $MLE > 0$  la distancia entre las trayectorias es creciente, lo que es un indicador de caos.

Existe una forma no analítica de medir el *MLE* si solo las entradas y las salidas de un sistema son accesibles. El procedimiento es el siguiente: el sistema debe ser iniciado desde dos puntos cercanos en el plano de fase, llamémoslos  $(x_a, y_a)$  y  $(x_b, y_b)$ . A medida que el sistema es iterado se mide la distancia euclídea entre las dos trayectorias ( $d_n$  en la muestra  $n_{th}$ ) (eq. 2.1), y la trayectoria  $b$  es relocalizada en cada iteración (eq. 2.3) obteniendo los puntos  $(x_{br}, y_{br})$  para realimentar el sistema. Entonces, el MLE puede ser calculado como se muestra en la ecuación 2.2. El proceso puede verse en la Fig. 2.1.

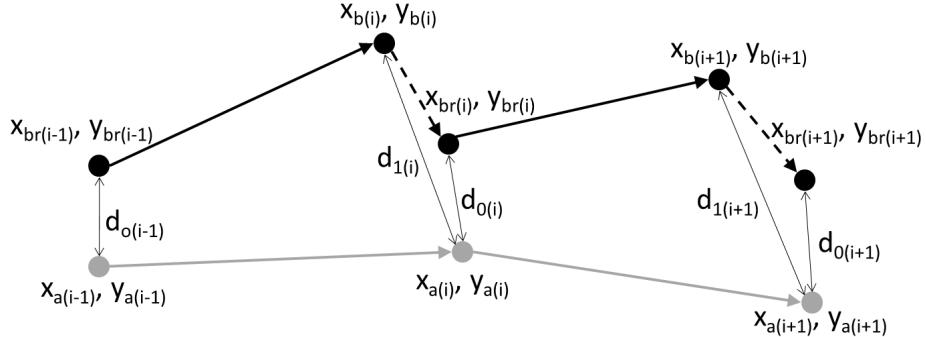


Figura 2.1: Algoritmo para calcular el MLE.

$$\begin{aligned} d_{0(i-1)} &= \sqrt{(x_{a(i-1)} - x_{br(i-1)})^2 + (y_{a(i-1)} - y_{br(i-1)})^2} \\ d_{1(i)} &= \sqrt{(x_{a(i)} - x_{b(i)})^2 + (y_{a(i)} - y_{b(i)})^2} \end{aligned} \quad (2.1)$$

$$MLE = \frac{1}{n} \sum_{i=2}^n \log_2 \frac{d_{1(i)}}{d_{0(i-1)}} \quad (2.2)$$

$$\begin{aligned} x_{br(i)} &= x_{a(i)} + (x_{b(i)} - x_{a(i)})d_{0(i-1)}/d_{1(i)} \\ y_{br(i)} &= y_{a(i)} + (y_{b(i)} - y_{a(i)})d_{0(i-1)}/d_{1(i)} \end{aligned} \quad (2.3)$$

## 2.2. Causal and Non-causal Entropy quantifiers implemented in FPGA

### 2.2.1. Introduction

In the development of new electronic realizations of NonLinear Systems, specially Multiatractor chaotic Systems it is critical to be able to know and monitor the system's properties. Always a thorough analysis of the systems is required to meet the objectives, mainly the changes in the statistical properties induced by discretization of the system. Then it is necessary to check that the system still satisfies the application's requirements, this demands further analysis. Tools coming from the nonlinear analysis like Lyapunov exponents, cross, and autocorrelation, Perron-Frobenius operator, Correlation Dimension are employed, and also statistical tools like the Shannon Entropy, Complexity, Bandt and Pompe embedding.

This work is part of a more ambitious project, which is the hardware development and implementation of tools for the analysis of nonlinear systems. These tools will mean a significant advance in the field of implementation of

## 2.2. CAUSAL AND NON-CAUSAL ENTROPY QUANTIFIERS IMPLEMENTED IN FPGA5

nonlinear systems. It will allow to more accurately understand and describe the behavior of the digital version of this type of systems. The complete package of tools that we intend to implement consists of:

- functional of the probability distribution: Shannon Entropy, Statistical Disequilibrium, and Statistical Complexity;
- time series's quantifiers, especially Lyapunov exponents, autocorrelation, cross-correlation and fractal dimensions;
- Perron-Frobenius operator, quantifiers of recurrent plots;
- Statistical tests proposed in standardized banks for studying random number generators (Marsaglia, NIST, etc.).

At the moment, there is no much literature on hardware implementations of these tools [?].

In the particular case of entropy, it is used in various applications, such as in the anomaly detection of IP data flows [?, ?]. In [?] an FPGA design and simulation of an entropy quantifier is introduced, however currently there are no available hardware implementations of this quantifier.

Within the project mentioned in this paper a system that calculates the entropy of a particular probability distribution (PDF) associated with a data set is implemented. Causal and non-causal PDFs are analyzed. Data can have a digital source (generated by code) or can come from sampling external analog signals. The development board used the *M1AFS-embedded kit*, based on the *M1AFS1500* chip that is known for having an analog block embedded in the same package of the FPGA.

Then, the numerical accuracy of the implemented quantifier is verified by comparing their results with a standard program. The maximum error detected determines the statistical accuracy of the system.

The organization of this paper is as follows: Section ?? both normalized entropies are presented for their consideration; Section 2.4.1 describes the hardware implementation and interfaces; in 2.4.2 the software is described in detail; Section 4.2.3 shows the obtained results of the system validation and in the measurement of the signals. In section 2.5.3, experimental results are interpreted and discussed. Finally, we present our conclusions in Section 2.4.5.

### 2.2.2. Causal and Non causal Entropy

Let  $X = \{x_i, i = 1, \dots, N\}$  of length  $N$  the output of a given source symbol, with alphabet  $\mathcal{A} = \{a_i, i = 1, \dots, M\}$ . Each element of  $X$  is  $x_i \in \mathcal{A}$ .

The most commonly used *PDF* is the normalized histogram of the  $N$  values of  $X$  between the  $M$  symbols of  $\mathcal{A}$ ; it is defined as  $PDF_{hist} = \{p_i, i = 1, \dots, M\}$ , where  $p_i$  is the probability of occurrence of the symbol  $a_i \in \mathcal{A}$ . Its normalized Shannon entropy is referred to as  $H_{hist}$  and is defined:

$$H_{hist} = \frac{\sum_{i=1}^M p_i \log p_i}{\log M} \quad (2.4)$$

The standard entropy  $H_{hist}$  quantifies the distribution of the elements of the series among all possible symbols.

If the source is a Pseudo Random Number Generator (PRNG), then all the symbols of the alphabet should appear the same number of times and its (optimal) value will be  $H_{hist} = 1$ .

$PDF_{hist}$  is non-causal since it does not consider the temporal order of time series' elements. This fact means that the  $X$  vector could be rearranged to generate other vector  $Y$ , which would have the same histogram as  $X$ , and, therefore identical  $PDF_{hist}$  and the same value for  $H_{hist}$ .

For quantifying statistical independence between consecutive elements, in this paper the causal PDF proposed by Bandt & Pompe in [?] is used. This PDF is obtained by assigning patterns of order to overlapped segments of length  $D$  of the time series.

The process for its calculation is the following: first  $D$  consecutive elements are grouped  $\{x_i, x_{i+1}, \dots, x_{i+D}\}$ . Then the (ascending or descending) order of the  $D$  values of each group is compared to the order of the vector  $\{1, 2, \dots, D\}$ . There are  $D!$  Possible ways to sort the numbers  $\{1, 2, \dots, D\}$ . If two values of  $x_i$  within the same group are identical, it is considered that the first is lower, to obtain a unique result. Each permutation is called *ordering pattern*[?].

The normalized histogram of the order patterns is the causal Bandt & Pompe's PDF  $PDF_{BP}$ . The normalized Shannon entropy of that  $PDF_{BP}$  is  $H_{BP}$  where the subscript  $BP$  means "Bandt & Pompe".

Bandt and Pompe suggest  $3 \leq D \leq 7$ . For this work, we adopted  $D = 6$ .

In the  $H_{BP}$  vs.  $H_{hist}$  plane [?], a higher value in any of the entropy values implies an increase in the uniformity of the involved  $PDF$ . The point  $(1, 1)$  represents an ideal case were both distributions, distribution of values and distribution of order patterns, are uniform.

A complete discussion about the convenience of using these quantifiers is beyond the scope of this work. A broad study can be found in [?, ?, ?, ?, ?].

### 2.2.3. Implemented Hardware

The hardware design was based on ACTEL's configuration on the 8051 microcontroller, and peripheral interfaces. It was developed using the software package *Libero Soc v11.3<sup>©</sup>*. The development board used was *M1AFS-EMBEDDED-KIT* that contains an FPGA *M1AFS1500* and peripherals [?].

The *M1AFS1500* embedded chip contains an analog block consisting of nine addressable adapters of four inputs each, a 32 inputs analog multiplexer and a configurable analog-digital converter.

For handling this block, a system also provided by ACTEL is used. This system is based on an 8051 microcontroller, and it also contains drivers of the peripheral among other things [?, ?].

The developed system can be divided into three main stages as shown in Fig. 2.15: the first phase is the Acquisition, which converts the incoming analog signals into digital words. The next step is the Calculation logic stage, it uses the SRAM to perform calculations and coordinate the interfaces, and the last stage is the Presentation which sends the results to a computer through the USB-to-UART interface.

## 2.2. CAUSAL AND NON-CAUSAL ENTROPY QUANTIFIERS IMPLEMENTED IN FPGA7

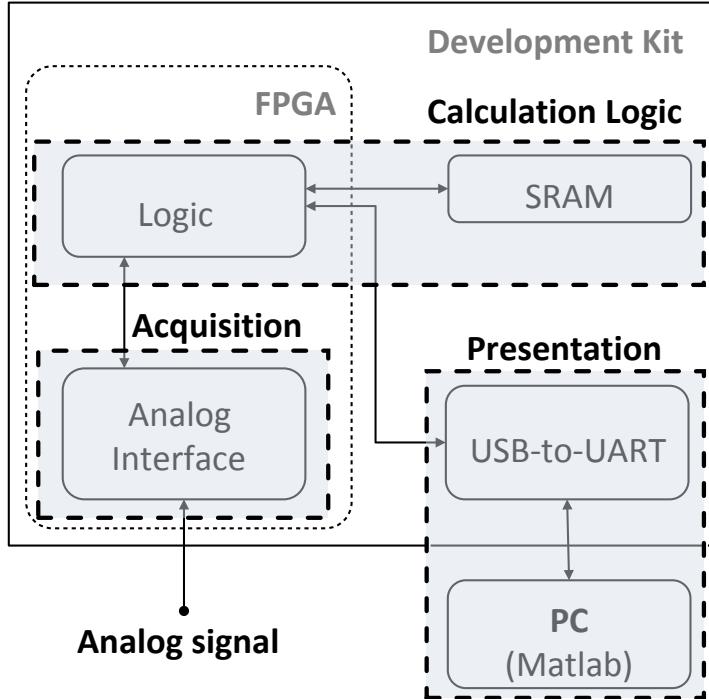


Figura 2.2: Scheme of the complete system.

### Acquisition Stage

The analog data to be evaluated is entered into the system by using the voltage input *AV2* of the analog block *Analog Quad 2*. This input is mapped to channel seven of the analog multiplexer, and it was configured with an input voltage of 0 V to 4 V. The analog-digital converter was configured with a 12-bit resolution. In this first prototype the maximum sample rate achieved was 16 ks/s, limited by the processing logic delay. This speed was enough for the required measurements. However, in a next stage an optimization of the design will be developed by increasing the operating frequency, among other improvements.

### Calculation logic

The calculations and synchronization between peripherals are made at this stage. Figure 2.16 shows the main blocks that compose it.

The heart of implementation is an 8051 Core that provides Actel in its library catalog. It is a microcontroller containing the central logic of the Intel 8051 microprocessor, without its peripherals. This micro has a Harvard architecture with a 16 bits address bus, limiting our design to 64 KB of memory code and 64 KB of data memory.

The application that performs the calculations presented in section ?? runs over this microcontroller. It is responsible for obtaining the PDFs (BP and hist) and to perform the calculations to get the entropies from the input data,

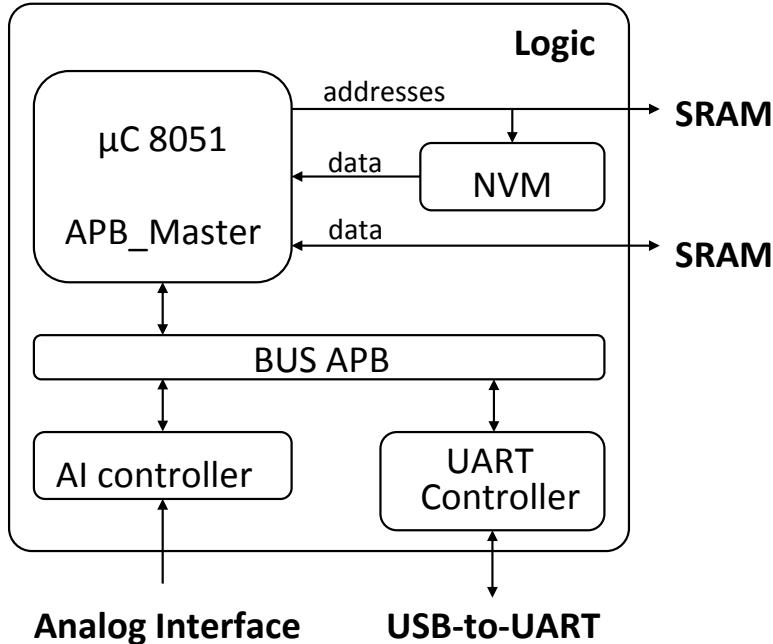


Figura 2.3: Details of the Calculation Logic stage.

according to Eq. ???. Section 2.4.2 describes in detail the developed software. In this particular FPGA, the code memory is a non-volatile memory (NVM) and is implemented in the internal flash of the FPGA blocks. It is mapped onto the addresses from 0x0000 to 0xFFFF and is written during compilation with the contents of a hexadecimal format file.

The system functionality is extended by connecting peripherals through the APB interface.

For handling communication with the PC, the UART controller is used. The output of this block is directed out of the FPGA and is connected to a USB-to-UART chip that is soldered to the board. The analog block is controlled by the AI controller, which routes and synchronizes its inputs.

### Presentation

The stage of data presentation involves the USB-to-UART chip adapter that is on the development board and is driven both by the program running on the FPGA as well as by the software that runs on the PC.

The USB-to-UART chip is responsible for adapting the input-output UART logic to an input-output USB standard by which is possible to interact with the PC. Moreover, the program running on the PC handles the user interface and is described in detail in the next section.

#### 2.2.4. Implemented Software

The system operation is achieved by the interaction of two programs. One running on the PC and another on the microcontroller instantiated in the FPGA.

## 2.2. CAUSAL AND NON-CAUSAL ENTROPY QUANTIFIERS IMPLEMENTED IN FPGA9

It can be seen a flow chart of the two programs and the interaction between them in Fig. 2.17. On the PC runs a Matlab's script that is responsible for opening

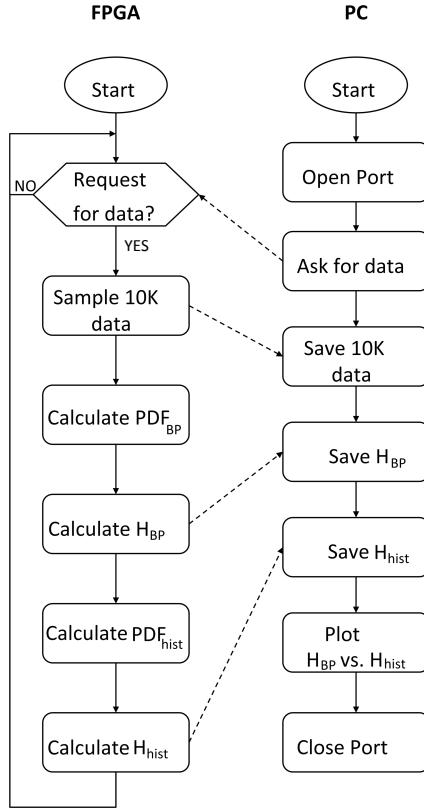


Figura 2.4: Flowchart of the implemented software.

the serial port where the USB is mapped, requesting data, taking the results of the port, then plotting them in a plane  $H_{BP}$  vs.  $H_{hist}$  and finally closing the port.

Over the microcontroller instantiated in the FPGA runs a program written in C language and compiled for the 8051 microcontroller using the *SoftConsole IDE v3.4*<sup>©</sup> tool. The firmware used is a modification of [?]. When a request for data from the UART port occurs, the data sampled in the analog input are stored. Then,  $PDF_{hist}$  y  $PDF_{BP}$  are calculated, and with this information, their respective entropies  $H_{hist}$  and  $H_{BP}$  are also calculated. These results are sent to the PC through the same port.

To validate the system, and verify how accurate the results are, the incoming sampled vector data is forwarded to the PC, so that it can calculate their entropies with PC using *Matlab*<sup>©</sup> and compare them with the results of the implemented system.

Cuadro 2.1: Quantifiers error evaluated in the FPGA with respect to the results calculated by the pattern program.

Generator	Source	Error $H_{BP}$	Error $H_{hist}$
Rand	Digital	$1,7421E^{-6}$	$2,6977E^{-6}$
Logistic	Digital	$0,4256E^{-6}$	$94,693E^{-6}$
Triangular	Analogic	$6,3445E^{-6}$	$2,0028E^{-6}$
Sinusoidal	Analogic	$6,3151E^{-6}$	$5,6506E^{-6}$
Square	Analogic	$0,1797E^{-6}$	$1,9930E^{-6}$
Ramp	Analogic	$245,00E^{-6}$	$1,0876E^{-6}$

### 2.2.5. Results

As said, to test the system, the obtained results were compared with the results achieved by a pattern program running on the PC. For this, 10000 samples of different waveforms were generated by both external (analog) and internal (digital) signals.

Two digital signals were produced by the code in the microcontroller, one corresponds to the rand() C function and the other to the chaotic Logistic map with parameter  $r = 4$ .

Analog signals were generated with the *HP33120A* waveform generator. With a range of 4 Vpp and 2 V of direct current to take advantage of the full spectrum of the analog-digital converter and increase the signal to noise ratio. In all four cases, the signal frequency was 100 Hz and the sampling rate of 16 ks/s.

Table 2.1 shows the absolute error between the calculated results of the quantifiers in the FPGA compared with the results calculated with the standard program running on the PC, over the same samples.

Fig. 2.18 shows in the  $H_{BP}$  vs.  $H_{hist}$  plane the results calculated by the FPGA.

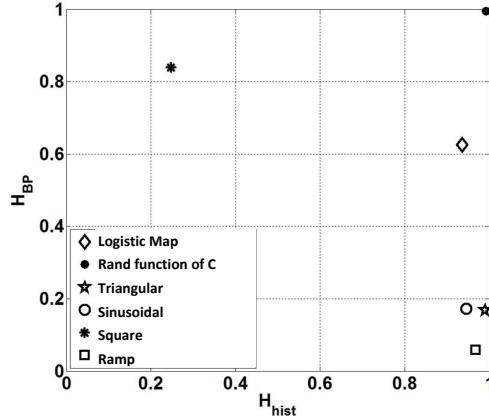


Figura 2.5: Measurement results.

Compilation results show the FPGA resources needed by the entire system and also the amount of memory occupied by the software running on the microcontroller. It is important to highlight that this is a rigid hardware imple-

## 2.2. CAUSAL AND NON-CAUSAL ENTROPY QUANTIFIERS IMPLEMENTED IN FPGA11

mentation, i.e. the first circuit in the FPGA (microcontroller, peripheral, etc.) are configured and then the software is loaded on it. The report returned after running the “Place and Route” tool is shown in Fig. 2.19. It can be seen that the implementation uses 19 % of the FPGA logic resources, 21 % of the input-output cells and 28 % of the memory blocks.

The compilation report of the software part of the system is shown in Fig. 2.20. We can see that the non-volatile flash memory is 15.4 % occupied.

On the other hand, of the 65536 addresses the SRAM have available just 61440 because some of this addresses are used by the APB bus, so the 76,7 % of the available memory is used.

```

Core Cells      : 7349 of 38400 (19%)
IO Cells       : 53 of 252 (21%)

RAM/ROM Usage Summary
Block Rams : 17 of 60 (28%)

```

Figura 2.6: Resources used by the system hardware.

Name	Start	End	Size	Max
PAGED EXT. RAM			0	256
EXTERNAL RAM	0x0000	0xb828	47145	65536
ROM/EPROM/FLASH	0x0000	0x276e	10095	65536

Figura 2.7: Resources used by the system software.

### 2.2.6. Discussion

The software had to be adapted to the microcontroller instantiated in the FPGA. The pattern program uses 64-bit floating point arithmetic (IEEE754-64 bits standard) and uses the math.h library [?]. For adapting the algorithm to be able to implement it in the microcontroller that was instantiated in the FPGA, we had to reduce the number of bits employed. We used 32-bit floating point arithmetic (IEEE754-32 bits standard). The calculation of the logarithm function was also required, so it was implemented using the CORDIC algorithm. These differences make the output of the implemented system not to be equal to that of a program running on the PC, which we take as pattern program. Thus, these differences were measured, to have a dimension and determine whether the results are correct.

Table 2.1 shows that the absolute error never exceeds the value  $245E^{-6}$ . This boundary indicates that there is a difference from the fifth decimal digit.

It can be seen in Fig. 2.18 that  $H_{BP}$  and  $H_{hist}$  quantifiers can clearly difference between the statistical properties of the analyzed data. Sinusoidal, Ramp and triangular signals have the highest values for  $H_{hist}$  because the present all the possible values that the Analog-digital converter can generate.

However, the mixing of these signals are not good because they are periodic signals, so they are totally predictable, this is reflected by the small values of  $H_{BP}$ . An interesting case to be analyzed is the square signal. The additive noise effect is specially marked in the areas where the value of the signal should be constant.

Two very narrow Gaussian curves appear around the ideal values of the  $PDF_{hist}$ , this does not affect too much the calculated value of  $H_{hist}$ , however for  $PDF_{BP}$ , the order pattern is derived directly from the noise, with a particular mean value, so the value of  $H_{BP}$  will be higher than expected.

The signal generated by rand function of C has the best statistical properties being located at the point  $\sim (1, 1)$ .

### 2.2.7. Conclusions and future work

It was developed and implemented a system that allows a measurement with the proper precision of the causal and non-causal entropy of external analog signals from the outside of the FPGA and internal signals generated by code. It was possible to measure signals and perform complex calculations with a small microcontroller as the 8051 instantiated in ACTEL AFS1500 FPGA. This first prototype meets the required specifications of accuracy and the quantity of resources established in the design. The following step will be to optimize the system regarding operating frequency and noise immunity. It is expected that the system will allow modifying, at runtime, the sampling rate, so it would be adaptable to the input signal, with the upper limit of 500 Ks/s set by the ADC.

A threshold level should be added from which a value will be considered different from another, thus, the problem with the additive noise in the calculation of  $H_{BP}$  will be solved. The code for this system occupies 15,4 % of the total flash memory of the instantiated micro, so it will be possible to add software to implement other quantifiers and functionalities. As for the issue of the resources available in the FPGA, the logic cells used is 7349, leaving almost 80 % of the available hardware resources to implement the systems under test, concurrently with the quantifiers.

## 2.3. Measuring the Jitter of Ring Oscillators by means of Information Theory Quantifiers

### 2.3.1. Introduction

*Jitter* is any light deviation from the mean period of a presumed periodic signal. There are many physical examples where jitter is relevant. Some examples from different areas are: (a) Stalberg et al [?] found that the time interval between the two fibre action potentials of two muscle fibers -belonging to the same motor unit in the normal human muscles- shows a variability or jitter; (b) Mecozzi et al [?] detected timing and amplitude jitter in optical links using highly dispersed pulse transmission; (c) Derickson et al [?] made a comprehensive timing jitter comparison in the case of mode-locked semiconductor lasers; (d) the California and Carnegie Planet Search at Keck Observatory [?] reported jitter in stars radial velocities; (e) Roberts & Guillemin studied the delays due to queueing in upstream multiplexing stages, in an Asynchronous Transfer Mode network (ATM); (f) Baron et al [?] considered the quality of the bunch clock signal of the Large Hadron Collider (LHC), in terms of jitter, a fundamental issue because it synchronizes all the electronics systems in the detector; (g) Marsalek et al analyzed the relationship between synaptic input and spike output jitter in individual neurons [?], etc.

### 2.3. MEASURING THE JITTER OF RING OSCILLATORS BY MEANS OF INFORMATION THEORY QUANTIFIERS

Furthermore, digital instruments are used in any modern experiment and the unavoidable jitter in the data acquisition systems produces uncertainties in time, and consequently in any spectrum determination.

This paper is devoted to Ring Oscillators (*RO*). Let us stress that in this particular application jitter is not always undesirable. Jitter is unwanted in applications that use the *RO* as a clock generator [?, ?, ?, ?, ?]. On the contrary random numbers generators *RNG* using *RO*'s, use jitter as the randomness source, [?, ?]. Jitter also improves the Electromagnetic Compatibility as to distribute the clock frequency over a band, improving the Electromagnetic Compatibility (EMC) [?].

Determination of jitter in *RO*'s has been studied in several papers: in [?] the study of three relevant time domain measures of jitter was presented. In [?] a model for jitter generation and distribution in *RO*'s was proposed. In this seminal paper the authors break up the jitter sources into deterministic and random (gaussian); furthermore each source is additionally classified into local or global. They demonstrate that the most important contributions are the local gaussian jitter and the global deterministic jitter and only the first one must be used as a randomness source of true random number generators (*TRNG*'s). The same approach was used in [?, ?, ?, ?]. Lubicz et al. described a practical and efficient method to estimate the entropy rate of a *TRNG* based on free running oscillators; they emphasized that their method does not require outputting and analyzing the clock signals with external equipment [?] (a methodology that introduces extra jitter and distortion in the measured signal due to the data acquisition chain).

Usually *deterministic jitter* is the name given to any *non-Gaussian jitter*. It is bounded and it is characterized by its peak to peak  $\Delta_{pp}$  value. Random jitter is the name used for Gaussian jitter and it is unbounded and characterized by its RMS value. Sometimes deterministic periodic jitter appears. It has a *period* that is the interval between two times of maximum (minimum) effect; the inverse of the time period is the *frequency of the jitter*. Periodic jitter with jitter frequency below 10Hz is usually named *wander* and the name *jitter* is reserved only to periodic jitter with frequencies at or above 10Hz. In communications, the *total jitter* is  $T = \Delta_{pp} + 2 n R_{rms}$  where  $n$  is a number between 6 and 8 related to the Bit Error Rate (*BER*).

*RO*'s are one of the main building blocks in analog and digital integrated circuits and have been extensively used as *on-chip oscillators* to generate clocks in high-speed circuits. Furthermore, *RO*'s can be easily implemented in programmable digital circuits like *FPGAs*. The main advantages of *RO*'s over integrated *LC* oscillators are their smaller chip area, their wider running range (that may be electrically tuned), and their lower power-consumption.

Either one wants to use the *RO* jitter or to eliminate it, jitter must be measured, and it is not a simple task. The main contribution of this paper is to provide a jitter measurement technique based on information theory quantifiers (*ITQ*). We use a stochastic model which randomness is related to the jitter strength. Every proposed *ITQ* used in this paper is based on an entropy, that is a Shannon functional of the probability distribution function (*PDF*) assigned to the time series of the stochastic process. Disequilibrium and complexities may be used as well [?, ?] but they do not represent an improvement in our case. In previous works [?, ?] we showed that many different *PDF*'s can be assigned to the same data string. The best choice depends on the specific application.

Two choices for the *PDF* are used in this paper: the *normalized histogram* and the *ordering patterns histogram*. A representation plane is used to compare different situations. Once a *PDF* is chosen, the Shannon Entropy is the basic functional that quantifies the uniformity of the *PDF*. *Normalized entropies*, *differential entropies*, and *rate entropies* are the other *ITQ*'s evaluated. In our case *differential entropies* give the best results and a *differential entropy plane* is used to compare their sensitivity as a jitter measure.

Organization of the paper is as follows: section ?? describes jitter in *RO*'s and explains how it is measured using random variables; section ?? details the evaluation of the considered *ITQ*; section ?? deals with the results using the proposed quantifiers. Finally, we present our conclusions in sec. ??.

### 2.3.2. Determination of jitter in *RO*'s

There are two different situations concerning jitter in *RO*'s: (a) for some applications it is enough to assure that jitter does not perturb the signal over an accepted limit. If this is the case the signal is observed on an oscilloscope with a mask over the display and it is enough to verify that the signal remains within tolerances; (b) in other cases an exact determination of jitter is required. One of these cases is the characterization of *RO*'s, considered in this paper.

Ideal *RO*'s are composed of an odd number of inverters. Each inverter has a propagation time and consequently rising and falling edges separated by half-periods go through the inverters. If all the propagation times are constant the output of this *ideal RO* is a square-wave with a discrete spectrum. But propagation times are not constant as there is jitter. Jitter distorts the delta like power spectrum as each  $\delta$  is converted into a wider maximum.

Let  $T/2$  be the half-period of the *ideal RO*. It is given by:

$$\frac{T}{2} = k \sum_{i=1}^k d_i \quad (2.5)$$

where  $k$  is the number of inverters and  $d_i$  is the propagation time through the  $i$ -th inverter. When jitter exists,  $d_i$  are random variables that can be modeled as:

$$d_i = D_i + \Delta d_i \quad (2.6)$$

where  $D_i$  is the mean value of  $d_i$  with nominal source voltage level and normal temperature, and  $\Delta d_i$  is the delay variation produced by both local physical events and global changes in the device working conditions (as VCC, temperature, etc.). Then jitter in *RO*'s is evidenced by the random displacement of the trailing (falling) edges from their otherwise perfectly periodic location. The direct measurement of this displacement has two main problems: (a) requires a very high-frequency instrument, because time resolution is limited by the sampling period  $T_s$ ; (b) this technique introduces extra jitter and distortions in the measured signal coming from the data acquisition chain. Then it is more convenient to use *indirect measurements*, by means of auxiliary random variables related to statistical properties related with jitter to measure jitter with minimal disturbance [?]. The general procedure is as follow:

### 2.3. MEASURING THE JITTER OF RING OSCILLATORS BY MEANS OF INFORMATION THEORY QUANTIFIERS

1. Sample the output with sampling period  $T_s$  to get a binary time series. In the ideal case of *no-jitter* the output is a *continuous and perfectly periodic square wave* with period  $T$ . Then it is possible to adjust  $T_s$  to make  $T/2 = m T_s$  with  $m \in N^+$ . The binary time series will be periodic with  $m$  1's followed by  $m$  0's. When jitter is present the binary series is not periodic but stochastic. This stochastic model is known as *alternating renewal process*.
2. Many different randomness quantifiers may be used to characterize the stochastic model associated with the measured jitter. In this paper, we propose the use of *ITQ*'s.

Note that jitter is accumulative and two basic situations arise: (a) if the jitter introduced by each stage is assumed to be totally independent of the jitter introduced by other stages, it means  $\sigma_T^2 = m * \sigma_s^2$ , where  $\sigma_s$  is the jitter of each sample, and it is supposed that all samples have jitter with the same normal distribution; (b) if jitter sources are totally correlated with one another then  $\sigma_T = m * \sigma_s$ .

#### 2.3.3. Information Theory Quantifiers

##### Time series and probability distribution functions

Shannon Entropy is the functional of  $P$  more frequently used in the literature (there are other functionals, like statistical complexity, disequilibrium, etc.). An important issue is  $P$  itself is not a uniquely defined object and do exist several approaches to “associate” a given  $P$  with a given time series. Just to mention some extraction procedures frequently used in the literature: *a)* time series histogram [?], *b)* binary symbolic-dynamics [?], *c)* Fourier analysis [?], *d)* wavelet transform [?, ?], *e)* partition PDF [?], *f)* permutation PDF [?, ?], *g)* discrete PDF [?], etc. There is ample liberty to choose among them and the specific application must be analyzed to make a good choice.

The general procedure to assign  $P$  to a given time series consists in the following steps:

- (a) define an alphabet  $\mathfrak{A} = \{s_j, j = 1, \dots, m\}$
- (b) convert the time series  $X = \{x_i, i = 1, \dots\}$  into a *symbolic sequence*  $A = \{a_i, a_i \in \mathfrak{A}\}$ .
- (c)  $P$  is given by the relative frequencies of the symbols:  $P = \{p_j, j = 1, \dots, m\}$  in the symbolic sequence  $A$ , where  $p_j$  is the relative frequency of symbol  $s_j$ .

$P$  may be *non-causal* or *causal* [?] depending on step b.  $P$  is *non-causal* when one symbol  $s_j \in \mathfrak{A}$  is assigned to each value  $x_i \in X$ . For example, the usual histogram technique used for time series of real numbers corresponds to this kind of assignment. Of course, in this method the temporal order of the time-series plays no role, and consequently the resulting  $P$  will not have any *causal information* and the *symbolic sequence* may be simply regarded as a *coarse-grained* description of  $X$  [?]. It is also possible to group  $W$  consecutive values of the time series –a *trajectory* of length  $W$ – and assign one symbol to the group.

Note that this procedure is equivalent to first assign a symbol to each value of the time series, then group  $W$  symbols into a *word* and finally construct a new alphabet consisting of words. If the original alphabet has  $m$  elements, there will be  $m^W$  possible words and one of this words will be assigned to the *trajectory* of  $W$  elements.  $P$  is given by the relative frequencies of all the possible words. Here  $P$  depends on the temporal order of  $X$  and consequently we call it a *causal P*. It is interesting to note that a *causal P* has information about statistics and also about temporal ordering of  $X$ . If a *non-causal P* is used instead, the analysis must be complemented with the evaluation of the Fourier transform, or the autocorrelation function of  $X$ , to recover the information about temporal ordering.

Let us stress even more the difference between a *non-causal* and a *causal P* by means of the following simple example. Let  $X = \{x_i, i = 1, 2, \dots\}$  be a time series generated by *randn* (Matlab's<sup>©</sup> function); let  $Y = \{y_i, i = 1, 2, \dots\}$  be the resulting series after a sorting process made by *sort* (Matlab's<sup>©</sup> function). Figs. 2.12.a and 2.12.b show the time series. One noncausal  $P$  is the normalized histogram, and  $P(X)$  is identical to  $P(Y)$  as Figures 2.12.c and 2.12.d reveal. One causal  $P$  may be obtained by the Bandt & Pompe procedure (for details about its determination see below) and Figs. 2.12.e and 2.12.f show that  $P(X)$  and  $P(Y)$  are completely different:  $P(X)$  is almost uniform, reflecting  $X$  is randomly ordered, but  $P(Y)$  has a delta-like shape, as far as  $Y$  is monotone increasing and only one ordering pattern is present.

Let us consider now the case of sampled digital signals like as sampled *RO*'s outputs are. Time series  $X$  is binary and has a natural alphabet with two symbols  $\mathfrak{A} = \{0, 1\}$ . The Shannon entropy of this alphabet is usually known as *Binary Entropy*  $S_2$ . Suppose  $W$  consecutive bits of  $X$  are grouped into a *word*, that is the decimal number  $w_i$  between 0 and  $2^W - 1$ ; consider these decimal numbers as the symbols of the new alphabet and let  $Z = \{w_i, i = 1, 2, \dots\}$  be the new symbolic series.  $S_W$  is the entropy of  $P_{hist}(Z)$  where subscript *hist* is for histogram;  $S_W$  is also known as *Block Entropy* of the binary time series  $X$ . Furthermore, if  $D$  consecutive decimal numbers  $w_i$  are grouped again and the  $D!$  permutation patterns are considered as symbols of the new alphabet, we get a new  $P_{BP}(Z)$ , given by the relative frequencies of the permutation patterns. The entropy is  $S_{BP}^{(D)}$ , the *Bandt & Pompe entropy* [?] of the binary time series. All the above-mentioned entropies are given by the same Shannon famous formula:

$$S = - \sum_{j=1}^m p_j \log(p_j), \quad (2.7)$$

and the only difference between them is the  $P$  assigned to the time series. In this paper  $\log$  means *base 2 logarithm*.

In this paper, we use a causal  $P$  determined with the Bandt & Pompe procedure. This procedure has been described in detail and successfully used in a number of papers concerning pseudo random number generation, system classifications, etc. [?, ?, ?, ?, ?, ?]. Let us summarize the basic procedure applied to our specific case:

- let  $Z = \{w_i, i = 1, 2, \dots\}$  be a numerical time series (in our case  $W$  bits decimal numbers) series;
- choose an embedding dimension  $D > 1$

### 2.3. MEASURING THE JITTER OF RING OSCILLATORS BY MEANS OF INFORMATION THEORY QUANTITATIVE

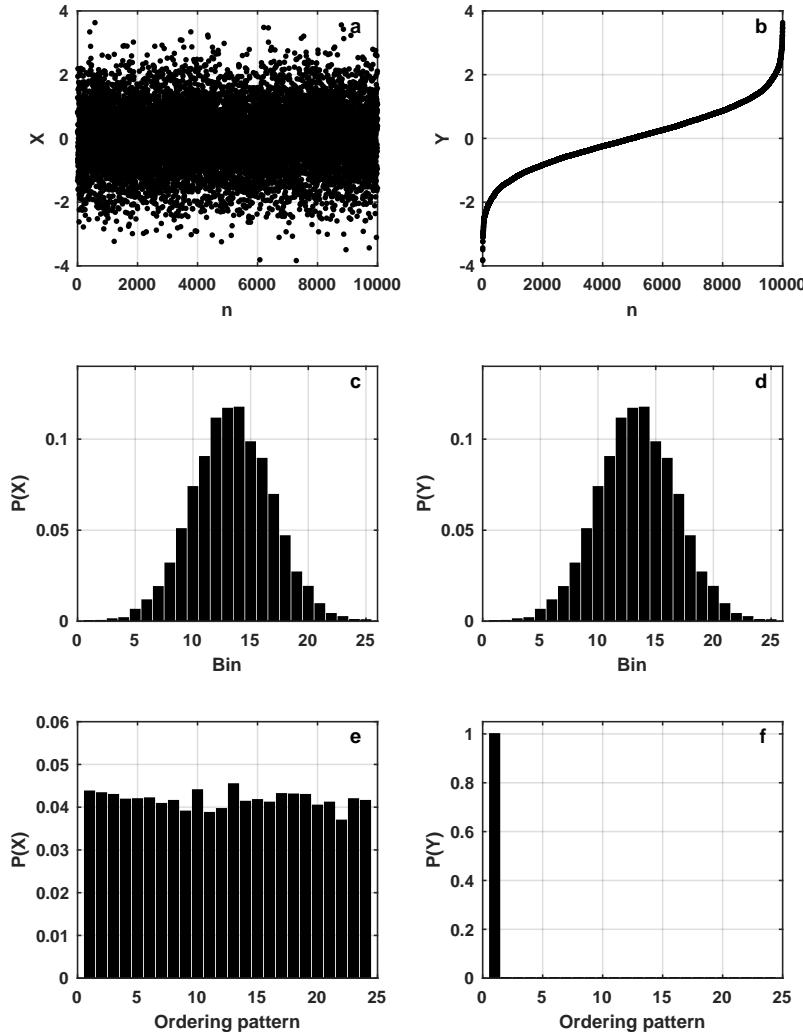


Figura 2.8: (see text) Time series  $X$  obtained by using function `randn` (a), its sorted version  $Y$  (b) and their causal (c, d) and non-causal histograms (e, f).

- assign to each  $w_i$  a  $D$ -dimensional vector of previous  $i, i-1, \dots, i-(D-1)$ :

$$(s) \mapsto (w_{i-(D-1)}, w_{i-(D-2)}, \dots, w_{i-1}, w_i) \quad (2.8)$$

Clearly, the greater the  $D$ -value, the more information about “the past” is incorporated into these vectors.

- look for *ordinal patterns* of length  $D$  [?, ?, ?]. By “ordinal pattern related to position  $i$  we mean the permutation  $\pi = (r_0, r_1, \dots, r_{D-1})$  of  $(0, 1, \dots, D-1)$  defined by

$$w_{i-r_{D-1}} \leq w_{i-r_{D-2}} \leq \dots \leq w_{i-r_1} \leq a_{i-r_0} \quad (2.9)$$

- In order to get a unique result consider that  $r_j < r_{j-1}$  if  $x_{i-r_j} = x_{i-r_{j-1}}$ .
- Thus for all the  $D!$  possible permutations  $\pi$  of order  $D$  is the probability distribution  $P = \{p(\pi)\}$  defined by

$$p(\pi) = \frac{\#\{s | s \leq M - D + 1; i \text{ has type } \pi\}}{M - D + 1} \quad (2.10)$$

In the last expression, the symbol  $\#$  stands for “number” and corresponds to the number assigned to the permutation using the lexicographic order.

The main advantages of the Bandt & Pompe method are *a)* its simplicity, *b)* the extremely fast nature of the pertinent calculation-process, *c)* its robustness in the presence of observational and dynamical noise, and *d)* its invariance with respect to nonlinear monotonous transformations. The Bandt & Pompe 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  $a_i < a_{i+k}$  should not depend on  $i$  [?]).

Let us stress some important issues involved in the calculations of the above-mentioned entropies:

1. The binary entropy  $S_2$  is noncasual while both, the block entropy  $S_W$  and the Bandt & Pompe entropy  $S_{BP}^{(D)}$ , are causal.
2. The block entropy  $S_W$  takes into account correlations between  $W$  consecutive bits. Bandt & Pompe entropy  $S_{BP}^{(D)}$  takes into account correlations between  $D$  consecutive  $W$ -length words. Both grouping procedures (decimal numbers of  $W$  bits and permutation patterns of  $D$  decimal numbers) may be done with or without superposition. The number of data required for good statistics is different depending the grouping procedures are made with superposition or not.
3. For  $S_W$  there is only one grouping process ( $W$  bits are grouped to obtain a decimal numbers time series  $Y$ ). Let us define  $\alpha$  as a statistical quality parameter, given by the quotient between the number of elements in the symbolic time series  $Y$  and the number of symbols in the alphabet. In this paper we will not accept  $\alpha < 10$ .

Obviously the quality factor  $\alpha$  increases with the length of the time series:

### 2.3. MEASURING THE JITTER OF RING OSCILLATORS BY MEANS OF INFORMATION THEORY QUANTIFIERS

- a) If the grouping of  $W$  bits is made with superposition, two consecutive  $W$ -length words share  $W - 2$  bits. Consequently starting with a file with a length of  $N$ -bits we get  $N - W + 1$  words. Furthermore, there are  $2^W$  symbols in the alphabet and  $\alpha = (N - W + 1)/(2^W)$ .
  - b) If  $S_W$  is evaluated without superposition the number of  $W$ -length words is  $\text{floor}\{N/W\}$  and the quality parameter becomes  $\alpha = \text{floor}\{N/W\}/(2^W)$ . For  $N \gg W$  the statistical quality factor is  $W$  times lower than the one with superposition.
4. In the case of  $S_{BP}^{(D)}$ , there are two grouping processes involved.
- a) If both grouping processes are made with superposition we get  $N - W - D + 2$  elements starting with a file  $N$ -bits length, and the quality factor is  $\alpha = (N - W - D + 2)/D!$ . In this case  $S_{BP}^{(D)}$  takes into account the correlations between  $W + D$  consecutive bits.
  - b) If the grouping process of  $W$  bits is made without superposition but the grouping of  $D$  decimal numbers is made with superposition we get  $\text{floor}\{N/W\} - D + 1$  elements and the statistical quality parameter is  $\alpha = (\text{floor}\{N/W\} - D + 1)/D!$ . In this case  $S_{BP}^{(D)}$  will include correlations between  $WD$  consecutive bits.
  - c) If the grouping process of  $W$  bits is made with superposition and the grouping of  $D$  decimal numbers is made without superposition we get  $\text{floor}\{(N - W + 1)/D\}$  elements starting from a file with  $N$  bits. The statistical quality factor is  $\alpha = \text{floor}\{(N - W + 1)/D\}/D!$  and  $S_{BP}^{(D)}$  takes into account correlations between  $W + D - 1$  bits.
  - d) If both grouping processes are made without superposition we get  $\text{floor}\{\text{floor}\{N/W\}/D\}$  elements starting from a  $N$ -bits length file. The statistical quality factor is  $\alpha = \text{floor}\{\text{floor}\{N/W\}/D\}/D!$  and  $S_{BP}^{(D)}$  takes into account correlations between  $WD$  consecutive bits.

#### Additional quantifiers

The Shannon Entropy  $S(P)$  is the startpoint for other quantifiers:

1. Normalized entropy  $H(P)$ : it is the Shannon Entropy divided by its maximum value. For example, if we use  $S_2$  (see above), the maximum entropy is obtained for equiprobability between two symbols. Its value is  $S_{max} = -1/2\log(1/2) - 1/2\log(1/2) = \log(2) = 1$ ; then, the normalized entropy is  $H_2 = S_2$ . If we use  $S_W$  the equiprobability between the  $2^W$  possible words ( $W$ -bits decimal numbers) produces  $S_{max} = W$  and  $H_W = S_W/W$ . Finally for  $S_{BP}^{(D)}$  the equiprobability between the  $D!$  ordinal patterns produces  $S_{max} = \log(D!)$  and  $H_{BP}^{(D)} = S_{BP}^{(D)}/\log(D!)$ .
2. Differential or conditional entropies  $h$  and  $h^*$  are:

$$h = S_{W+1} - S_W \quad (2.11)$$

$$h^* = S_{BP}^{(D+1)} - S_{BP}^{(D)} \quad (2.12)$$

In the above expressions  $W = 1, 2, \dots$  and  $D = 2, 3, \dots$ ,  $S_0 = 0$  and  $S_{BP}^{(1)} = 0$ . These differential or conditional entropies give the average amount of

information required to predict the  $(W + 1)$  (or  $(D + 1)$ ) symbol, given the preceding  $W$  (or  $D$ ) symbols.

3. Finally the *rate entropies*  $h_0$  and  $h_0^*$  are given by:

$$h_0 = \lim_{W \rightarrow \infty} h = \lim_{W \rightarrow \infty} S_W/W \quad (2.13)$$

$$h_0^* = \lim_{D \rightarrow \infty} h^* = \lim_{D \rightarrow \infty} S_{BP}^{(D)}/(D - 1) \quad (2.14)$$

Let us tell in advance that we shall show in section ?? that quantifiers  $S_W$ ,  $S_{BP}^{(D)}$ ,  $H_W$  and  $H_{BP}^{(D)}$  are dependent on parameters  $W$  and  $D$ . This is a drawback if we want to use them as jitter measures. On the other hand, the estimators  $h$  and  $h^*$  of the *rate entropies*  $h_0$  and  $h_0^*$  [?, ?] instead, are independent of  $W$  and  $D$  and we will show in Section ?? that in the case of sampled *RO*'s they also present a minimum for the correct sampling ratio making them good measure of the quality of both *RO*'s and *PRNG*'s derived from them.

### 2.3.4. Results

An evenly sampled output of a jitter-less *RO* was simulated with Matlab<sup>©</sup> and an output file with a length of  $N_b = 7,000,000$  of bits was generated. A set of a hundred values of the sampling ratio  $r = T_s/T \in [6,5,9,5]$ , was explored (where  $T_s$  is the sampling period and  $T$  is the *RO* output period). Jitter with a normal distribution and a set with different values of variance  $\sigma_s$  (see below) were added to the original file. Our method emulates the real process of sampling the noisy output of a real *RO*; the detailed code is published in Mathworks[?].

For each value of  $\sigma_s$ , ten surrogates (each one with a different random initial condition) were generated and new files with  $N_b$  bits each were stored. It was assumed that jitter of individual samples is independent, normal distributed random variables, with zero mean value and variance  $\sigma_i = \sigma_s$ . Consequently, the variance of the accumulated jitter over one period  $T$  is given by  $\sigma_T^2 = r\sigma_s^2$  [?]. The values considered are  $\sigma_T = \{0, 0,001, 0,002, 0,003, 0,004, 0,005, 0,007, 0,01, 0,02, 0,02, 0,04, 0,05, 0,07, 0,1\}$ .

For each file all the quantifiers defined in ?? were evaluated for  $D \in [2, 10]$  and  $W \in [1, 26]$ . The details about evaluation, advantages and drawbacks of each quantifier are reported in section ??: they are  $S_W$ ,  $S_{BP}^{(D)}$ ,  $H_W$ ,  $H_{BP}^{(D)}$ ,  $h$  and  $h^*$ . Let us only show here the more relevant results to show the reason the last two quantifiers ( $h$  and  $h^*$ ) are the best ones.

- In the case of normalized entropy  $H_W$ , it strongly depends on  $W$ . Furthermore the analysis of  $H_W$  as a function of  $r$  shows that it does not allow to determine an optimum value of the sampling ratio  $r$  (see Fig. ??). This is an important issue if the quantifiers are going to be used for experimental setups.
- In the case of the normalized Bandt & Pompe entropy  $H_{BP}^{(D)}$ , a strong dependence on the embedding dimension  $D$  is additionally present. Again it is not easy to determine the optimum value of  $r$  from the analysis of this parameter as a function of  $r$  (see Fig. ??).

### 2.3. MEASURING THE JITTER OF RING OSCILLATORS BY MEANS OF INFORMATION THEORY QUANTIFIERS

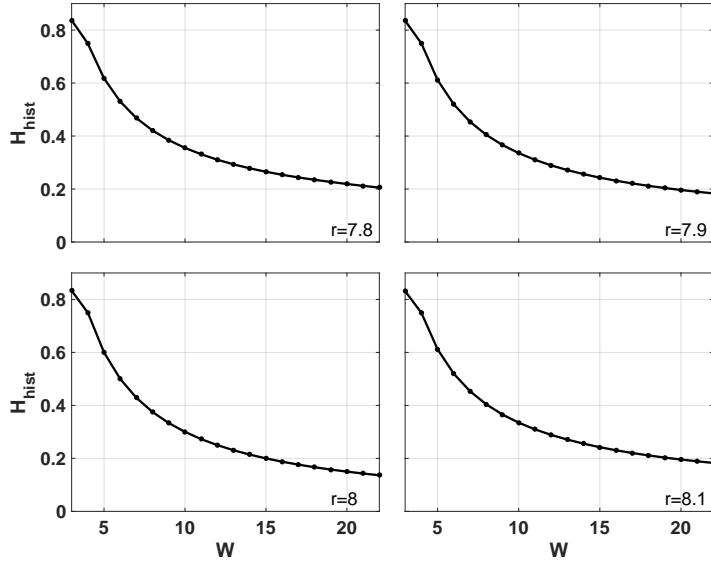


Figura 2.9: Normalized entropy  $H_W$  as a function of  $W$  for a jitter-less RO sampled with different values of  $r$ .

- A similar behavior appears in all the other functionals related with these two entropies. In summary, our results show that both  $h$  y  $h^*$  are independent of any arbitrary parameter used in their statistical determination. These two quantifiers have also been considered in two excellent articles [?, ?].

Our results show that two quantifiers,  $h$  and  $h^*$ , are appropriate to be used as jitter measures because:

- (a) for  $\sigma_T = 0$  (jitter-less output) they rapidly approach to a constant limiting value as both  $D$  and  $W$  increase toward  $\infty$  and this limiting value is independent of both  $D$  and  $W$ ;
- (b) they are increasing monotone (and almost proportional) functions of  $\sigma_T$ .
- (c) From their analysis, it is possible to detect the optimum value of the sampling ratio  $r$ . Let us show these claims in the following figures that are representative of all our results.

Figure ?? shows the Bandt & Pompe differential entropy  $h^*$ , as a function of  $D$ , with  $W$  as a parameter, for a ring without jitter. It can be seen that there is a threshold value  $W = 4$  over which all the curves collapse into one for every value of  $D$ . Furthermore, Fig. ?? also shows that for  $D \geq 8$  all the curves collapse into one, regardless the value of  $W$ . In conclusion, if  $D \geq 8$  and  $W \geq 4$  one obtains a quantifier independent of both  $D$  and  $W$ . The influence of jitter on this quantifier is shown in Figure ??, where  $h^*$  is plotted as a function of  $D$  with  $\sigma_T$  as a parameter. The values considered are  $\sigma_T = \{0(\text{no jitter}), 0,001, 0,002, 0,003, 0,004, 0,005, 0,007, 0,01, 0,02, 0,04, 0,05, 0,07, 0,1\}$ . The inset

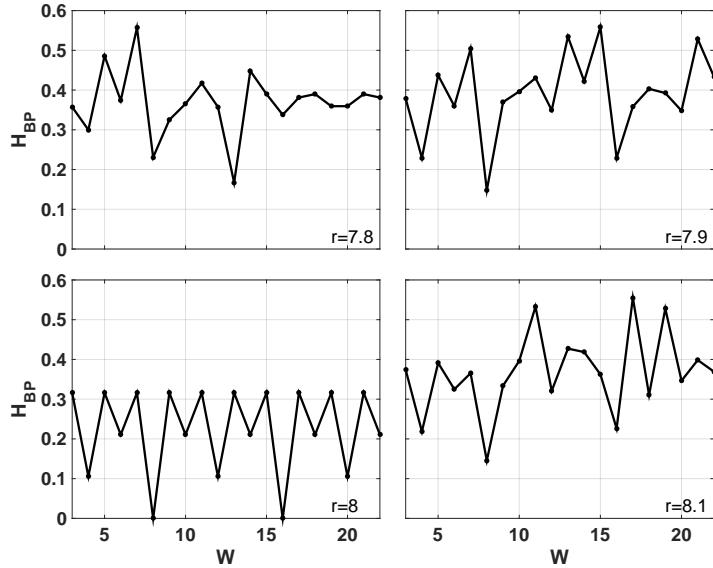


Figura 2.10:  $H_{BP}^{(D)}$  as a function of  $W$  for a jitter-less  $RO$  sampled with different values of  $r$ . Calculations are made without superposition of words

of Fig. ?? shows  $h^*$  as a function of  $\sigma_T$  for  $D = 8$ . This inset shows that this quantifier is an increasing monotone function of  $\sigma_T$ . Finally Fig. ?? shows  $h^*$  as a function of the sampling ratio  $r$ . In this figure, it is shown that there is a minimum for the right  $r$  (in this case  $r = 8$ ). Furthermore sensitivity of  $h^*$  as a function of jitter is maximum for this same ideal value of  $r$ .

Let us now analyze the second quantifier,  $h$ . This quantifier only depends on  $W$  because  $D$  is not used to define the  $PDF$  assigned to the data series. Fig. ?? shows jitter-less case,  $h$  is almost independent of  $W$  for  $W \geq 4$ . In this paper, we adopted  $W = 6$ . Figure ?? shows the influence of jitter over this quantifier. It is clear from the inset in this figure that, for the selected value  $W = 6$ ,  $h$  is an increasing monotone function of jitter variance  $\sigma_T$ .

Fig. ?? shows that  $h$  has a minimum when the value of  $r$  takes its optimum value ( $r = 8$ ). Note that this minimum is robust also in the presence of jitter.

Further analysis must be done to assure that the selected values  $W = 6$  and  $D = 8$  produce symbolic files with a good statistics. For a given alphabet  $\mathcal{A}$  with  $m$  elements, and a given symbolic file of length  $n$ , the quality parameter  $\alpha = n/m$ , see ???. Quality is better as  $\alpha$  increases and a minimum value  $\alpha = 10$  was accepted. According to section ?? the selected values  $W = 6$  and  $D = 8$  provide  $\alpha_h \simeq 10^5$ ,  $\alpha_{h^*} \simeq 175$  with superposition and 29 without superposition. All cases give  $\alpha > 10$  as required.

A comparison between both quantifiers is shown in Figure ???. Markers correspond to variances  $\sigma_T = \{0, 0,001, 0,002, 0,003, 0,004, 0,005, 0,007, 0,01, 0,02, 0,03, 0,04, 0,05, 0,07, 0,1\}$ . Note that the slope of any of these curves is  $dh^*/dh$  and it is equal to the quotient between slopes of curves in the insets of Figs. ??, and ???. If  $dh^*/dh > 1$ ,  $h^*$  is more sensitive than  $h$  to measure jitter. The slope slightly increases from  $\sim 2,47$  for  $W = 5$  to  $\sim 5,54$  for  $W = 19$  showing that  $h^*$  becomes more sensitive as  $W$  increases.

### 2.3. MEASURING THE JITTER OF RING OSCILLATORS BY MEANS OF INFORMATION THEORY QUANTITATIVES

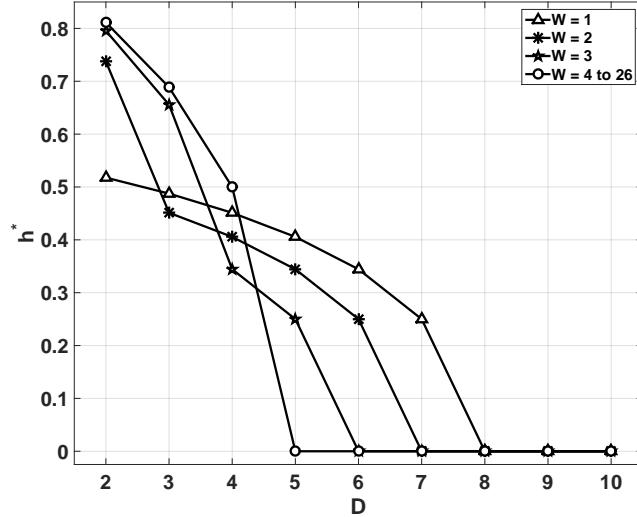


Figura 2.11:  $h^*$  as a function of  $D$  for a jitter-less  $RO$  sampled with  $r = 8$ .

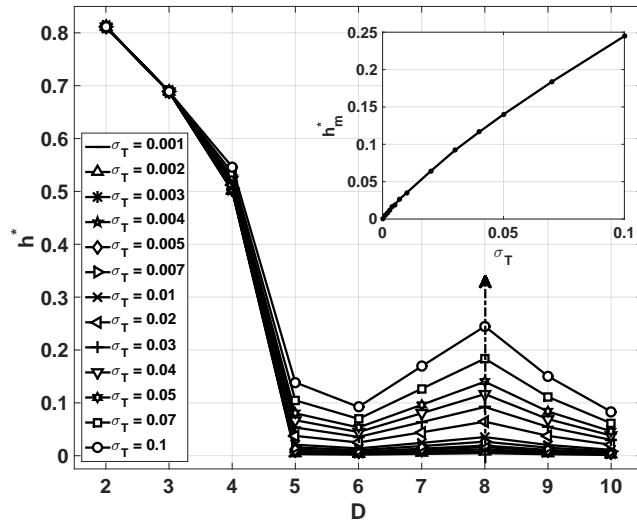


Figura 2.12:  $h^*$  as a function of  $D$  for a  $RO$  sampled with  $r = 8$  for a word length  $W = 6$  for jitter with several variances. The inset shows  $h^*$  as a function of  $\sigma_T$  for  $r = 8$ ,  $W = 6$  and  $D = 8$ .

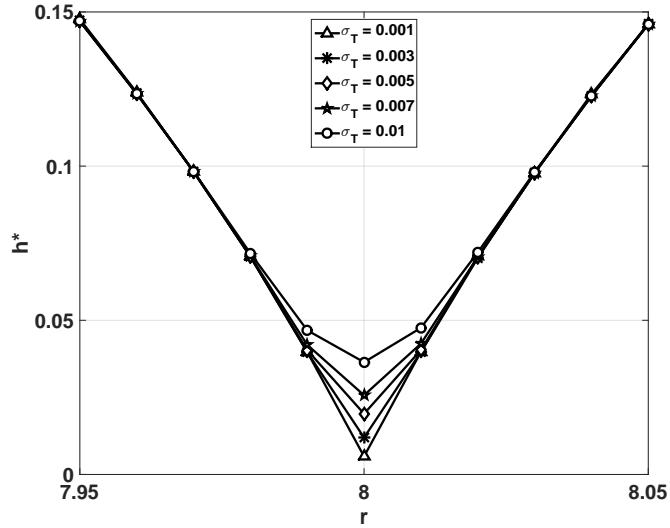


Figura 2.13:  $h^*$  as a function of  $r$  for  $r \in [7.95, 8.05]$ , with several  $\sigma_T$ ,  $W = 6$  and  $D = 8$ . The curve has a minimum at the correct value  $r = 8$ .

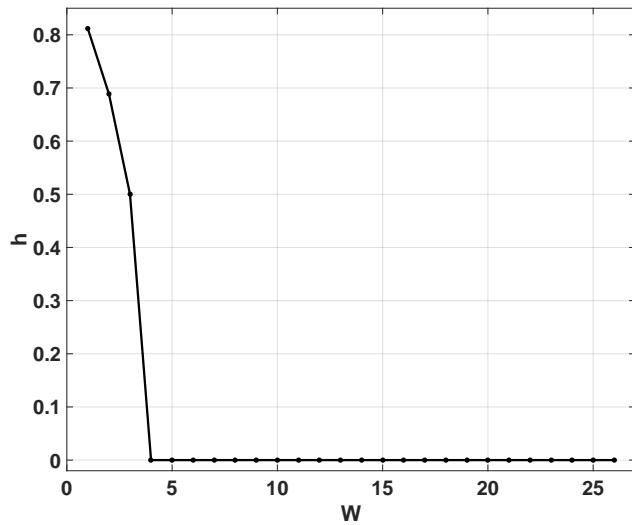


Figura 2.14:  $h$  as a function of  $W$  for a jitter-less  $RO$  sampled with  $r = 8$ .

### 2.3. MEASURING THE JITTER OF RING OSCILLATORS BY MEANS OF INFORMATION THEORY QUANTITATIVE

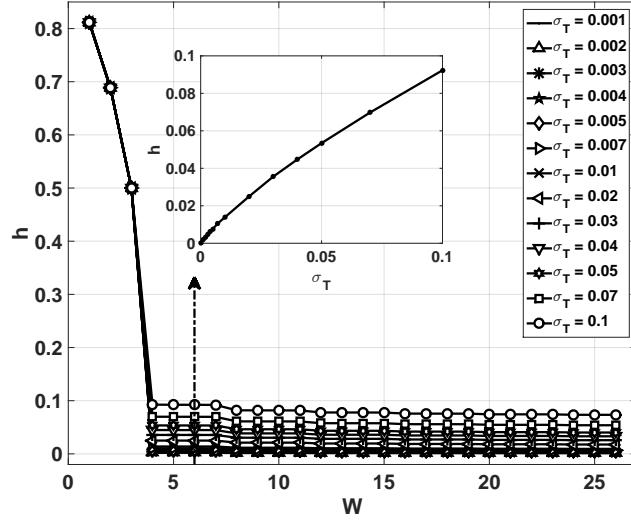


Figura 2.15:  $h$  as a function of  $W$  for a  $RO$  sampled with  $r = 8$ , for jitter with several variances. The inset shows  $h$  as a function of  $\sigma_T$  for  $r = 8$  and  $W = 6$ .

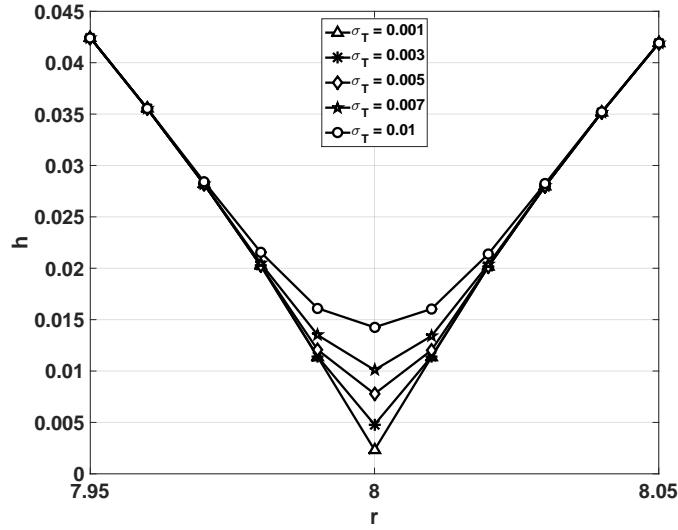


Figura 2.16:  $h$  as a function of  $r$  for  $r \in [7.95, 8.05]$ , with several  $\sigma_T$  and  $W = 6$ . The curve has a minimum at the optimum value  $r = 8$ .

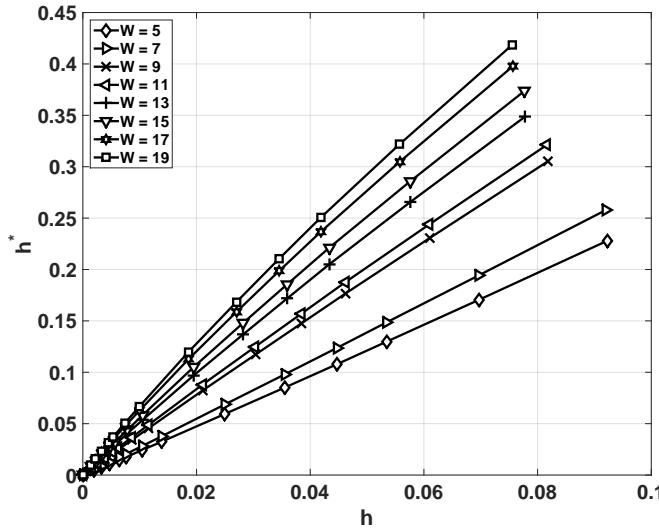


Figura 2.17:  $h^*$  as a function of  $h$  for  $r = 8$ ,  $D = 8$  and different values of  $W$ .

We also evaluated  $h^*$  without the superposition of bits between consecutive natural numbers but keeping the superposition of  $D - 1$  natural numbers between ordering patterns (In all cases  $h$  was evaluated with superposition of  $W - 1$  consecutive bits). Results are depicted in Fig. ?? where it is shown that removing the superposition the sensitivity of this quantifier increases. Of course, we get a smaller amount of  $W$  bits natural numbers form the original seven million binary file, and consequently, the statistical quality is lower than that of the original calculation with superposition. To increase  $\alpha$  up to its previous value, longer binary files are required.

### 2.3.5. Conclusions

Given their usefulness as *PRNG* and clock generators, *ROs* are becoming one of the main building blocks of digital circuits. Jitter is unavoidable in *ROs*, and consequently, it needs to be characterized. Mixing and distribution of values are the main properties to consider. Several *ITQ* quantifiers were evaluated here.  $S_W$ ,  $S_{BP}^{(D)}$ ,  $H_W$  and  $H_{BP}^{(D)}$  turn out to be dependent on parameters  $W$  and  $D$ . This is a drawback if we use them as jitter measures. On the other hand, it is no possible to calculate *rate entropies*,  $h_0^*$  and  $h_0$ , since an infinite number of data is necessary for their calculation. The two *differential entropies*,  $h^*$  and  $h$ , instead, are independent of the parameters used for their determination and are estimators of the *rate entropies*. We have shown in Section ?? that in the case of sampled *ROs* they also present a minimum for the correct sampling ratio making them a good measure of the quality of both *RO*'s and *PRNG*'s derived from them.

The dual entropy plane determined by these quantifiers has demonstrated to satisfactorily discern between the *PRNG*'s two main desired properties, the equi-probability among all possible values and the statistical independence between

### 2.3. MEASURING THE JITTER OF RING OSCILLATORS BY MEANS OF INFORMATION THEORY QUANTITATIVE

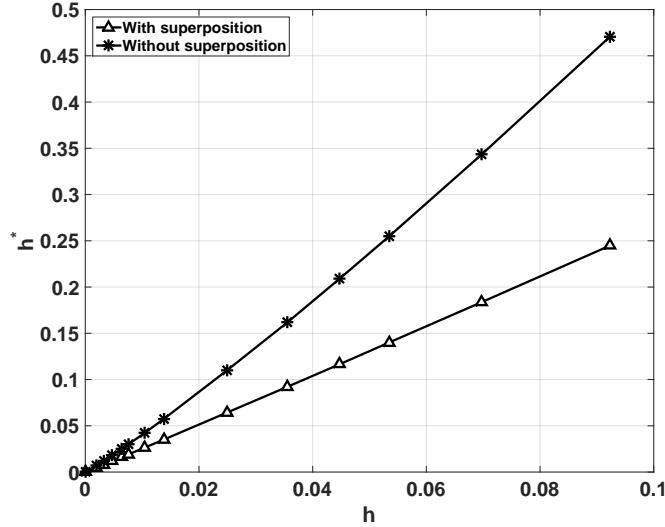


Figura 2.18:  $h^*$  as a function of  $h$  for  $r = 8$ ,  $W = 6$  and  $D = 8$ . Two procedures to obtain  $W$ -bits natural numbers are considered: with and without superposition (see text).

consecutive values. Thus, it allows clearly seeing what needs to be improved in a given sequence. The examples presented here have demonstrated the need to use both histograms for characterizing sequences.



## Capítulo 3

# Generadores de Números Aleatorios

### 3.1. Sistemas Caóticos

Ha quedado claro que existen sistemas deterministas que rompen con el preconcepto de que los sistemas físicos pueden clasificarse en dos conjuntos disjuntos: sistemas deterministas y sistemas estocásticos. En esa concepción antigua un sistema determinista es aquél para el cual conocemos el modelo y por lo tanto es posible predecir con exactitud la evolución de sus variables de estado. Se utilizan en su descripción ecuaciones diferenciales o de recurrencia. Por otra parte un sistema estocástico es aquél para el cual el modelo no se conoce o se lo supone sumamente complejo como para ser obtenido, de modo que se adopta la estrategia de estudiar sus variables de estado en forma estadística. Se utilizan entonces en la descripción ecuaciones diferenciales o de recurrencia estocásticas.

El caos determinista demostró que complejidad en la evolución temporal no es sinónimo de complejidad en el modelo, cuando hay no linealidad: modelos deterministas muy simples originan señales de aspecto estocástico. La sensibilidad a las condiciones iniciales hace que en estos sistemas la predictibilidad sea a corto plazo (luego de un tiempo finito es imposible predecir la evolución) lo que ubica a estos sistemas en una posición intermedia entre determinista y estocástico.

Como consecuencia se desarrollaron en los últimos años un número creciente de aplicaciones de los sistemas caóticos, empleándolos principalmente como generadores de ruido controlado, generadores de números pseudoaleatorios, portadoras de señales, sistemas de encriptado, etc.

Hoy en día, los sistemas dinámicos son un objeto de estudio interdisciplinario, aunque originalmente fue una rama de la física. Todo comenzó a mediados del 1600, cuando Newton inventó las ecuaciones diferenciales, descubriendo sus leyes del movimiento de gravitación universal, y las combinó con las leyes de Kepler sobre el movimiento planetario. Específicamente, Newton resolvió el problema de los dos cuerpos (por ejemplo el sistema tierra-sol).

Subsecuentes generaciones de matemáticos y físicos intentaron extender los métodos analíticos de Newton al problema de los tres cuerpos (por ejemplo luna-

tierra-sol), pero curiosamente para resolver este problema se necesitó mucho más esfuerzo. Luego de décadas de esfuerzo, se dieron cuenta de que el problema de los tres cuerpos era esencialmente imposible de resolver, en el sentido de obtener las fórmulas explícitas.

La ruptura vino con el trabajo de Poincaré a finales del 1800. Él introdujo un nuevo punto de vista que enfatizaba las cuestiones cualitativas más que las cuantitativas (por ejemplo, ¿es estable el sistema luna-tierra-sol?). Poincaré desarrolló una poderosa aproximación geométrica que es usada hoy para estudiar sistemas dinámicos y también fue el primero en vislumbrar la posibilidad de caos, en el cual un sistema determinístico exhibe un comportamiento aperiódico que depende sensiblemente de las condiciones iniciales, haciendo así imposible la predicción a largo plazo.

Pero el caos se mantuvo en segundo plano hasta la segunda mitad del 1900, en donde los osciladores no lineales jugaron un rol vital en el desarrollo de tecnologías de radio, radar, lazos de enganche de fase y láser. Por el lado matemático, los osciladores no lineales también estimularon la invención de nuevas técnicas matemáticas. Los métodos geométricos de Poincaré se fueron extendiendo para producir un conocimiento mucho más profundo de la mecánica clásica.

La invención de la computadora por el 1950 fue una línea divisoria en la historia de los sistemas dinámicos. La computadora nos permite experimentar con ecuaciones en una forma que antes era imposible, y así desarrollar alguna intuición acerca de los sistemas no lineales. Estos experimentos llevaron a Lorenz a descubrir en 1963 el movimiento caótico de un atractor extraño, mientras estudiaba un modelo simplificado de la circulación de convección para comprender mejor la notoria impredecibilidad del clima. Lorenz encontró que la solución a sus ecuaciones nunca caían al equilibrio o a un estado periódico. Además, si comenzaba sus simulaciones de dos condiciones iniciales ligeramente diferentes, los comportamientos resultantes pronto serían totalmente diferentes. Como consecuencia de ello, el sistema es inherentemente impredecible, pequeños errores en las mediciones del estado actual de la atmósfera (o cualquier sistema caótico) sería amplificado rápidamente. Pero Lorenz también mostró que había estructura en el caos, cuando fueron ploteadas en tres dimensiones, las soluciones a sus ecuaciones cayeron sobre un set de puntos en forma de mariposa. Él sostuvo que este sistema tenía que ser “un infinito complejo de superficies”. Lo que hoy podríamos considerar como un ejemplo de fractal.

El trabajo de Lorenz tuvo un pequeño impacto hasta 1970, los años del boom del caos. Se desarrollaron teorías completamente nuevas basadas en consideraciones sobre atractores caóticos, como turbulencia de fluidos y biología de las poblaciones y se encontraron comportamientos caóticos en reacciones químicas, circuitos electrónicos, osciladores mecánicos, semiconductores y oscilaciones biológicas como el ritmo cardíaco y circadiano.

Hoy, la teoría del caos es un herramienta más para el estudio de sistemas dinámicos y los sistemas caóticos forman parte de una gran cantidad de dispositivos.

### 3.2. Caos en redes neuronales

El problema de el caos en las redes neuronales ha recibido mucha atención recientemente. Las actividades en este campo pueden ser divididas en tres ca-

tegorías:

- Intentos por explicar experimentalmente el comportamiento aperiódico observado en una sola neurona perturbada o en un pequeño ensamble de neuronas.
- Intentos por explicar el comportamiento temporal complejo del cerebro y los posibles roles del caos en el procesamiento de la información.
- El estudio de las rutas al caos y las propiedades de los atractores caóticos en modelos de redes neuronales.

Las redes neuronales artificiales proveen soluciones efectivas a problemas en diversos campos, en particular, pueden servir como generadores de señales caóticas. Las aplicaciones de señales caóticas son muy diversas, pero en este caso son especialmente atractivas ya que en los algoritmos de aprendizaje se utiliza una búsqueda aleatoria, en estos casos un generador neuronal de caos puede ser una parte de la red neuronal determinística que se está entrenando.

### 3.2.1. El modelo de Hopfield

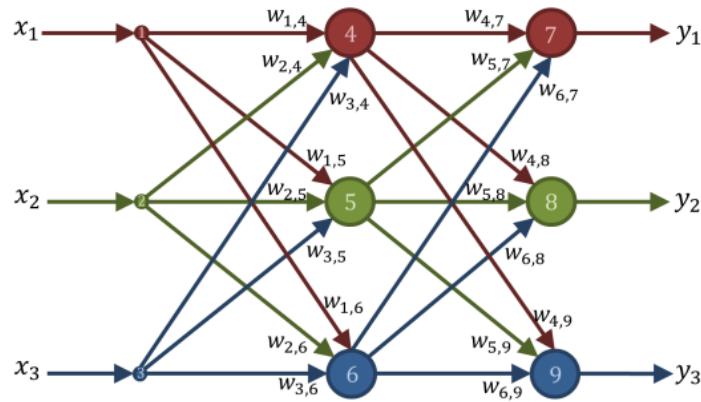
Una de las piedras fundamentales para el reciente renacimiento en el campo de las redes neuronales fué el modelo asociativo propuesto por Hopfield en 1982. La aproximación de Hopfield es un enfoque teórico para pensar ensambles entre unidades de cómputo.

El perceptrón multicapa es una RNA formada por capas de neuronas. Las neuronas pueden pertenecer a la capa de entrada, capas ocultas o capa de salida. Estas neuronas no incorporan memoria por lo que su salida depende del estado de sus entradas en el instante actual (no tienen retardo), además, como el nombre de sus capas lo sugiere, las conexiones son hacia adelante. Es por esto que la matriz de pesos tiene solo algunos valores distintos de cero, no hay conexiones hacia atrás, ni en la misma capa, ni sobre la misma neurona, ni salteándose capas. En la figura ?? se ve un ejemplo para un perceptrón pequeño y su matriz de pesos.

Al contrario de los preceptrones multicapa ,los sistemas adaptativos y los mapas autoorganizados, las redes de Hopfield si tienen realimentación entre neuronas. Este tipo de arquitectura tiene como campo principal de aplicación la optimización de procesos. Se basa en el planteamiento de una memoria asocialitiva; se hace necesario entonces definir una función de energía. Pone de manifiesto la analogía existente entre su modelo y la física estadística clásica, lo que permite usar sus bien conocidas herramientas matemáticas. Además, es muy interesante que se destaca la facilidad de implementación en FPGA y VLSI.

Esta red recurrente se basa en almacenar información en un sistema que presenta una configuración dinámica estable, es decir, se plantea como una memoria asocialitiva o memoria direccionable por contenido. Intuitivamente, la idea de Hopfield es localizar cada patrón que se requiere almacenar a la red en el fondo de un valle de la función de energía. Se parte de un determinado estado inicial (información de partida) tras lo cual se deja evolucionar el sistema hasta llegar a un estado estable. Este estado estable será el patrón que se corresponde con nuestro estado inicial (reconocimiento de patrones).

Hopfield, en su trabajo destaca tres diferencias con el perceptrón multicapa:



$\begin{matrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{matrix}$	$\begin{matrix} w_{1,4} & w_{1,5} & w_{1,6} \\ w_{2,4} & w_{2,5} & w_{2,6} \\ w_{3,4} & w_{3,5} & w_{3,6} \end{matrix}$	$\begin{matrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{matrix}$	Vienen de la capa 1
$\begin{matrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{matrix}$	$\begin{matrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{matrix}$	$\begin{matrix} w_{4,7} & w_{4,8} & w_{4,9} \\ w_{5,7} & w_{5,8} & w_{5,9} \\ w_{6,7} & w_{6,8} & w_{6,9} \end{matrix}$	Vienen de la capa 2
$\begin{matrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{matrix}$	$\begin{matrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{matrix}$	$\begin{matrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{matrix}$	Vienen de la capa 3
Llegan a la capa 1	Llegan a la capa 2	Llegan a la capa 3	

Figura 3.1: Perceptrón multicapa y matriz de pesos asociada. Puede verse que la topología de la red y la configuración de la matriz de pesos son biunívocas.

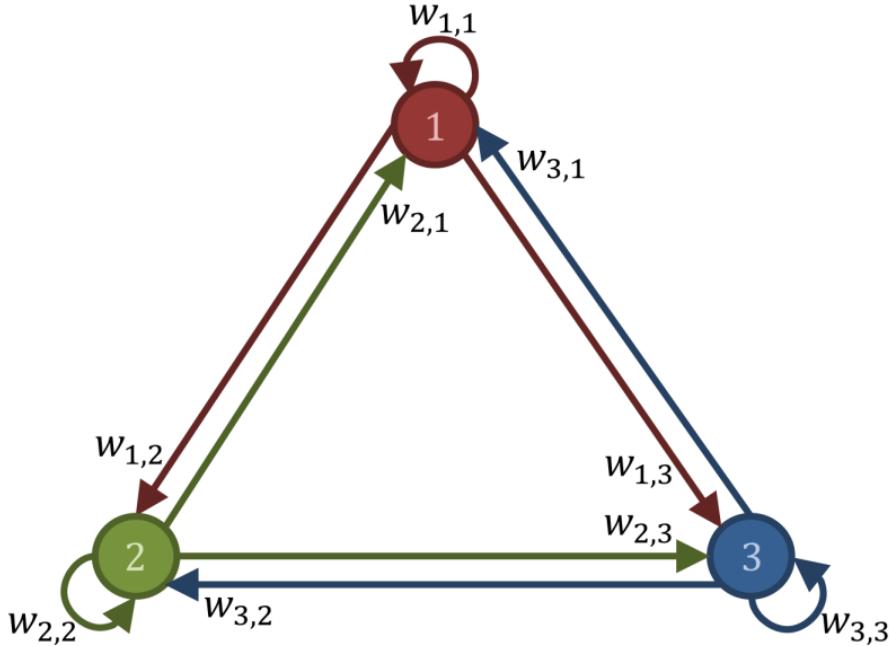


Figura 3.2: Red de Hopfield. Ahora, la matriz de pesos tiene todos sus valores permitidos.

- Su modelo incluye realimentaciones, que son básicas en su modo de funcionamiento.
- La elección de la arquitectura del perceptrón multicapa se realiza en forma arbitraria.
- El perceptrón multicapa funciona de manera síncrona, es decir, todas las neuronas cambian al mismo tiempo. La red de Hopfield permite un funcionamiento tanto síncrono como asíncrono, aunque el funcionamiento asíncrono es el más habitual en las neuronas biológicas.

El grafo de la red cambia con respecto al perceptrón multicapa, la representación no es la de un grafo separable por capas con conexiones hacia adelante, sino la de un grafo completo como se ve en la figura ??.

### 3.2.2. Un caso de estudio

La red neuronal usada tiene el modelo de tiempo contínuo

$$\dot{u} = -u + W \cdot f(u); \quad u \in \mathbb{R}^3 \quad (3.1)$$

en donde  $u$  es un vector de tres dimensiones,  $W$  es la matriz de pesos y  $f$  es la función de activación

$$u = \begin{pmatrix} x \\ y \\ z \end{pmatrix}; \quad W = \begin{pmatrix} w_{1,1} & w_{1,2} & w_{1,3} \\ w_{2,1} & w_{2,2} & w_{2,3} \\ w_{3,1} & w_{3,2} & w_{3,3} \end{pmatrix}; \quad f = \begin{pmatrix} \arctan x \\ \arctan y \\ \arctan z \end{pmatrix} \quad (3.2)$$

Ecuaciones que se corresponden con el diagrama de la figura ???. Vemos de la ecuación que se corresponde con una red de Hopfield de memoria diferencial. No disponemos de computadoras analógicas, por lo que el sistema debe ser convertido a tiempo discreto. Aunque el paquete de programas Matlab incluye rutinas para el cálculo de ecuaciones diferenciales, perdemos el control de paso de tiempo necesario para calcular el exponente de Lyapunov con el método descripto en la sección 2.1, además lo necesitamos para una futura implementación en hardware. Usamos para esto una aproximación de Euler de primer orden, en donde la derivada se aproxima con un trapecio de base  $\Delta t$ .

$$\begin{aligned} \frac{u_{n+1} - u_n}{\Delta t} \approx \dot{u}_n &= -u + W \cdot f(u_n) \Rightarrow \\ \Rightarrow u_{n+1} &= (1 - \Delta t)u_n + \Delta t W \cdot f(u_n) \\ &= Gu_n + \Omega f(u_n) \end{aligned} \quad (3.3)$$

En la figura ?? se muestra nuestra nueva red neuronal en tiempo discreto. Sus coeficientes dependen del paso de tiempo. Este sistema se approxima al de tiempo continuo en el límite  $\Delta t \rightarrow 0$ , en nuestro caso se verificó que el sistema converge al planteado. Pudo verse que con  $\Delta t = 1$  y  $\Delta t = 0,1$  las soluciones en el espacio de fases fueron las mismas, para hacer los cálculos utilizamos  $\Delta t = 0,01$ .

Se barrió un parámetro (peso de un axón) para identificar la existencia de caos en función de éste. Siguiendo a [?] en donde se reporta una transición al caos en torno a un juego de parámetros, utilizamos la siguiente matriz de pesos: en donde  $p$  es el parámetro a barrer entre  $-0,35$  y  $0,55$  en pasos de  $9 \times 10^{-5}$ .

Para cada valor del parámetro se le da condiciones iniciales al sistema  $[1, 68; -0, 292; -3, 47]$  y se lo deja evolucionar  $800s$ , esto es  $200s$  más que el transitorio más largo reportado en [?], con esto nos aseguramos de descartar el transitorio y que el sistema se encuentra en régimen permanente. Se calcula el MLE para  $t \in (800; 1000]$ .

De esta forma se genera la figura ?? en donde se muestra el MLE en función del parámetro. Como es usual, el MLE no es una función suave, sinó que es una función discontinua que presenta saltos abruptos en todo el dominio, sin embargo, se encontraron zonas de caos robusto fente al parámetro  $p$  en algunos intervalos, especialmente en  $p \in (0, 0223; 0, 0791)$ , esto significa que el caos persiste con una variación no infinitesimal del parámetro, esta zona es muy buscada para implementaciones prácticas.

Para mostrar la transición al caos y la relación entre el MLE y el espacio de fases, se eligieron dos parámetros  $p_1 = -0,2725$  y  $p_2 = 0,268$ , para  $p_1$  el  $MLE = -2,2 \times 10^{-3}$ , para  $p_2$  el  $MLE = 1,55 \times 10^{-2}$ . Se muestra la trayectoria resultante para cada uno en la figura ??.

Para el atractor caótico, dos trayectorias generadas a partir de condiciones iniciales muy cercanas deben, al cabo de un tiempo, separarse y oscilar en trayectorias distintas. En la figura ?? puede verse este efecto.

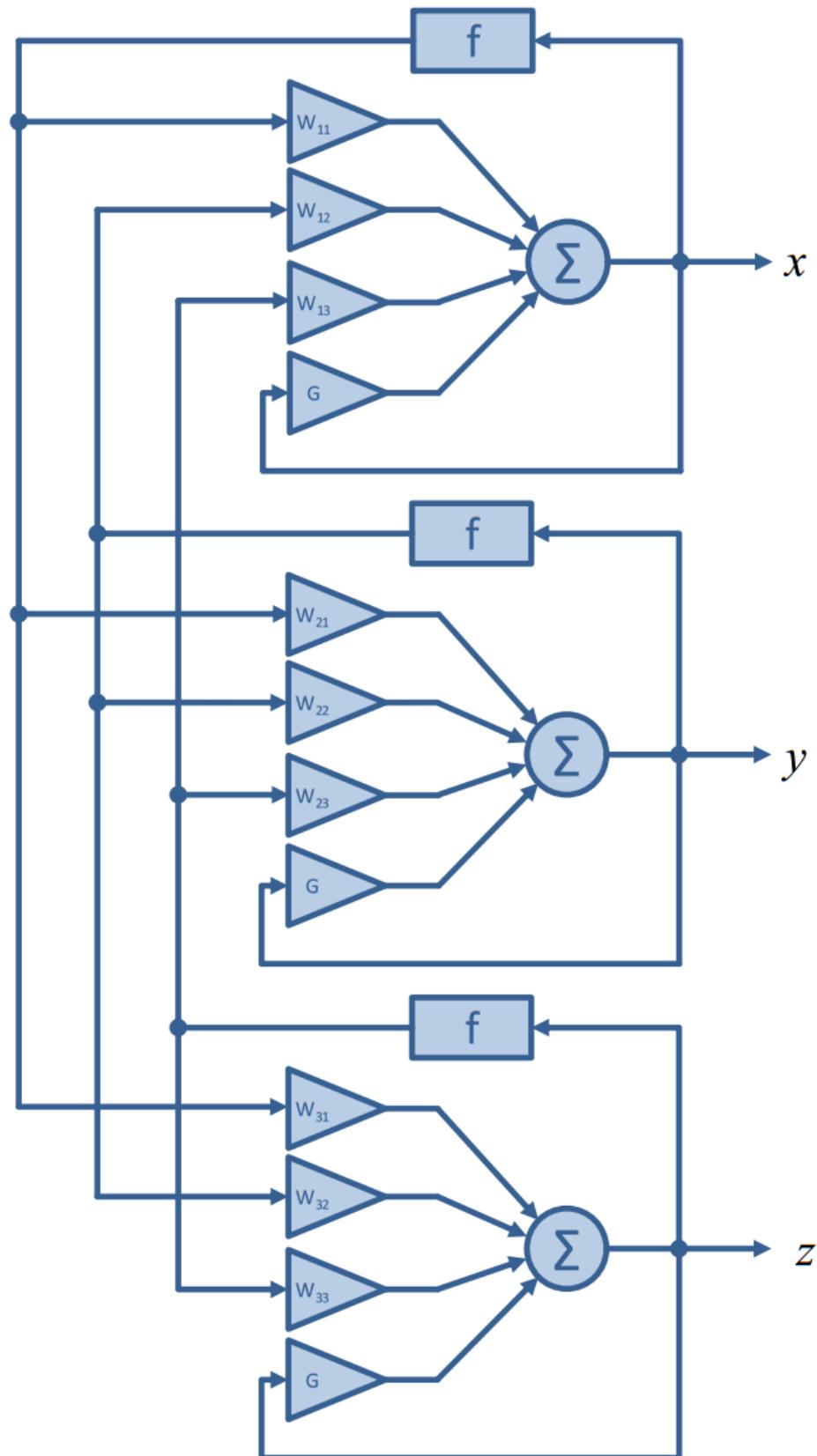


Figura 3.3: Red utilizada. Se trata de una red de Hopfield tridimensional de memoria diferencial, el diseño está orientado a una posterior implementación en FPGA.

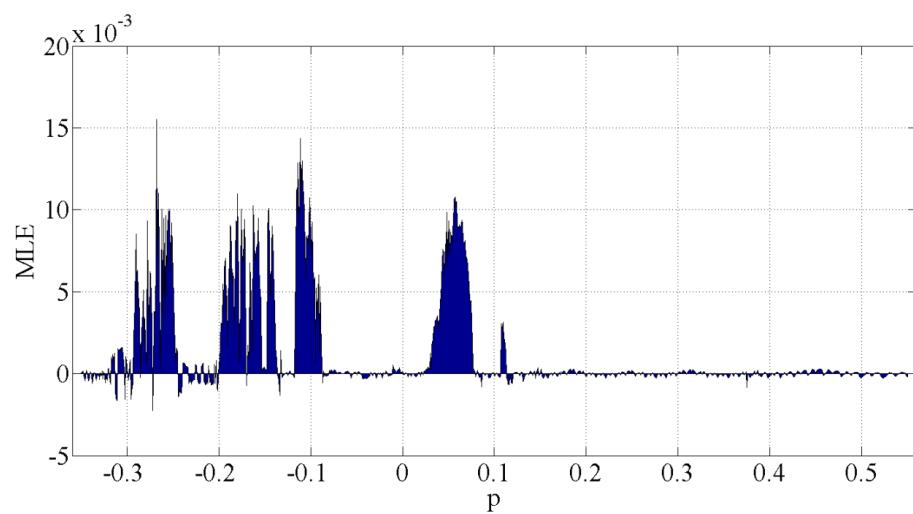


Figura 3.4: Exponente de Lyapunov en función del parámetro  $p$ . Existe caos en toda la zona en la que es positiva.

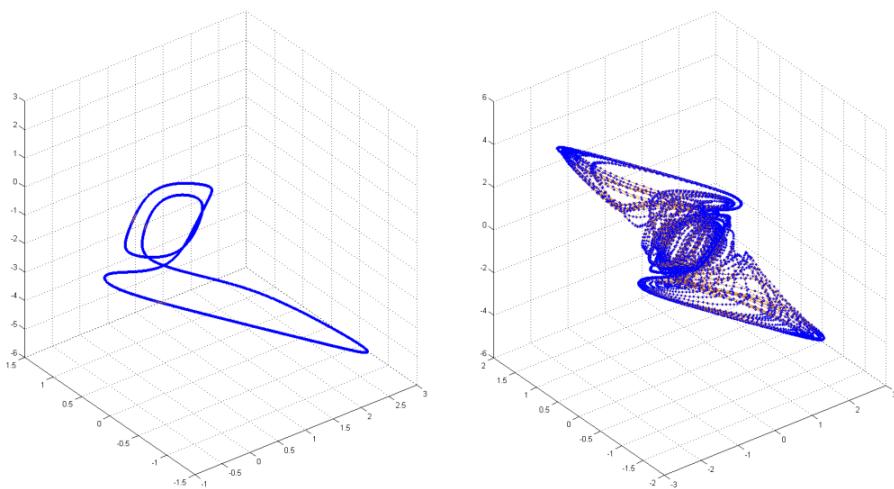


Figura 3.5: Dos trayectorias características del sistema en el espacio de fases. La trayectoria de la izquierda se corresponde con un  $MLE < 0$  y la positiva con un  $MLE > 0$ .

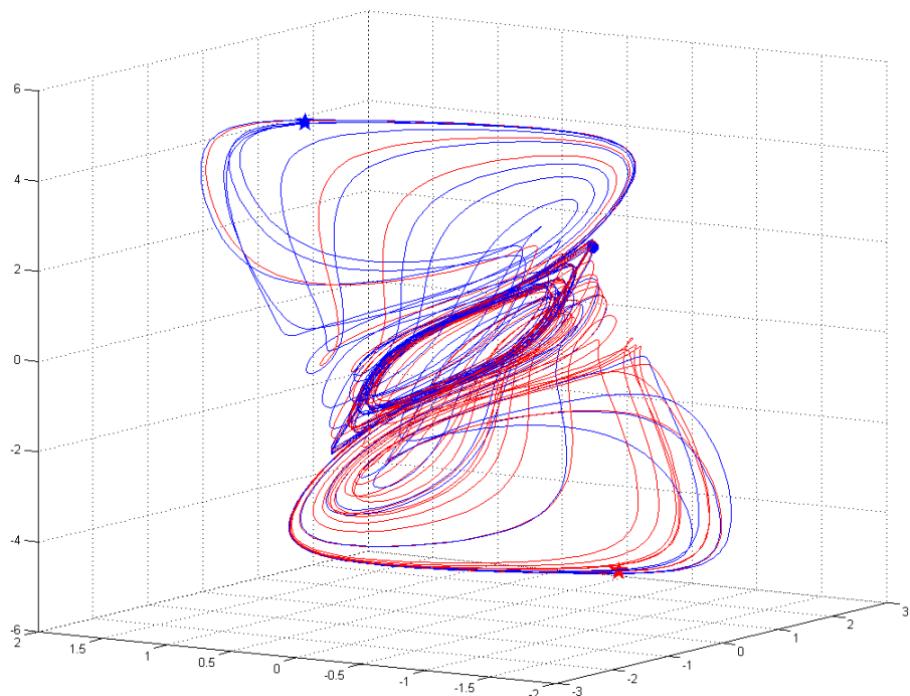


Figura 3.6: Dos trayectorias de la red de Hopfield para condiciones iniciales próximas. Las condiciones iniciales están marcadas con dos puntos grandes cerca del centro del atractor y los valores después de  $\Delta t = 3s$  con estrellas en ambos extremos de la figura.



## Capítulo 4

# El problema de la Aritmética Discreta

### 4.1. Analysis of the digital implementation of a chaotic deterministic-stochastic attractor (EAMTA 2012)

Otro que no tengo el latex, se lo tengo que pedir a Luciana

In this work the implementation, of chaos-based pseudo random number generators (PRNG), onto a Field Programmable Gate Array (FPGA), is analyzed. Any digital implementation requires the choice of an algorithm to discretize time and a representation standard to represent real numbers. Each choice modifies the stochasticity degree of the system and also defines a different amount of resources on the FPGA. The main contribution of this paper is to propose an optimum design methodology for applications in which the chaotic system is going to replace a stochastic system. This is the case with PRNG. In stochastic systems the randomness degree must be measured. In this paper we use the global indicator proposed by Marsaglia in his widely used DIEHARD test-suite. Results are exemplified for the Lorenz chaotic oscillator but the same methodology may be used with other low dimensional chaotic systems.

### 4.2. Complexity of switching chaotic maps

#### 4.2.1. Introduction

In the last years digital measuring systems become the standard in all experimental sciences. By using *virtual instruments* and new programmable electronic devices, such as Digital Signal Processors (*DSP*) and Field Programmable Gate Arrays (*FPGA*) experimenters may design and modify their own measuring systems.

The effect of finite precision in these new devices needs to be investigated. This issue is critical if chaotic systems must be implemented, because due to roundoff errors digital implementations will always become periodic with a period  $T$  and unstable orbits with a low period may become stable destroying

completely the chaotic behavior. Grebogi and coworkers [?] studied this subject and they shaw that the period  $T$  scales with roundoff  $\epsilon$  as  $T \sim \epsilon^{-d/2}$  where  $d$  is the correlation dimension of the chaotic attractor.

To have a large period  $T$  is one an important property of a chaotic map. Stochasticity and mixing are also relevant. Furthermore to characterize these properties several quantifiers were studied [?]. Among them the use of an entropy-complexity representation ( $H - C$  plane) deserves special consideration[?, ?, ?, ?, ?]. A fundamental issue is the criterium to select the distribution function (*PDF*) assigned to the time series. Causal and non causal options are possible. Here we consider the non-causal traditional *PDF* obtained by normalization of the histogram of the time series. Its statistical quantifier is the normalized entropy  $H_{hist}$  that is a measure of equiprobability among all allowed values. We also consider a causal *PDF* that is obtained by assigning ordering patterns to segments of trajectory of length  $D$ . This PDF were first proposed by Bandt & Pompe in a seminal paper [?]. The corresponding entropy  $H_{BP}$  was also proposed as a quantifier by Bandt & Pompe. Amigó and coworkers proposed the number of forbidden patterns as a quantifier of chaos [?]. Essentialy they reported the presence of forbidden patterns as an indicator of chaos. Recently it was shown that the name forbidden patterns is not convenient and it was replaced by *missing patterns*(MP) [?].

Switching systems naturally arise in power electronics and many other areas in digital electronics. They have also interest in transport problems in deterministic ratchets [?] and it is known that synchronization of the switching procedure affects the output of the controlled system. Nagaraj et al [?] studied the case of switching between two maps. They shaw that the period  $T$  of the compound map obtained by switching between two chaotic maps is higher than the period of each map. Liu et al [?] studied different switching rules applied to linear systems to generate chaos. Switching chaos was also addressed in [?]. Skipping values of the time series is another simple technique used to increase mixing in chaotic maps [?].

In this paper we study the statistical characteristic of two well known maps: the tent map (TENT) and logistic map (LOG). Three additional maps are generated: 1) SWITCH, generated by switching between TENT and LOG; 2) EVEN, generated by skipping all the elements in odd position in SWITCH time series and 3) ODD, generated by discarding all the elements in an even position in SWITCH time series. Floating point, decimal numbers and binary numbers are used. All these specific numerical systems may be implemented in modern programmable logic boards.

The main contributions of this paper are:

1. the definition of different statistical quantifiers and their relationship with the properties of the time series generated by the map.
2. the study of how this quantifiers are modified by the numerical representation using floating point, decimal and binary numbers. It is specially interesting to note that some systems (TENT) with very nice statistical properties in the world of the real numbers, become “pathological” when numerical representations are used.
3. the effect of switching between two different maps, on the period and the statistical properties of the time series. Floating point, decimal and binary

numerical representations are considered.

4. the effect of skipping values in any of these maps

Organization of the paper is as follows: section ?? describes the statistical quantifiers used in the paper and the relationship between their value and characteristics of the causal and non causal PDF considered; section 4.2.3 shows and discuss the results obtained for all the numerical representations. Finally section 4.2.5 deals with final remarks and future work.

### 4.2.2. Information theory quantifiers

The first step to quantify the statistical properties of the values (amplitude statistics) of a time series  $\{x_i, (i = 1, \dots, N)\}$ , using information theory is to determine the concomitant PDF because all the quantifiers are functionals of the PDF associated to the time series. This is an issue studied in detail in previous papers [?]. Let us summarize the procedure:

1. a finite alphabet with  $M$  symbols  $\mathbf{A} = \{a_1, \dots, a_M\}$  is chosen.
2. one of these symbols is assigned: (a) to each value of the time series or (b) to each portion of length  $D$  of the trajectory.
3. the normalized histogram of the symbols is the desired *PDF*.

Note that if option (a) is chosen in step 2 then the PDF is *non causal*, because all the information about the time evolution of the system generating  $\{x_i\}$  is completely lost. On the contrary if option (b) is chosen in step 2 then the PDF is *causal*, in the sense it has some information about the temporal evolution.

Of course there are infinite possibilities to choose the alphabet  $\mathbf{A}$  as well as the length  $D$ . Bandt & Pompe made a proposal for a causal PDF that has been shown to be easy to implement and useful in a great variety of applications. The procedure is the following [?, ?, ?]:

- Given a series  $\{x_t : t = 0, \Delta t, \dots, M\Delta t\}$ , a sequence of vectors of length  $d$  is generated.

$$(s) \mapsto (x_{t-(d-1)\Delta t}, x_{t-(d-2)\Delta t}, \dots, x_{t-\Delta t}, x_t) , \quad (4.1)$$

Each vector turns out to be the “history” of the value  $x_t$ . Clearly, the longer the length of the vectors  $D$ , the more information about the history would the vectors have but a higher value of  $N$  is required to have an adequate statistics.

- The permutations  $\pi = (r_0, r_1, \dots, r_{D-1})$  of  $(0, 1, \dots, D-1)$  are called “order of patterns” of time  $t$ , defined by:

$$x_{t-r_{D-1}\Delta t} \leq x_{t-r_{D-2}\Delta t} \leq \dots \leq x_{t-r_1\Delta t} \leq x_{t-r_0\Delta t}. \quad (4.2)$$

In order to obtain an unique result it is considered  $r_i < r_{i-1}$  if  $x_{t-r_i\Delta t} = x_{t-r_{i-1}\Delta t}$ .

In this way, all the  $D!$  possible permutations  $\pi$  of order  $D$ , and the PDF  $P = \{p(\pi)\}$  is defined as:

$$p(\pi) = \frac{\#\{s | s \leq M - D + 1; (s) \text{ has type } \pi\}}{M - D + 1}. \quad (4.3)$$

In the last expression the  $\#$  symbol means “number”.

This procedure has the advantages of being *i*) simple, *ii*) fast to calculate, *iii*) robust in presence of noise, and *iv*) invariant to lineal monotonous transformations.

It is applicable to weak stationarity processes (for  $k = D$ , the probability that  $x_t < x_{t+k}$  doesn't depend on the particulary  $t$  [?]). The causality property of the PDF allows the quantifiers (based on this PDFs) to discriminate between deterministic and stochastic systems [?].

According to this point Bandt and Pompe suggested  $3 \leq D \leq 7$ .  $D = 6$  has been adopted in this work.

Based on our previous research [?, ?] we have employed two *PDF*'s: (a) the normalized histogram of the time series amplitudes  $\{x_i\}$  (that is a non-causal *PDF*), and (b) the Bandt & Pompe *PDF* (that is a causal *PDF*). The entropies  $H_{hist}$  and  $H_{BP}$ , the statistical complexity  $C_{BP}$  are used as quantifiers.

We also used the number of missing patterns  $MP$  as a quantifier[?]. As shown recently by Amigó *et al.* [?, ?, ?, ?], in the case of deterministic one-dimensional maps, not all the possible ordinal patterns can be effectively materialized into orbits, which in a sense makes these patterns “forbidden”. Indeed, the existence of these *forbidden ordinal patterns* becomes a persistent fact that can be regarded as a “new” dynamical property. Thus, for a fixed pattern-length (embedding dimension  $D$ ) the number of forbidden patterns of a time series (unobserved patterns) is independent of the series length  $N$ . Remark that this independence does not characterize other properties of the series such as proximity and correlation, which die out with time [?, ?].

A full discussion about the convenience of using these quantifiers is out of the scope of this work. Nevertheless reliable bibliographic sources do exist [?, ?, ?, ?, ?, ?, ?, ?].

The entropies  $H_{hist}$  and  $H_{BP}$  are the normalized version of the Of course there are infinite possibilities to choose the alphabet as well as the length  $d$ . Bandt & Pompe made a proposal for a causal PDF that has been shown to be easy to implement and useful in a great variety of applications. The procedure is the following [?, ?, ?]: a) Given a series  $\{x_t : t = 0, \Delta t, \dots, M\Delta t\}$ , a sequence of vectors of length  $d$  is generated.

$$(s) \mapsto (x_{t-(d-1)\Delta t}, x_{t-(d-2)\Delta t}, \dots, x_{t-\Delta t}, x_t), \quad (4.4)$$

Each vector turns out to be the “history” of the value  $x_t$ . Clearly, the longer the length of the vectors  $d$ , the more information about the history would the vectors have. b) The permutations  $\pi = (r_0, r_1, \dots, r_{d-1})$  of  $(0, 1, \dots, d-1)$  are called “order of patterns” of time  $t$ , defined by:

$$x_{t-r_{d-1}\Delta t} \leq x_{t-r_{d-2}\Delta t} \leq \dots \leq x_{t-r_1\Delta t} \leq x_{t-r_0\Delta t}. \quad (4.5)$$

In order to obtain an unique result it is considered  $r_i < r_{i-1}$  if  $x_{t-r_i\Delta t} = x_{t-r_{i-1}\Delta t}$ .

In this way, all the  $d!$  possible permutations  $\pi$  of order  $d$ , and the PDF  $P = \{p(\pi)\}$  is defined as:

$$p(\pi) = \frac{\#\{s | s \leq M - Dd + 1; (s) \text{ has type } \pi\}}{M - d + 1}. \quad (4.6)$$

In the last expression the  $\#$  symbol means “number”.

This procedure has the advantages of being *i*) simple, *ii*) fast to calculate, *iii*) robust in presence of noise, and *iv*) invariant to lineal monotonous transformations.

It is applicable to weak stationarity processes (for  $k = d$ , the probability that  $x_t < x_{t+k}$  doesn't depend on the particularity  $t$  [?]). The causality property of the PDF allows the quantifiers (based on this PDFs) to discriminate between deterministic and stochastic systems [?].

The choice of the embedding dimension  $d$  is crucial because it determines the minimal length acceptable of the original temporal series ( $M \gg d!$ ) needed to obtain an adequate statistics. According to this point Bandt and Pompe suggested  $3 \leq d \leq 7$ .  $d = 6$  has been adopted in this work.

Based on our previous research [?] we have employed the statistical complexity  $C$  and the entropy  $H$  to define a plane where the stochasticity of the chaotic system may be represented. A full discussion about the convenience of using these quantifiers is out of the scope of this work. Nevertheless reliable bibliographic sources do exist [?, ?, ?, ?, ?, ?].

The entropy  $H[P]$  is the normalized version of the Entropy proposed by Shannon [?]:

$$H[P] = S[P]/S_{max}, \quad (4.7)$$

where  $S[P] = -\sum_{j=1}^M p_j \ln(p_j)$   
and  $S_{max}$  is the normalizing constant:

$$S_{max} = S[P_e] = \ln M, \quad (4.8)$$

and  $P_e = \{1/M, \dots, 1/M\}$  is the uniform distribution. The number of symbols  $M$  is equal to  $N$  for  $H_{hist}$  and it is equal to  $D!$  for  $H_{BP}$ .

The statistical complexity  $C[P]$  is given by:

$$C[P] = Q_J[P, P_e] \cdot H[P], \quad (4.9)$$

, and  $Q_J$  is named “disequilibrium” and it is the distance between  $P$  and  $P_e$  in the probability space. The metric used in this paper is based on the Jensen-Shannon divergence [?]:

$$Q_J[P, P_e] = Q_0 \cdot \left\{ S\left[\frac{P + P_e}{2}\right] - S[P]/2 - S[P_e]/2 \right\}. \quad (4.10)$$

The normalization constant  $Q_0$  is:

$$Q_0 = -2 \left\{ \left( \frac{N+1}{N} \right) \ln(N+1) - 2 \ln(2N) + \ln N \right\}^{-1}. \quad (4.11)$$

From the statistical point of view the disequilibrium  $Q_J$  is an intensive magnitude, and it is 0 if and only if  $P = P_e$ . It has been proved that the  $C[P]$  quantifies the presence of nonlinear correlations typical of chaotic systems [?, ?].

The complexity  $C[P]$  is independent from the entropy  $H[P]$ , as far as different  $P$ 's share the same entropy  $H[P]$  but they have different complexity  $C[P]$ .

Two representation planes are considered:  $H_{BP}$  vs  $H_{hist}$  [?] and  $H_{BP}$  vs  $C_{BP}$  [?]. In the first plane a higher value in any of the entropies,  $H_{BP}$  and  $H_{hist}$ , implies an increase in the uniformity of the involved *PDF*. The point  $(1, 1)$  represents the ideal case with uniform histogram and uniform distribution of ordering patterns. In the second plane not the entire region  $0 < H_{BP} < 1$ ,  $0 < C_{BP} < 1$  is achievable. In fact for any *PDF* the pairs  $(H, C)$  of possible values fall between two extreme curves in the plane  $H-C$  [?]. Fig. ?? shows two regions labeled as *deterministic* and *stochastic*. In fact transition from one region to the other are smooth and the division is a bit arbitrary. A more detailed discussion can be seen in [?]. Ideal random systems having uniform Bandt & Pompe *PDF*, are represented by the point  $(1, 0)$  [?] and a delta-like *PDF* corresponds with the point  $(0, 0)$ .

#### 4.2.3. Results

Five pseudo chaotic maps were studied. For each one a floating point representation, a decimal numbers representation with  $1 \leq P \leq 27$  and a binary numbers representation with  $1 \leq B \leq 27$  are considered. For each representation 1000 time series were generated using randomly chosen initial conditions within the interval  $[0, 1]$ . The studied maps are tent (TENT), logistic (LOG) a sequential switching between TENT and LOG (SWITCH). Furthermore a skipping randomization procedure is applied to SWITCH [?], discarding the values in the odd positions (EVEN) or the values in the even positions (ODD) respectively. Let us detail our results for each of these maps.

#### 4.2.4. Simple maps.

Here we report our results for both maps:

##### 1. Tent map (TENT)

$$x_{n+1} = \begin{cases} 2x_n & \text{if } 0 \leq x_n \leq 1/2 \\ 2(1-x_n) & \text{if } 1/2 < x_n \leq 1 \end{cases}, \quad (4.12)$$

with  $x_n \in \mathcal{R}$ . The Tent map has been extensively studied in the literature because theoretically it has nice statistical properties that can be analytically obtained. For example it is easy to proof that it has a uniform histogram and consequently an ideal  $H_{hist} = 1$ . The Perron-Frobenius operator and its corresponding eigenvalues and eigenfunctions may be also be analytically obtained for this map [?].

When this map is implemented in a computer using any numerical representation system (even floating point!) truncation errors rapidly increases and makes the unstable fixed point in  $x^* = 0$  becomes stable producing a short transitory followed by an infinite number of 0's[?, ?]. Some authors [?] have proposed to add a random perturbation to avoid this drawback of the Tent map. But this procedure introduces statistical properties of the random perturbation that are mixed with those of the Tent map itself.

Here we study the Tent map “as it is” without any artifact to evaluate its real instead of theoretical statistical properties. Note that to effectively

work in a given representation it is necessary to change the expression of the map in order to make all the operations in the chosen representation numbers. For example, in the case of TENT the expression in decimal numbers is:

$$x_{n+1} = \begin{cases} 2x_n & \text{if } 0 \leq x_n \leq 1/2 \\ \epsilon \times \text{floor}\left\{\frac{2 - 2x_n}{\epsilon}\right\} & \text{if } 1/2 < x_n \leq 1 \end{cases}, \quad (4.13)$$

with  $\epsilon = 10^{-P}$  for decimal numbers and  $\epsilon = 2^{-B}$  for binary numbers. In Eq. 4.13  $x_n$  is either a decimal number with  $P$  digits or a binary number with  $B$  bits.

Figs. 4.3 (a) to (e) show the different quantifiers for floating point and decimal numerical representation. In each figure from (a) to (c) a dashed line shows the value for the floating point representation. In figures (d) and (e) the star corresponds to the floating point case. In decimal representations the value of  $H_{hist}$  remains almost constant for  $11 \leq P \leq 16$  (see Fig. 4.3 (a)). Its value is  $\langle H_{hist} \rangle = 0,8740$  with a variance  $\sigma_{H_{hist}} = 2,5 \times 10^6$ . For lower or higher values pf  $P$  entropy decreases. This effect is due precisely to the stabilization of the fixed point at  $x = 0$ . For ordering patterns entropy  $H_{BP}$  an almost constant value is obtained for  $8 \leq P \leq 15$ . The value is  $H_{BP} \approx 0,6287$  with variance  $\sigma_{H_{BP}} = 4,8 \times 10^{-6}$  (see Fig. 4.3 (b)). This rather small maximum value may be understood by seeing Fig. 4.3 (c), where the number of MP. is minimal for  $P$  within the same range but it is still large: 645 patterns are missing and only 75 ordering patterns are present in the time series. Then, even with a uniform distribution between these 75 patterns, entropy can not be higher than  $\ln(75)/\ln(720) \approx 0,65$ . A more complete perspective of the statistical properties is obtained in Fig. 4.3 (d) showing the representative point in the  $H_{hist}, H_{BP}$  plane for different precisions. Note that the best choice for maximum stochasticity is obtained for  $11 \leq P \leq 15$ , with maximum attainable values for both entropies. Increasing the number of decimal figures makes Tent map worst in the sense the system approaches the state for the floating point representation (the star at  $(0,0)$ ). Statistical complexity  $C_{BP}$  is also maximal for  $8 \leq P \leq 15$ . Fig. 4.3 (e) shows the representation on the  $H_{BP}, C_{BP}$  plane. In this plane it is also clear that the more stochastic option corresponds with  $11 \leq P \leq 15$  but even in the optimum case the representative point is located in a position very similar to other chaotic maps, very far from the ideal point for stochastic systems in this plane that is  $(1,0)$  [?]. Binary numerical representation of the Tent map remains very near to the floating point values for  $1 \leq B \leq 27$  (see Fig. 4.3 (f)). The conclusion is it is convenient to use a decimal numbers representation with  $P = 11$  to get the optimum time series for the Tent map. A higher number of decimal figures does not improve the statistical properties of the time series. Furthermore binary and floating point representations are not allowed.

2. Logistic map (LOG) Logistic map is representative of the very large family of quadratic maps.

$$x_{n+1} = 4x_n(1-x_n), \quad (4.14)$$

with  $x_n \in \mathcal{R}$ . Figs. 4.4 (a) to (f) show the statistical properties of LOG map in floating point and decimal numbers representation. This map does

not show the anomalies pointed above for the tent map. For  $P \geq 10$  the values of  $H_{hist}$ ,  $H_{BP}$  and  $C_{BP}$  remains almost identical to the values for the floating point representation. Their values are:  $\langle H_{hist} \rangle = 0,8621$  with variance  $\sigma_{H_{hist}} = 0,062 \times 10^{-6}$ ;  $\langle H_{BP} \rangle = 0,6292$  with variance  $\sigma_{H_{BP}} = 0,060 \times 10^{-6}$ ;  $\langle C_{BP} \rangle = 0,4842$  with variance  $\sigma_{C_{BP}} = 0,0195 \times 10^{-6}$ . Missing patterns stabilize in 645 for  $P \geq 8$  making  $H_{BP}$  to rise to its floating point value  $\langle H_{BP} \rangle = 0,629$  with variance  $\sigma_{H_{BP}} = 0,060 \times 10^{-8}$ . Note again that the stable value of mission patters missing patterns 645 makes the optimum  $H_{BP} \leq \ln(75)/\ln(720) \simeq 0,65$ . Then  $P = 10$  is the most convenient choice because an increase in the number of decimal figures does not improve the statistical properties. Figs. ?? show the corresponding figures for binary representations. The histogram entropy  $H_{hist}$  does not reach its floating point value within the maximum number of bits used. In the case of missing patterns the stable number 645 is obtained with  $B \geq 25$ . It means that using  $B = 25$  one obtains a time series with good statistical properties regarding the missing patterns, but distribution among the allowed binary values is not as uniform as can be obtained with a higher value of  $B$ .

In summary, a comparison between LOG and TENT maps shows that, in the case of decimal representation, the best choice for TENT ( $P = 11$ ) produces a higher value for  $H_{hist}$  than the best choice for LOG ( $P = 10$ ). Ordering patterns and the statistical properties related to them, are almost identical for the optimum choices in both maps. In the case of binary numbers only LOG can be used because TENT is highly anomalous.

### Sequential switching

1. Sequential switching between Tent and Logistic maps (SWITCH) SWITCH may be expressed as a composition between  $M_1 \circ M_2$  given by the following recurrence:

$$\left\{ \begin{array}{l} x_{n+2} = 4 x_{n+1} (1 - n + 1) \\ x_{n+1} = \begin{cases} 2 x_n & \text{if } 0 \leq x_n \leq 1/2 \\ 2 (1 - x_n) & \text{if } 1/2 < x_n \leq 1 \end{cases} \end{array} \right.$$

with  $x_n \in \mathcal{R}$ . Results with sequential switching are shown in Figs. 4.6 (a) to (f) for decimal numbers. The floating point entropy value is  $H_{hist} = 0,8658$ , a value very similar to the one obtained for the TENT map and higher to that obtained for LOG. For decimal numbers this value is reached for  $12 \leq P \leq 27$ . It means it is enough to use 12 decimal figures to get the same distribution of values in the time series. Regarding ordering patterns the number of MP decreases to 586, a value lower lower than the one obtained for any of two simple maps TENT and LOG. It means the entropy  $H_{BP}$  may increase up to  $\ln(134)/\ln(720) \simeq 0,74$  With decimal numbers the entropy  $H_{BP}$ stabilizes at  $P = 9$  with  $\langle H_{BP} \rangle \simeq 0,657$  and variance  $\sigma_{H_{BP}} \simeq 0,13 \times 10^{-7}$ . Note that the entropies  $H_{hist}$  and  $H_{BP}$  are not monotonously increasing with  $P$ . Considering all the quantifiers  $P = 12$  is the minimum number of decimal figures and statistical characteristics of this combined map are better than those for each individual map. Results with sequential switching in binary numbers are shown in Figs. 4.7. Results

for a number of bits  $B \simeq 27$  are equivalent to those obtained for  $P \simeq 9$  for decimal numbers. It means both representation are valid and equivalent in the sense they will require similar hardware resources.

2. Skipping is a usual randomizing technique that increases the mixing quality of a single map and correspondingly increases the value of  $H_{BP}$  and decreases  $C_{BP}$  of the time series. Skipping does not change the values of  $H_{hist}$  and  $C_{hist}$  evaluated using the non causal PDF (normalized histogram)[?]. In the case under consideration we study Even and Odd skipping of the sequential switching of Tent and Logistic maps.
  - a) Even skipping of the sequential switching of Tent and Logistic maps (EVEN).  
If  $\{x_n, (n = 1, \dots, \infty)\}$  is the time series generated by 1 discard all the values in odd positions and retain the values in even positions.
  - b) Odd skipping of the sequential switching of Tent and Logistica maps.  
If  $\{x_n, (n = 1, \dots, \infty)\}$  is the time series generated by 1 discard all the values in even positions and retain all the values in odd positions.  
The reason for studying even and odd skipping cases is the sequential switching map  $M_{switch}$  is the composition of two different maps. Even skipping may be expressed as  $M_{TENT} \circ M_{LOG}$  while odd skipping may be expressed as  $M_{LOG} \circ M_{TENT}$ .

This is very interesting to note that a great improvement is obtained using any of the skipping strategies but EVEN is slightly better than ODD.

MP are reduced to  $MP \simeq 163$  for EVEN and  $MP \simeq 164$  for ODD, increasing the maximum allowed Bandt & Pompe entropy that reaches the mean value  $\langle H_{BP} \rangle \simeq 0,905$  with variance  $\sigma_{H_{BP}} \simeq 0,107 \times 10^{-6}$  for EVEN, and  $\langle H_{BP} \rangle \simeq 0,854$  with variance  $\sigma_{H_{BP}} \simeq 0,285 \times 10^{-6}$  for a decimal representation with  $9 \leq P \leq 27$ . The complexity is reduced to  $\langle C_{BP} \rangle \simeq 0,224$  with  $\sigma_{C_{BP}} \simeq 0,166 \times 10^{-6}$  for EVEN and  $\langle C_{BP} \rangle \simeq 0,282$  with  $\sigma_{C_{BP}} \simeq 0,281 \times 10^{-6}$  for ODD.

Quantifiers related to the normalized histogram slightly degrades with the skipping procedure. For example  $H_{hist}$  reduces from 0,866 without skipping to 0,813 for any EVEN or ODD.

Results in binary numbers are similar to those obtained for the equivalent number of figures in decimal numbers. For example the minimum in MP is reached for  $B = 27$ , and this number of bits is almost equivalent to  $P = 9$ .

In Figs. 4.8 and Figs. 4.9 are shown the results for EVEN. We do not give the Figs. for ODD because they are very similar, as pointed above.

### Period $T$ as a function of $P$ and $B$

The issue of how the period  $T$  is related with the representation with  $P$  decimal digits was studied by Grebogi and coworkers [?]. There they show that the period  $T$  scales with roundoff  $\epsilon$  as  $T \sim \epsilon^{-d/2}$  where  $d$  is the correlation dimension of the chaotic attractor. Nagaraj et al [?] studied the case of switching between two maps. They show that the period  $T$  of the compound map obtained

Cuadro 4.1: Period  $T$  as a function of  $P$  for the maps considered

map	m	b
TENT	0.436	-0.0705
LOG	0.422	0.0141
SWITCH	0.438	0.0276
EVEN	0.438	- 0.2734
ODD	0.438	- 0.2734

Cuadro 4.2: Period  $T$  as a function of  $B$  for the maps considered

map	m	b
TENT	-	-
LOG	0.494	-1.219
SWITCH	0.494	-0.871
EVEN	0.494	-1.871
ODD	0.494	-1.871

by switching between two chaotic maps is higher than the period of each map and they found that a "random" switching improves the results. Here we considered sequential switching to avoid the use of another random variable, because it can include its own statistical properties in the time series. We studied decimal and binary numbers representations. Fig. ?? shows  $T$  vs  $P$  in semi logarithmic scale. A straight line can fit the points and has the expression  $\log_{10}T = m \times P + b$  for decimal numbers and  $\log_2 T = m \times B + b$  for binary numbers, where  $m$  is the slope and  $b$  is the  $y$ -intercept. Results for all considered maps are summarized in Table 4.1 and 4.2.

Results are compatible for those obtained in [?]. Switching between maps increase de period  $T$  but the skipping procedure decrease it esentially to one half.

#### 4.2.5. Conclusions

In summary:

- 
- 
- 

produces a non-monotonous evolution toward the floating point result. This result is relevant because it shows that increasing the precision is not always recommended.

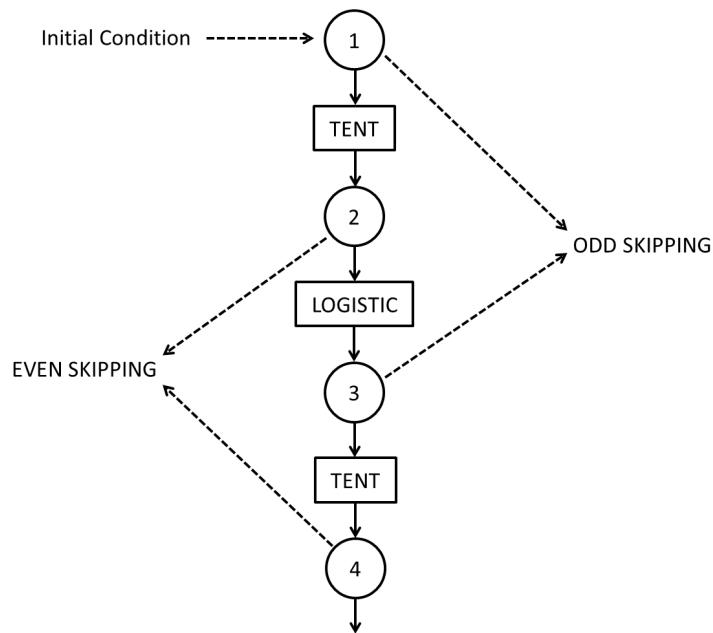


Figura 4.1: ZONA CH REHACER

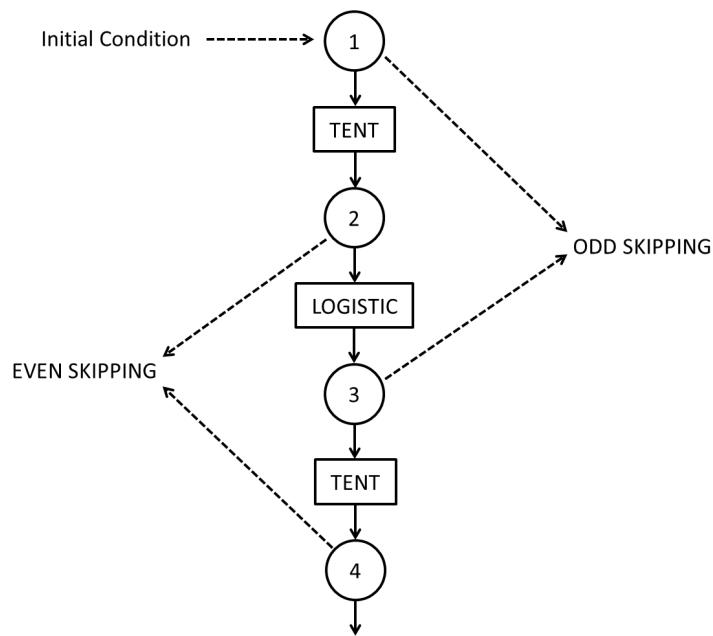


Figura 4.2: Sequential switching between Tent and Logistic maps. In the figure are also shown even and odd skipping strategies

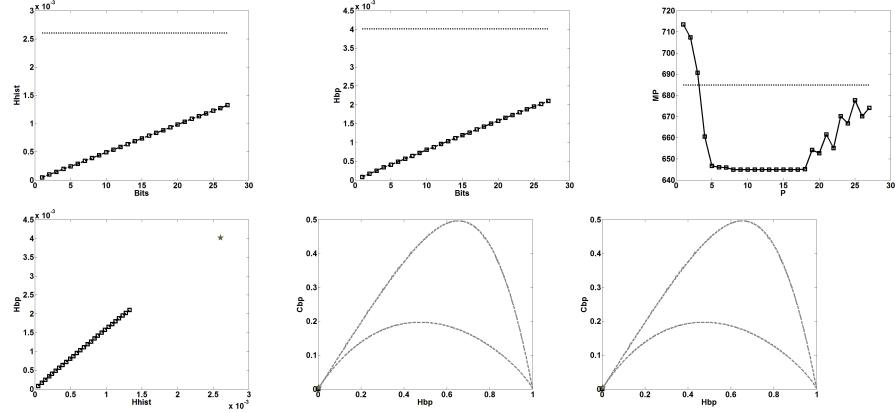


Figura 4.3: Statistical properties of the Tent map using different numerical representations. Figures (a) to (e) correspond to decimal representation: (a)  $H_{hist}$  vs  $P$  (b)  $H_{BP}$  vs  $P$  (c) Number of missing ordering patterns  $MP$  vs  $P$ . In Figures (a) to (c) dashed line correspond to floating point numbers. (d) representation in the  $H_{hist}, H_{BP}$  plane in the the decimal numerical system. The star represents the state for floating points numbers. (e) representation in the  $H_{BP}, C_{BP}$  plane. The star represents the state for floating points numbers. (f) representation in the  $H_{BP}, C_{BP}$  plane for binary numerical system. The star represents the state for floating points numbers.

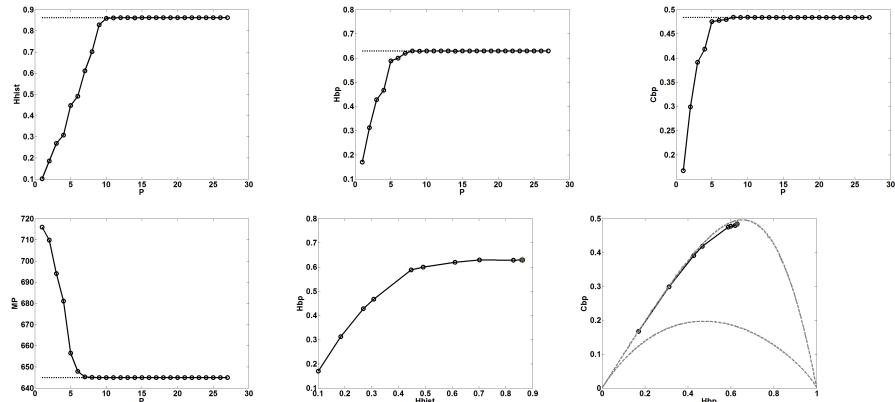


Figura 4.4: Statistical properties of the LOG map using different numerical representations. Figures (a) to (f) correspond to decimal representation: (a)  $H_{hist}$  vs  $P$  (b)  $H_{BP}$  vs  $P$  (c)  $C_{BP}$  vs  $P$  (d) Number of missing ordering patterns  $MP$  vs  $P$ . In Figures (a) to (d) dashed line correspond to floating point numbers. (d) representation in the  $H_{hist}, H_{BP}$  plane in the the decimal numerical system. The star represents the state for floating points numbers. (e) representation in the  $H_{hist}, H_{BP}$  plane. The star represents the state for floating point numbers; (f) representation in the  $H_{BP}, C_{BP}$  plane. The star represents the state for floating points numbers. (f) representation in the  $H_{BP}, C_{BP}$  plane for binary numerical system. The star represents the state for floating points numbers.

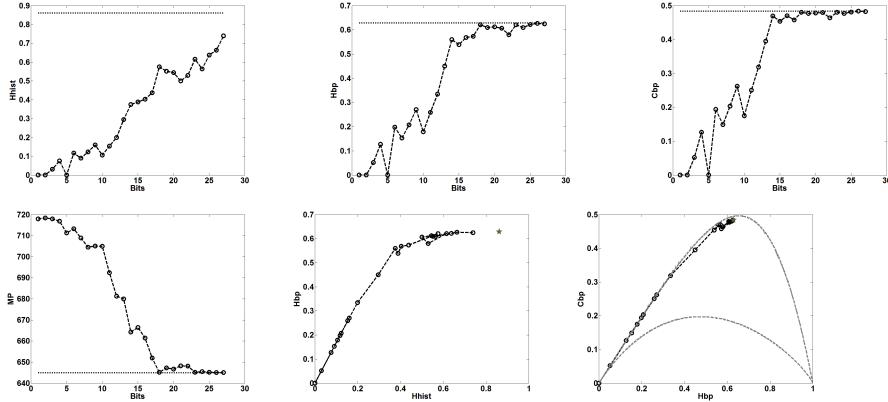


Figura 4.5: Statistical properties of the LOG map using binary representation: (a)  $H_{hist}$  vs  $P$  (b)  $H_{BP}$  vs  $P$  (c)  $C_{BP}$  vs  $P$  (d) Number of missing ordering patterns  $MP$  vs  $P$ . In Figures (a) to (d) dashed line correspond to floating point numbers. (d) representation in the  $H_{hist}, H_{BP}$  plane in the the decimal numerical system. The star represents the state for floating points numbers. (e) representation in the  $H_{hist}, H_{BP}$  plane. The star represents the state for floating point numbers; (f) representation in the  $H_{BP}, C_{BP}$  plane. The star represents the state for floating points numbers. (f) representation in the  $H_{BP}, C_{BP}$  plane for binary numerical system. The star represents the state for floating points numbers.

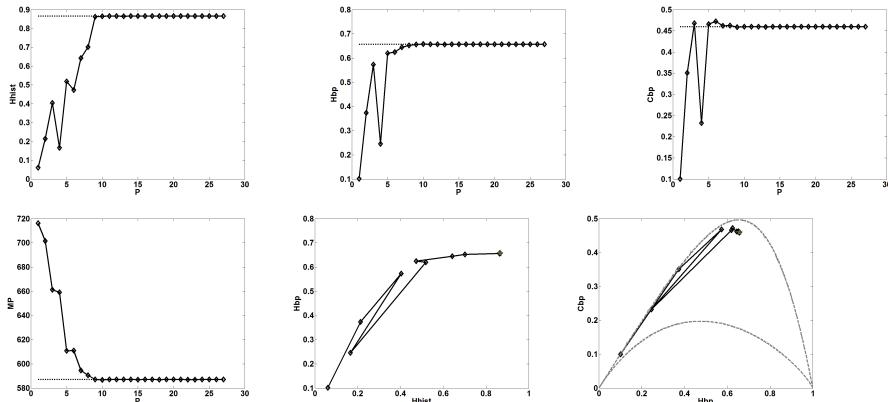


Figura 4.6: Statistical properties of the SWITCH map using decimal representation: (a)  $H_{hist}$  vs  $P$  (b)  $H_{BP}$  vs  $P$  (c)  $C_{BP}$  vs  $P$  (d) Number of missing ordering patterns  $MP$  vs  $P$ . In Figures (a) to (d) dashed line correspond to floating point numbers. (e) representation in the  $H_{hist}, H_{BP}$  plane in the the decimal numerical system. The star represents the state for floating points numbers. (f) representation in the  $H_{BP}, C_{BP}$  plane. The star represents the state for floating points numbers. (The star represents the state for floating points numbers).

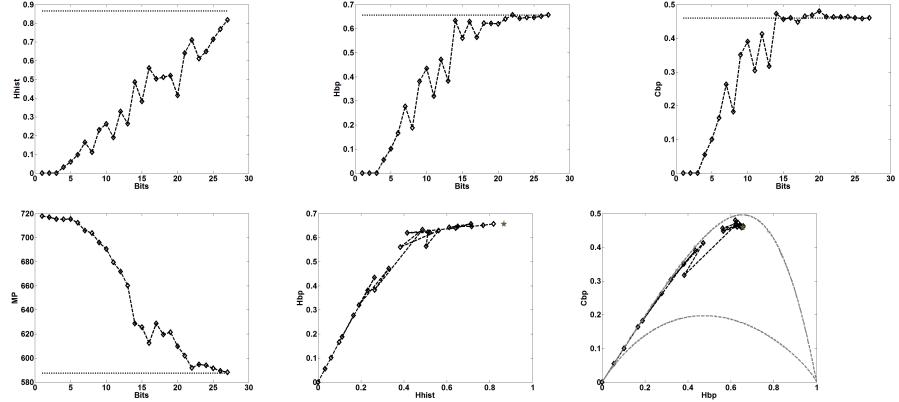


Figura 4.7: Statistical properties of the SWITCH map using binary representation: (a)  $H_{hist}$  vs  $P$  (b)  $H_{BP}$  vs  $P$  (c)  $C_{BP}$  vs  $P$  (d) Number of missing ordering patterns  $MP$  vs  $P$ . In Figures (a) to (d) dashed line correspond to floating point numbers. (e) representation in the  $H_{hist}, H_{BP}$  plane in the the binary numerical system. The star represents the state for floating points numbers. (f) representation in the  $H_{BP}, C_{BP}$  plane. The star represents the state for floating points numbers. (The star represents the state for floating points numbers).

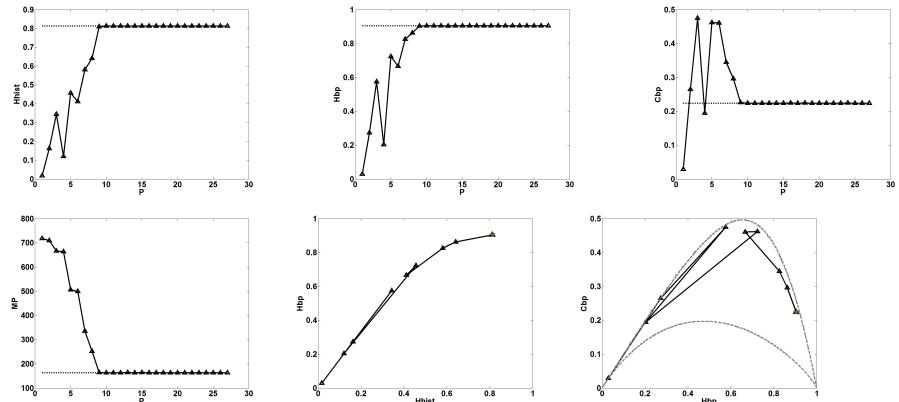


Figura 4.8: Statistical properties of EVEN, obtained by skipping the values in the odd position of the time series of SWITCH, using decimal representation: (a)  $H_{hist}$  vs  $P$  (b)  $H_{BP}$  vs  $P$  (c)  $C_{BP}$  vs  $P$  (d) Number of missing ordering patterns  $MP$  vs  $P$ . In Figures (a) to (d) dashed line correspond to floating point numbers. (e) representation in the  $H_{hist}, H_{BP}$  plane in the the decimal numerical system. The star represents the state for floating points numbers. (f) representation in the  $H_{BP}, C_{BP}$  plane. The star represents the state for floating points numbers.

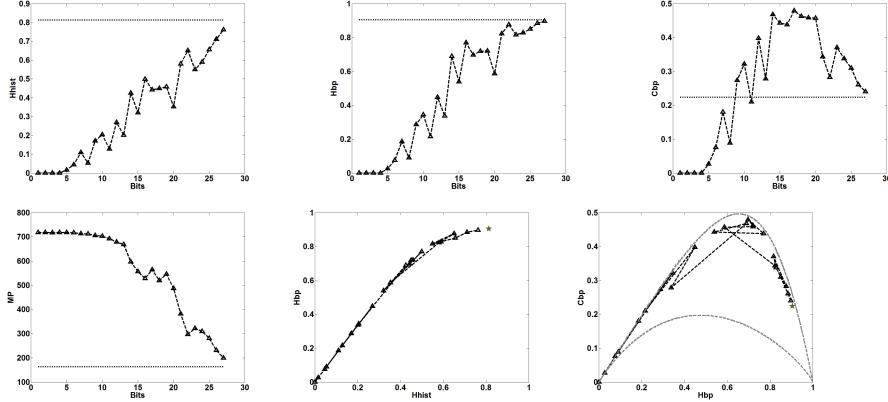


Figura 4.9: Statistical properties of EVEN, obtained by skipping the values in the odd position of the time series of SWITCH, using binary representation: (a)  $H_{hist}$  vs  $P$  (b)  $H_{BP}$  vs  $P$  (c)  $C_{BP}$  vs  $P$  (d) Number of missing ordering patterns  $MP$  vs  $P$ . In Figures (a) to (d) dashed line correspond to floating point numbers. (e) representation in the  $H_{hist}, H_{BP}$  plane in the the binary numerical system. The star represents the state for floating points numbers. (f) representation in the  $H_{BP}, C_{BP}$  plane. The star represents the state for floating points numbers.

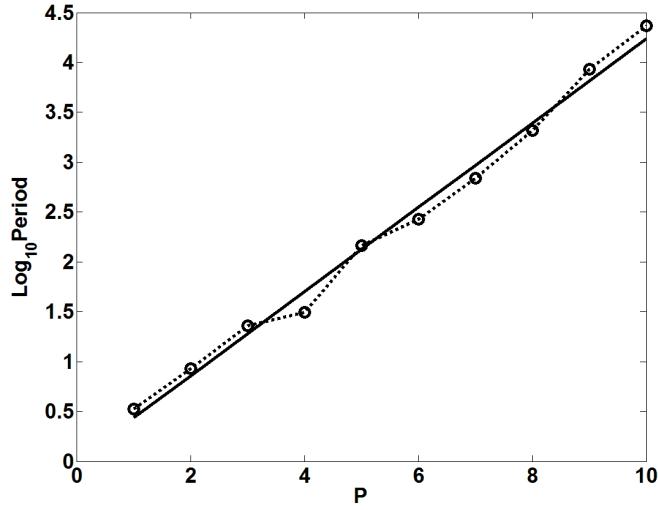


Figura 4.10: Period  $T$  as a function of de number of decimal digits  $P$  for the LOG map.



# **Capítulo 5**

## **Conclusiones**



## **Apéndice A**

# **Field Programmable Gate Array (FPGA)**

Cosas que distraen en la tesis.



# Bibliografía