

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>  |
| <b>2. Cuantificadores de Aleatoriedad</b>  | <b>3</b>  |
| 2.1. Máximo Exponente de Lyapunov . . . . .  | 3         |
| 2.2. Algoritmo evolutivo para la búsqueda de caos . . . . .  | 4         |
| 2.2.1. Estado actual del avance . . . . .  | 8         |
| 2.3. Cuantificadores de la teoría de la información . . . . .  | 12        |
| 2.3.1. Entropía de Shannon y Complejidad Estadística . . . . .   | 13        |
| 2.3.2. Determinación de la distribución de probabilidad . . . . .  | 15        |
| 2.3.3. Planos doble entropía y entropía-complejidad . . . . .  | 18        |
| 2.4. Cuantificador de entropías implementado en FPGA . . . . .   | 21        |
| 2.4.1. <i>Hardware</i> Implementado . . . . .  | 22        |
| 2.4.2. <i>Software</i> Implementado . . . . .  | 25        |
| 2.4.3. Resultados . . . . .  | 26        |
| 2.4.4. Discusión . . . . .   | 27        |
| 2.4.5. Conclusiones y trabajo futuro . . . . .   | 28        |
| 2.5. Dinámica de los ITQ's con AWGN y banda limitada . . . . .   | 28        |
| 2.5.1. Filtrado digital . . . . .  | 28        |
| 2.5.2. Resultados . . . . .  | 30        |
| 2.5.3. Discusión . . . . .   | 35        |
| <b>3. Sistemas caóticos</b>  | <b>37</b> |
| 3.1. Sistemas Caóticos . . . . .   | 37        |
| 3.2. Caos en redes neuronales . . . . .  | 38        |
| 3.2.1. El modelo de Hopfield . . . . .   | 39        |
| 3.2.2. Un caso de estudio . . . . .  | 41        |
| <b>4. El problema de la Aritmética Discreta</b>  | <b>47</b> |
| 4.1. Analysis of the digital implementation of a chaotic deterministic-stochastic attractor (EAMTA 2012) . . . . . | 47        |
| 4.2. Complexity of switching chaotic maps . . . . .  | 47        |
| 4.2.1. Introduction . . . . .  | 47        |
| 4.2.2. Information theory quantifiers . . . . .  | 49        |
| 4.2.3. Results . . . . .   | 52        |
| 4.2.4. Simple maps. . . . .  | 52        |
| 4.2.5. Conclusions . . . . .   | 56        |

|   |           |
|---|-----------|
| <b>5. Generadores de TRNG usando ROs en FPGA</b>              | <b>63</b> |
| 5.1. Introduction . . . . .                                   | 63        |
| 5.2. Determinación del <i>jitter</i> en <i>RO's</i> . . . . . | 65        |
| 5.3. Results . . . . .  | 66        |
| 5.4. Conclusions . . . . .                                    | 72        |
| <b>6. Conclusiones</b>  | <b>75</b> |
| <b>A. Field Programmable Gate Array (FPGA)</b>                | <b>77</b> |
| <b>Bibliografía</b>   | <b>79</b> |

# 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.

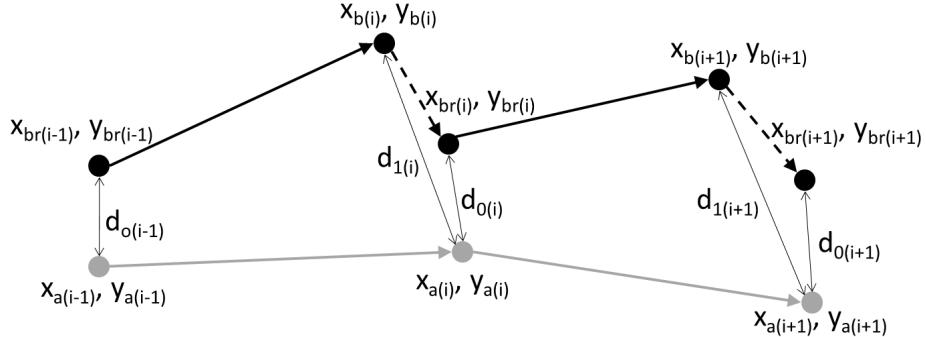


Figura 2.1: Algoritmo para calcular el MLE.

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

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

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

## 2.2. Algoritmo evolutivo para la búsqueda de caos

Propusimos emplear un método eurístico para buscar parámetros del sistema implementado de tal forma que se maximice la caoticidad de su salida. Nuestro trabajo tiene la ventaja que realiza una búsqueda inteligente mediante el empleo de un algoritmo genético, lo que minimiza el tiempo de cómputo.

Un algoritmo evolutivo es un método de búsqueda dirigido basado en la probabilidad. Un juego de entidades que representan posibles soluciones compite con otros, evolucionando en mejores soluciones [?].

Las entidades que representan posibles soluciones al problema son llamados *cromosomas* y el grupo de cromosomas es llamados *población inicial*.

Desde la población inicial, o los primeros padres, se genera un hijo mediante el cruce entre ellos. Luego, ellos son mutados en forma aleatoria para crear la próxima generación. Cada generación es comparada con la previa para descartar los “peor adaptados” así los coeficientes (cromosomas) mutan hacia los “mejor adaptados”.

Cuando se aplican estos algoritmos en funciones continuas, siempre convergen hacia el máximo local. Sin embargo, si el espacio de coeficientes es fractal,

existen áreas bien definidas en donde el la función objetivo es positiva, negativa, cero o no existente. Este es el caso si la función a maximizar es el MLE y el espacio de exploración es el de parámetros.

## Resultados

Para evaluar la viabilidad del método, se generó el siguiente algoritmo y se probó sobre el mapa logístico.

En la figura ?? podemos ver el diagrama de flujo principal. El bloque *Evolution* fue descompuesto en otro sub-diagrama para simplificar la descripción. Este segundo diagrama puede verse en la figura ??, esta surutina maneja la evolución de los parámetros.

El algoritmo inicia con una inicialización general de parámetros como el número máximo de generaciones *max\_gen*, el número máximo de mutaciones *max\_mut* y el número máximo de cambios en cada mutación *max\_step*. Luego se definen los primeros dos padres, ellos definirán los márgenes de búsqueda. Además se calcula su *fitness function Fp*. A partir de este punto se itera la segunda generación, se elige en forma aleatoria un valor de parámetro *r* con una distribución aleatoria entre los primeros dos padres, generando un nuevo hijo. Luego este hijo entra en la subrutina *Evolution* cuya salida es el valor de *r* evolucionado y su correspondiente *Fc*.

Luego se evalúa si este hijo avolucionó muy cerca de sus padres o no. Si la distancia entre ellos es más grande que el parámetro *max\_hop*, entonces este hijo es considerado como adulto, en caso contrario debe competir con su padre más cercano sobreviviendo el más apto.

Este proceso se repite hasta que se llega al máximo número de generaciones *max\_gen*. El grupo final de adultos es la solución al problema de buscar los máximos MLE locales.

La subrutina *Evolution* de la figura 2.3) es un algoritmo muy simple basado en mutaciones. El primer paso es generar una mutación del hijo con una probabilidad uniformemente distribuida entre  $\pm max\_step$ , tambien se calcula su *fitness function Fm*, que se compara con la del individuo original *Fc*. Entonces sobrevive el mejor adaptado para dar lugar a la siguiente mutación. Este procedimiento se repite hasta que se llega al máximo número de mutaciones *max\_mut*.

Como resultado podemos ver el *MLE* del mapa logístico en función de su único parámetro *r* en la figura 2.4. La línea continua muestra el *MLE* en pasos continuos de *r*, mientras que los puntos destacados son el resultado del algoritmo propuesto.

El bloque que calcula el *MLE* fue sintetizado y verificado experimentalmente en un Altera CYCLONE III FPGA y los resultados de la compilación mostrados en la figura 2.5. Los resultados del *Timing Analysis* reportan que la máxima frecuencia es de  $84,95MHz$ . El reporte de compilación muestra que la utilización de la lógica no excede el 20 %, es decir un total de 20307 de elementos lógicos, 54 % de los bits de memoria totales y 8 % de los multiplicadores embebidos.

En la figura 2.6 se muestra la salida del Signal Tap. La señal *salida* es la suma de los *MLE* luego de cada iteración. La segunda señal llamada *cuenta\_sal* corresponde a la sumatoria actual. Finalmente, cada flanco descendente de la señal *listoD1* indica que la salida es un dato válido. La salida fué procesada con Matlab para obtener la curva mostrada en la figura2.7. El valor del MLE

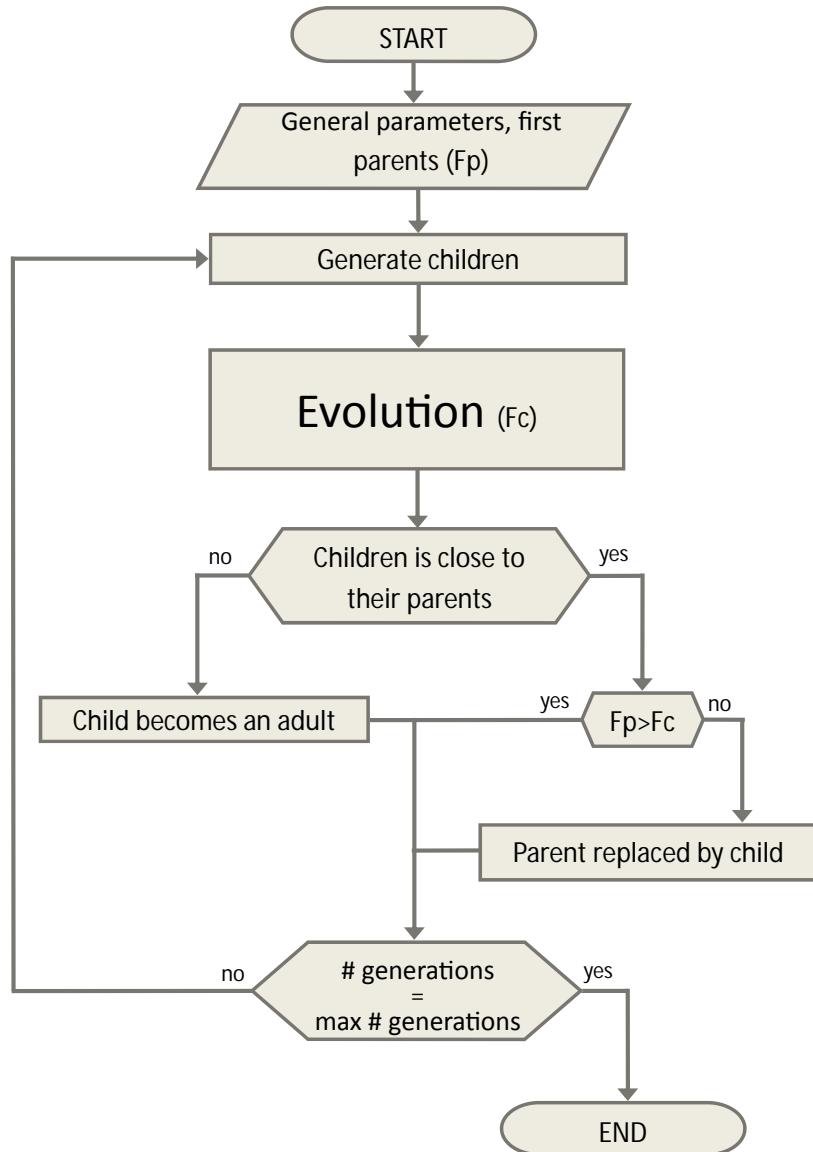


Figura 2.2: Main flow chart.

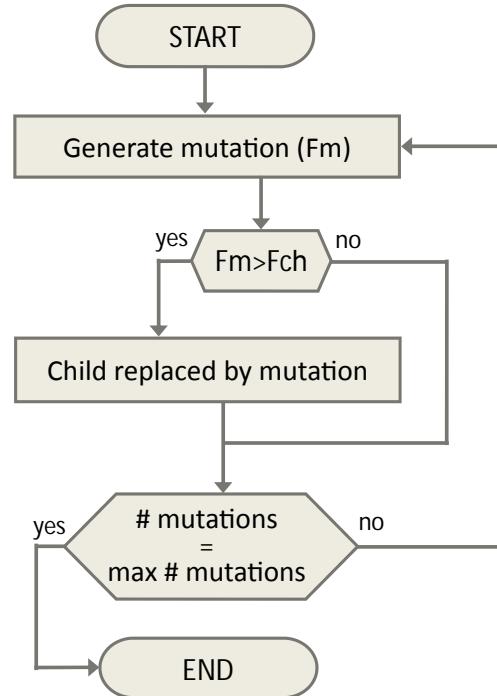


Figura 2.3: Evolution flow chart.

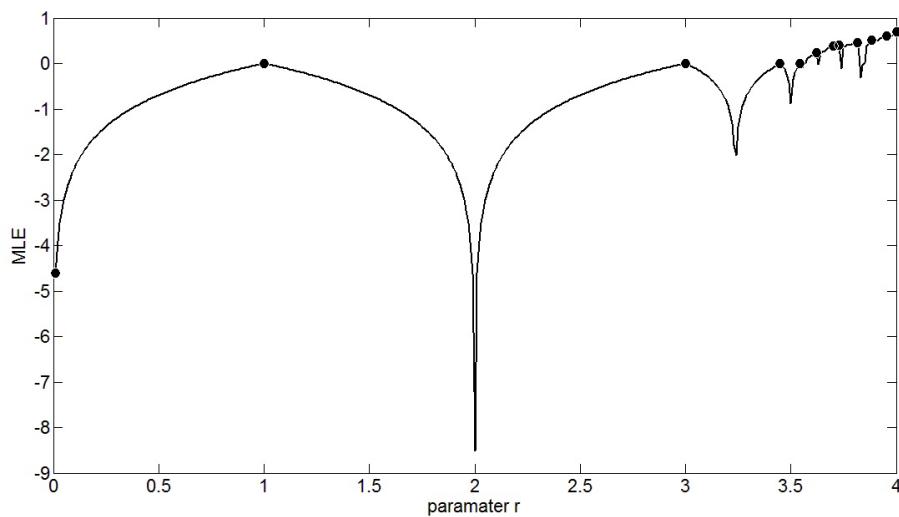


Figura 2.4: Algorithm results.

| Flow Summary                       |  |
|------------------------------------|--|
| Flow Status                        | Successful - Fri Apr 19 10:20:17 2013    |
| Quartus II 32-bit Version          | 12.1 Build 177 11/07/2012 SJ Web Edition |
| Revision Name                      | CalculaLyap                              |
| Top-level Entity Name              | TOP                                      |
| Family                             | Cyclone III                              |
| Device                             | EP3C120F780C7                            |
| Timing Models                      | Final                                    |
| Total logic elements               | 29,307 / 119,088 ( 25 % )                |
| Total combinational functions      | 26,048 / 119,088 ( 22 % )                |
| Dedicated logic registers          | 18,014 / 119,088 ( 15 % )                |
| Total registers                    | 18014                                    |
| Total pins                         | 197 / 532 ( 37 % )                       |
| Total virtual pins                 | 0  |
| Total memory bits                  | 2,133,356 / 3,981,312 ( 54 % )           |
| Embedded Multiplier 9-bit elements | 48 / 576 ( 8 % )                         |
| Total PLLs                         | 1 / 4 ( 25 % )                           |

Figura 2.5: Compilation report of the MLE calculator.

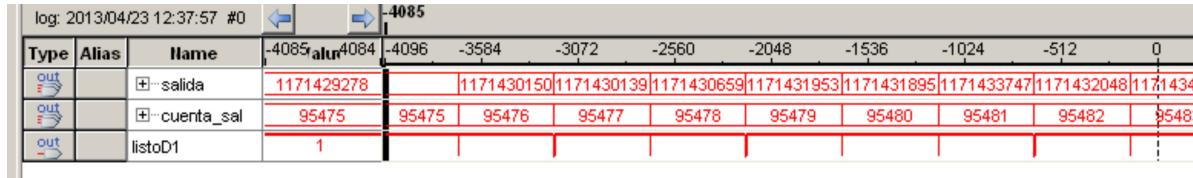


Figura 2.6: Signal Tap output.

en la iteración 250000 es 0,1415, lo que es consistente con el MLE obtenido con Matlab.

### 2.2.1. Estado actual del avance

Actualmente estamos en etapa de desarrollo de la implementación en hardware de este algoritmo. En este segundo caso, el sistema bajo prueba es la familia de mapas cuadráticos bidimensionales descritos en el capítulo ?? y cuya ecuación es ??.

$$\begin{aligned}
 x_{(i+1)} &= a_0 + a_1 x_{(i)} + a_2 x_{(i)}^2 + \\
 &\quad + a_3 x_{(i)} y_{(i)} + a_4 y_{(i)}^2 + a_5 y_{(i)} \\
 y_{(i+1)} &= b_0 + b_1 x_{(i)} + b_2 x_{(i)}^2 + \\
 &\quad + b_3 x_{(i)} y_{(i)} + b_4 y_{(i)}^2 + b_5 y_{(i)}
 \end{aligned} \tag{2.4}$$

La población inicial es de 12 coeficientes iniciales del mapa caótico empleado. En la figura 2.8 puede verse un diagrama en bloques general del sistema. Este consiste en dos bloques principales conectados al sistema caótico bajo prueba a

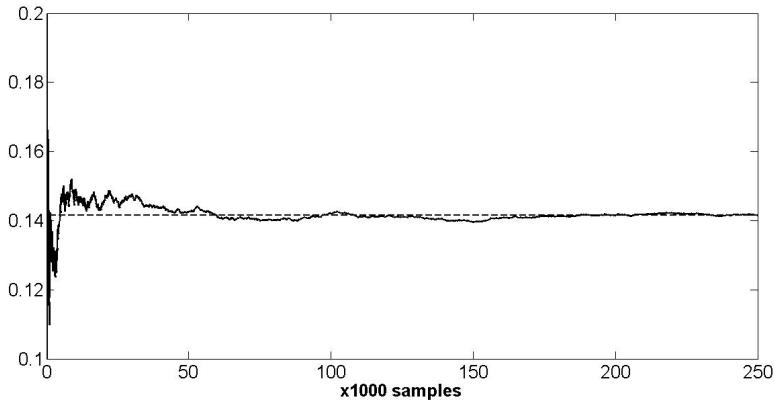


Figura 2.7: Lyapunov curve

través de una interface wishbone. Esto independiza el sistema del cuantificador y permite cambiar fácilmente el sistema bajo prueba.

El sistema caótico es duplicado en dos bloques, *SystemA* y *SystemB*. Cada uno de ellos es inicializado con los puntos en el espacio de fases  $(x_{a(i-1)}, y_{a(i-1)})$  y  $(x_{br(i-1)}, y_{br(i-1)})$  respectivamente. Cuando los sistemas caóticos terminan de calcular sus salidas, la señal digital *habilita* se pone en cero y el bloque *D1* es habilitado para calcular la distancia euclídea entre las salidas  $(x_{a(i)}, y_{a(i)})$  y  $(x_{br(i)}, y_{br(i)})$ .

Luego se habilitan los bloques concurrentes *L* y *Rel*. Los puntos relocalizados que alimentan al bloque *SystemB* son calculados por el bloque *Rel*. Este bloque solo necesita los valores actuales de  $d_1$  los valores previos de  $d_0$ , como se muestra en la ecuación 2.3. Cuando los puntos relocalizados  $x_{br(i)}$  y  $y_{br(i)}$  están disponibles, los bloques *SystemA* y *SystemB* se habilitan para obtener la siguiente iteración. También se habilita el bloque *D<sub>0</sub>* para calcular el valor actual de  $d_{0(i)}$ , que se utilizará en la siguiente iteración.

Finalmente, el bloque *L* realiza la división entre  $d_0$  y  $d_1$  para luego calcular el valor absoluto y el logaritmo de esta división. El mapa es iterado  $N = 250000$  veces y el resultado de la sumatoria dividido por  $N$  para asegurar la convergencia del método.

Cada bloque fue implementado utilizando lenguaje VHD e IP *cores* provistos por *Altera* (*megafunctions*) cada vez que fue posible, debido a que estos *cores* están optimizados para este dispositivo. Las operaciones de punto flotante como las sumas, multiplicaciones, valores absolutos y logaritmos fueron calculadas con dichas *megafunctions*.

La figura 2.9 muestra la implementación del bloque *D1* en el entorno gráfico Quartus. Los puntos de salida y entrada *a* y *b*, son tomados luego de que la señal *habilita* se pone en cero. Entonces, las señales son procesadas de acuerdo a la ecuación 2.1 para calcular la distancia euclídea  $d_1$ .

La lógica del algoritmo genético fue implementada en lenguaje VHD. Esta lógica junto con el registro de los 12 coeficientes se muestran en la figura 2.10. También se muestran los resultados de la compilación en la figura 2.11. Puede verse que esta implementación ocupa pocos recursos del dispositivo.

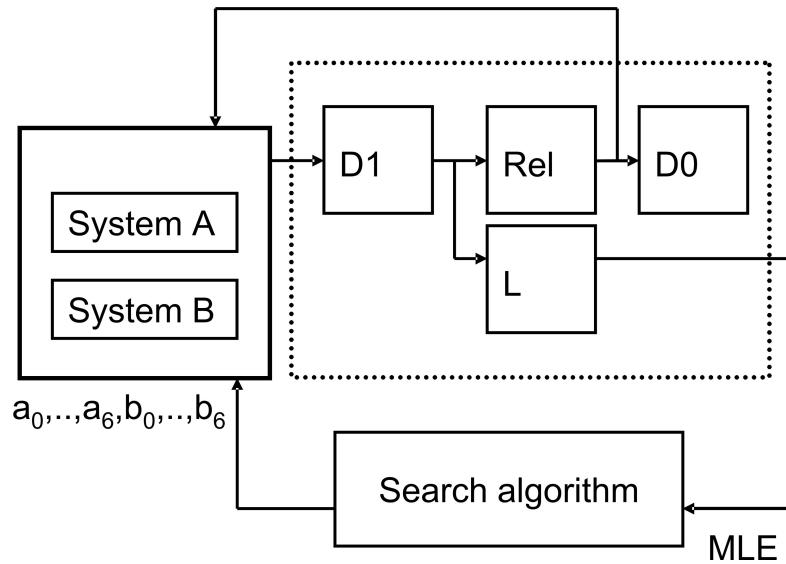
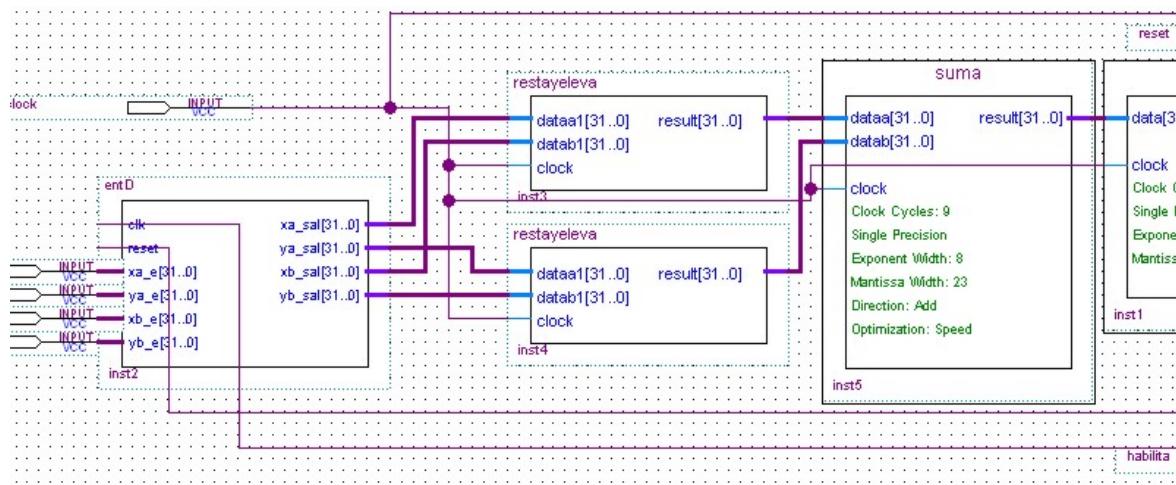


Figura 2.8: Enabling flow of the System.

Figura 2.9:  $D_1$  Block.

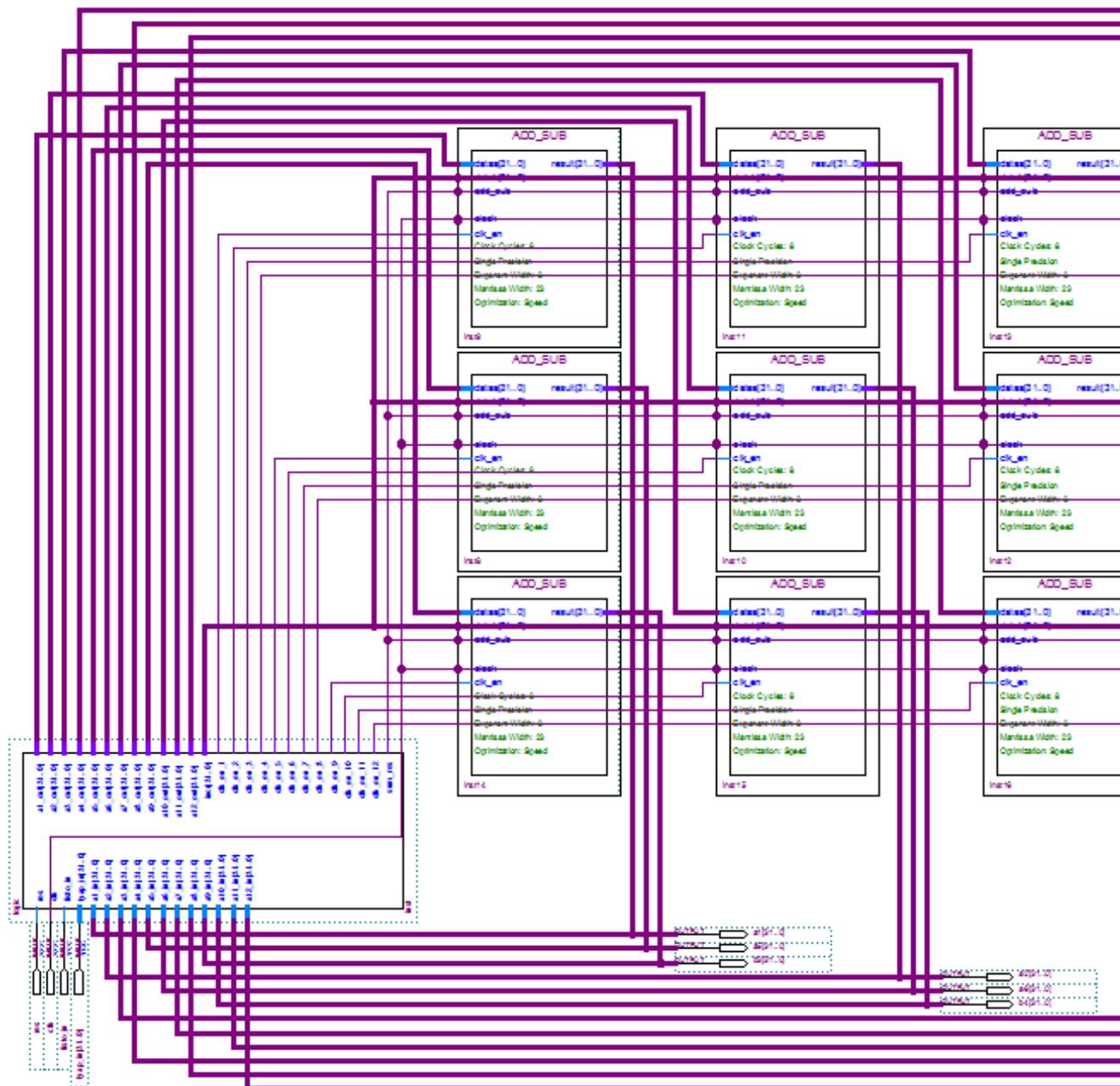


Figura 2.10: Circuit of the evolutive algorithm.

| Flow Summary                       |   |
|------------------------------------|---|
| Flow Status                        | Successful - Mon Apr 22 11:03:28 2013         |
| Quartus II 64-Bit Version          | 12.1 Build 243 01/31/2013 SP 1 SJ Web Edition |
| Revision Name                      | feedback                                      |
| Top-level Entity Name              | feedback                                      |
| Family                             | Cyclone III                                   |
| Device                             | EP3C120F780C7                                 |
| Timing Models                      | Final   |
| Total logic elements               | 10,136 / 119,088 ( 9 % )                      |
| Total combinational functions      | 9,700 / 119,088 ( 8 % )                       |
| Dedicated logic registers          | 5,093 / 119,088 ( 4 % )                       |
| Total registers                    | 5093  |
| Total pins                         | 419 / 532 ( 79 % )                            |
| Total virtual pins                 | 0   |
| Total memory bits                  | 432 / 3,981,312 ( < 1 % )                     |
| Embedded Multiplier 9-bit elements | 0 / 576 ( 0 % )                               |
| Total PLLs                         | 0 / 4 ( 0 % )                                 |

Figura 2.11: Compilation report of the evolutive algorithm.

### 2.3. Cuantificadores de la teoría de la información

Los sistemas dinámicos son sistemas que evolucionan en el tiempo. En la práctica, solo es posible medir una serie de tiempo escalar  $X(t)$  la cual puede ser función de las variables  $V = \{v_1, v_2, \dots, v_k\}$  que describe la dinámica subyacente (por ejemplo  $dV/dt = f(V)$ ). Tratamos de inferir propiedades de un sistema no conocido a partir del análisis de los datos guardados de datos observacionales. ¿Cuanta información revelan estos datos sobre la dinámica del sistema o procesos subyacentes?

El contenido de información de un sistema se evalúa típicamente mediante una función de distribución de probabilidad (PDF)  $P$  que describe la distribución de alguna cantidad mensurable o observable, generalmente una serie de tiempo  $X(t)$ . Podemos definir los cuantificadores de la Teoría de la Información como medidas capaces de caracterizar las propiedades relevantes de las PDFs asociadas a estas series temporales, y de esta manera debemos extraer juiciosamente información sobre el sistema dinámico en estudio. Estos cuantificadores representan métricas en el espacio de PDFs para conjuntos de datos, permitiendo comparar diferentes conjuntos y clasificarlos de acuerdo a sus propiedades de procesos subyacentes, de manera amplia, estocástica vs. determinística.

En nuestro caso, nos interesa la dinámica caótica. Por lo tanto, nos centramos en las métricas que toman en cuenta el orden temporal de las observaciones de forma explícita; es decir, el enfoque es fundamentalmente de naturaleza *causal* y *estadística* en la naturaleza. En un enfoque puramente estadístico, las correlaciones entre los valores sucesivos de las series temporales se ignoran o

simplemente se destruyen a través de la construcción del PDF; mientras que un enfoque causal se centra en las PDFs de secuencias de datos.

Los cuantificadores seleccionados se basan en el recuento de símbolos y en la estadística de patrones de orden. Las métricas a utilizar pueden clasificarse de forma amplia en dos categorías: las que cuantifican el *contenido de información* de los datos en comparación con los relacionados con su *complejidad*. Obsérvese que aquí nos estamos refiriendo al espacio de funciones de densidad de probabilidad, no al espacio físico. Para clarificar y simplificar, introducimos solamente los cuantificadores de la Teoría de la Información que se definen en PDFs discretas, ya que solo estamos tratando con datos discretos (series temporales). Sin embargo, todos los cuantificadores también tienen definiciones para el caso continuo [?].

### 2.3.1. Entropía de Shannon y Complejidad Estadística

La entropía es una cantidad básica que puede considerarse como una medida de la incertidumbre asociada (información) al proceso físico descrito por  $P$ . Al tratar con el contenido de la información, la entropía de Shannon se considera a menudo como la fundamental y más natural [?]. Considerada como una medida de la incertidumbre, es el ejemplo más paradigmático de estos cuantificadores de información.

Sea  $P = \{p_i; i = 1, \dots, N\}$  con  $\sum_{i=1}^N p_i = 1$ , una distribución de probabilidad discreta, con  $N$  el número de estados posibles del sistema bajo estudio. La medida de la información logarítmica de shannon se denota como

$$S[P] = - \sum_{i=1}^N p_i \ln [p_i] . \quad (2.5)$$

Si  $S[P] = S_{\min} = 0$ , estaremos en posición de predecir con total certeza cuáles de los posibles resultados  $i$ , cuyas probabilidades están dadas por  $p_i$ , tendrán lugar realmente. Nuestro conocimiento del proceso subyacente descrito por la distribución de probabilidad es máximo en este caso. Por el contrario, nuestro conocimiento es mínimo para una distribución uniforme  $P_e = \{p_i = 1/N; i = 1, \dots, N\}$  dado que cada resultado exhibe la misma probabilidad de ocurrencia, y la incertidumbre es máxima, es decir,  $S[P_e] = S_{\max} = \ln N$ . Estas dos situaciones son casos extremos, por lo tanto nos centramos en la entropía de Shannon "normalizada",  $0 \leq H \leq 1$ , dada como

$$H[P] = S[P]/S_{\max} . \quad (2.6)$$

Contrariamente al contenido de la información, no existe una definición universalmente aceptada de complejidad. Aquí, nos centramos en describir la *complejidad de las series temporales* y no nos referimos a la complejidad de los *sistemas* subyacentes. Un sistema complejo no genera necesariamente una salida compleja. De hecho, los modelos "simples" pueden generar datos complejos, mientras que los sistemas "complicados" pueden producir datos de salida de baja complejidad [?].

Una noción intuitiva de una complejidad cuantitativa atribuye valores bajos tanto a datos perfectamente ordenados (es decir, con entropía de Shannon que se va desapareciendo) como a datos aleatorios no correlacionados (con entropía

Shannon máxima). Por ejemplo, la complejidad estadística de una simple oscilación o tendencia (ordenada), pero también de ruido blanco no correlacionado (no ordenado) sería clasificada como baja. Entre los dos casos de mínima y máxima entropía, los datos son más difíciles de caracterizar y por lo tanto la complejidad debe ser mayor. Buscamos alguna función  $C[P]$  que cuantifique las estructuras presentes en los datos que se alejan de estos dos casos. Estas estructuras se relacionan con la organización, la estructura correlacional, la memoria, la regularidad, la simetría, los patrones y otras propiedades [?].

Asumimos que el grado de estructuras correlacionales sería capturado adecuadamente por algún funcional  $C[P]$  de la misma manera que la entropía de Shannon  $S[P]$  [?] “capta” la aleatoriedad. Claramente, las estructuras ordinales presentes en un proceso no son cuantificadas por medidas de aleatoriedad y, por consiguiente, son necesarias medidas de complejidad estadística o estructural para una mejor comprensión (caracterización) de la dinámica del sistema representada por sus series temporales [?].

Una medida adecuada de complejidad puede definirse como el producto de una medida de información y una medida de desequilibrio, es decir, algún tipo de distancia de la distribución equiprobable de los estados accesibles de un sistema. En este sentido, en [?] los autores introdujeron una eficaz *Medida de Complejidad Estadística* (SCM)  $C$ , que es capaz de detectar detalles esenciales de los procesos dinámicos subyacentes al conjunto de datos. Basado en el trabajo de López-Ruiz [?], esta medida de complejidad estadística [?, ?] se define a través de la forma del producto

$$C[P] = Q_J[P, P_e] \cdot H[P] \quad (2.7)$$

de la entropía de Shannon normalizada  $H$ , ver eq. (2.6), y el desequilibrio  $Q_J$  definido en términos de la divergencia de Jensen-Shannon  $J[P, P_e]$ . Esto es,

$$Q_J[P, P_e] = Q_0 J[P, P_e] = Q_0 \{S[(P + P_e)/2] - S[P]/2 - S[P_e]/2\}, \quad (2.8)$$

en la divergencia de Jensen-Shannon mencionada arriba,  $Q_0$  es una constante de normalización tal que  $0 \leq Q_J \leq 1$ :

$$Q_0 = -2 \left\{ \frac{N+1}{N} \ln(N+1) - \ln(2N) + \ln N \right\}^{-1}, \quad (2.9)$$

y es igual a la inversa del máximo valor posible de  $J[P, P_e]$ . Este valor es obtenido cuando una de las componentes de  $P$ , digamos  $p_m$ , es igual a uno y todos los  $p_j$  restantes son cero.

La divergencia de Jensen-Shannon, que cuantifica la diferencia entre las distribuciones de probabilidad, es especialmente útil para comparar la composición simbólica entre diferentes secuencias [?]. Obsérvese que la SCM introducida anteriormente depende de dos distribuciones de probabilidad diferentes: una asociada con el sistema analizado,  $P$ , y la otra con la distribución uniforme,  $P_e$ . Además, se demostró que para un valor dado de  $H$ , el rango de valores posibles de  $C$  varía entre un mínimo  $C_{min}$  y un máximo  $C_{max}$ , restringiendo los posibles valores del SCM [?].

Por lo tanto, está claro que información adicional importante relacionada con la estructura correlacional entre los componentes del sistema físico se proporciona evaluando la medida de la complejidad estadística.

### 2.3.2. Determinación de la distribución de probabilidad

La evaluación de los cuantificadores derivados de la Teoría de la Información supone algún conocimiento previo sobre el sistema; específicamente para aquellos introducidos previamente (entropía de Shannon y complejidad estadística), una distribución de probabilidad asociada a la serie temporal en análisis debe proporcionarse antes. La determinación del PDF más adecuado es un problema fundamental porque la PDF  $P$  y el espacio de muestra  $\Omega$  están intrincadamente vinculados.

Las metodologías usuales asignan a cada valor de la serie  $X(t)$  (o conjunto de valores consecutivos no superpuestos) un símbolo de un alfabeto finito  $A = \{a_1, \dots, a_M\}$ , creando así una *secuencia simbólica* que puede considerarse como una descripción de la serie cronológica en cuestión. Como consecuencia, las relaciones de orden y las escalas temporales de la dinámica se pierden por completo.

Es importante resaltar que  $P$  en si, no es un objeto con una definición única y existen varias aproximaciones para “asociar” una dada  $P$  con una dada serie de tiempo. Solo para mencionar algunos criterios de extracción utilizados frecuentemente en la literatura: *a)* histogramas de series temporales [?], *b)* dinámica simbólica binaria [?], *c)* análisis de Fourier [?], *d)* transformadas wavelet [?, ?], *e)* PDF de particiones [?], *f)* PDF de permutaciones [?, ?], *g)* PDF discreta [?], etc. Hay una amplia libertad para elegir entre ellas y la aplicación específica debe ser analizada para hacer una buena elección.

Se puede incorporar debidamente la información causal si se incluye información sobre la dinámica pasada del sistema en la secuencia simbólica, es decir, los símbolos del alfabeto  $A$  se asignan a una porción del espacio de fase o trayectoria. Bandt y Pompe (BP) [?] introdujeron una metodología simbólica simple y robusta que toma en cuenta el ordenamiento temporal de las series temporales comparando valores vecinos en una serie temporal. La propiedad de causalidad de la PDF permite que los cuantificadores (basados en esta PDF) discriminan entre sistemas determinísticos y estocásticos [?]. Los datos simbólicos son: *(i)* creados por la clasificación de los valores de la serie; y *(ii)* definidos por el reordenamiento de los datos embebidos en orden ascendente, lo que equivale a una reconstrucción de espacio de fase con dimensión de embedding (longitud de patrón)  $D$  y retardo de tiempo  $\tau$ . De esta forma, es posible cuantificar la diversidad de los símbolos de ordenación (patrones) derivados de una serie temporal escalar. Obsérvese que la secuencia de símbolos apropiada surge naturalmente de la serie temporal, y no se necesitan suposiciones basadas en modelos. El procedimiento es el siguiente:

- Dada una serie  $\{x_t; t = 0, \Delta t, \dots, N\Delta t\}$ , se genera una secuencia de vectores de longitud  $D$ .

$$(s) \mapsto (x_{t-(d-1)\Delta t}, x_{t-(d-2)\Delta t}, \dots, x_{t-\Delta t}, x_t) \quad (2.10)$$

Cada vector resulta ser la “historia” del valor  $x_t$ . Evidentemente, cuanto más larga sea la longitud de los vectores  $D$ , mayor será la información sobre la historia de los vectores, pero se requiere un valor más alto de  $N$  para tener una estadística adecuada.

- Las permutaciones  $\pi = (r_0, r_1, \dots, r_{D-1})$  de  $(0, 1, \dots, D - 1)$  es llamado

“patrón de orden” de tiempo  $t$ , definido por:

$$x_{t-r_{D-1}\Delta t} \leq x_{t-r_{D-2}\Delta t} \leq \cdots \leq x_{t-r_1\Delta t} \leq x_{t-r_0\Delta t} \quad (2.11)$$

Para obtener un resultado único se considera  $r_i < r_{i-1}$  si  $x_{t-r_i\Delta t} = x_{t-r_{i-1}\Delta t}$ . De esta forma, todas las  $D!$  permutaciones posibles  $\pi$  de orden  $D$ , y la PDF  $P = \{p(\pi)\}$  es definida como:

$$p(\pi) = \frac{\#\{s | s \leq N - D + 1; (s) \text{ has type } \pi\}}{N - D + 1} \quad (2.12)$$

En estas últimas expresiones, el símbolo  $\#$  denota cardinalidad.

Por lo tanto, una distribución de probabilidad de patrones de orden  $P = \{p(\pi_i), i = 1, \dots, D!\}$  se obtiene de la serie temporal. De esta manera, el vector definido por la ecuación (4.6) se convierte en un símbolo único  $\pi$ . Se establece  $r_i < r_{i-1}$  si  $x_{s-r_i} = x_{s-r_{i-1}}$  para la obtener una única solución. La única condición para la aplicabilidad del método BP es una suposición estacionaria muy débil: para  $k \leq D$ , la probabilidad para  $x_t < x_{t+k}$  no debe depender de  $t$ . Con respecto a la selección de los parámetros, Bandt y Pompe sugirieron trabajar con  $3 \leq D \leq 6$  para longitudes de series de tiempo típicas, y específicamente se consideró un retraso de tiempo  $\tau = 1$  en su publicación principal.

Para destacar la diferencia entre una *P causal* y una *no causal*, consideremos una serie de valores  $X = \{x_i, i = 1, 2, \dots\}$  generada por la función *randn* de Matlab's <sup>©</sup>; consideremos también la serie  $Y = \{y_i, i = 1, 2, \dots\}$  como la resultante de ordenar la serie  $X$  en forma ascendente. Esto se puede ver en el ejemplo en la figura 2.12, en la figura 2.12a se muestran 1000 valores sorteados con una distribución uniforme entre 0 y 1, también mostramos en la figura 2.12b la versión ordenada de la serie de la figura 2.12a, son los mismos valores pero ordenados en forma ascendente. Una *P no causal* es el histograma normalizado que mostramos en las figuras 2.12c y 2.12d, en donde puede verse que  $P(X)$  es idéntica a  $P(Y)$ , por lo que todos los cuantificadores que se calculen a partir de ellas serán idénticos para las dos series. Una *P causal* puede ser obtenida mediante el procedimiento de Bandt & Pompe descripto arriba, en este caso  $P(X)$  de la figura 2.12e es bastante uniforme y  $P(Y)$  de la figura 2.12f tiene una forma tipo delta. En este caso, *P* registra que  $Y$  es monótonamente creciente y presenta un solo patrón de orden.

Recientemente, la entropía de permutación se amplió para incorporar también información de amplitud. Ponderar las probabilidades de patrones individuales de acuerdo a su varianza mitiga los problemas potenciales con respecto a los patrones de “alto ruido, baja señal”, porque los patrones de baja varianza que están fuertemente afectados por el ruido se ponderan en las distribuciones de patrones ordinales ponderados resultantes. Por lo tanto, una posible desventaja de las estadísticas de los patrones ordinales, es decir, la pérdida de información de amplitud, se puede abordar mediante la introducción de pesos con el fin de obtener una “entropía de permutación ponderada (WPE)” [?]. Los pesos no normalizados se calculan para cada ventana temporal para la serie de tiempo  $X$ , tal que

$$w_j = \frac{1}{D} \sum_{k=1}^D \left( x_{j+k-1} - \bar{X}_j^D \right)^2. \quad (2.13)$$

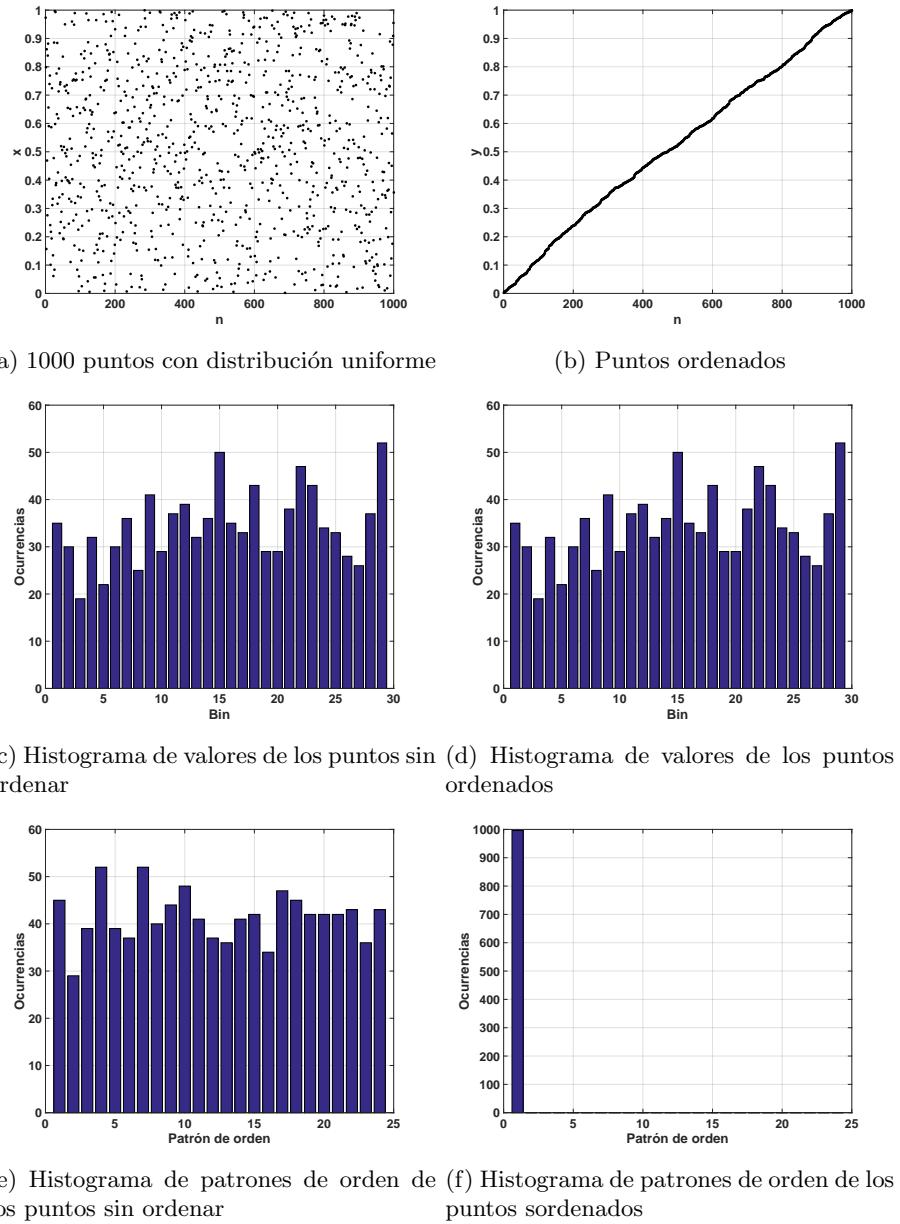


Figura 2.12: Comparación entre histogramas causal y no causal

En la ecuación anterior  $x_{j+k-1} - \bar{X}_j^D$  denota la media aritmética del actual vector de embedding de longitud  $D$  y su varianza  $w_j$  se utiliza entonces para ponderar las frecuencias relativas de cada patrón ordinal  $p_j$ . Originalmente, se propuso esta técnica para discriminar patrones sumergidos en un bajo nivel de ruido. Nosotros también aprovechamos el hecho de que los puntos fijos no se computan en el WPE.

Se calculó la entropía de Shannon normalizada  $H$  y la complejidad estadística  $C$  de estas PDFs, y los valores obtenidos se denotan como:

- $H_{hist}$ , es la entropía de Shannon normalizada aplicada a una PDF no causal  $P_{hist}$
- $H_{BP}$ , es la entropía de Shannon normalizada aplicada a una PDF causal  $P_{BP}$
- $H_{BPW}$ , es la entropía de Shannon normalizada aplicada a una PDF causal con contribuciones de amplitud  $P_{BPW}$
- $C_{BP}$ , es la complejidad estadística normalizada aplicada a una PDF causal  $P_{BP}$
- $C_{BPW}$ , es la complejidad estadística normalizada aplicada a una PDF causal con contribuciones de amplitud  $P_{BPW}$

### 2.3.3. Planos de doble entropía y entropía-complejidad

Una visualización particularmente útil de los cuantificadores de la Teoría de la Información es su yuxtaposición en los gráficos bidimensionales. Se definen cuatro planos de información:

1. Entropía causal vs. entropía no-causal,  $H_{BP} \times H_{hist}$
2. Entropía causal con contribución de amplitudes vs. entropía no-causal,  $H_{BPW} \times H_{hist}$
3. Complejidad causal vs. entropía causal,  $C_{BP} \times H_{BP}$
4. Complejidad causal con contribución de amplitudes vs. entropía causal con contribución de amplitudes,  $C_{BPW} \times H_{BPW}$

Estas herramientas de diagnóstico demostraron ser particularmente eficientes para distinguir entre el caos determinista y la naturaleza estocástica de una serie de tiempo ya que los cuantificadores de permutación tienen comportamientos distintos para diferentes tipos de procesos.

En la Fig. ?? se muestran los planos  $H_{BP} \times H_{hist}$  y  $H_{BPW} \times H_{hist}$  colapsados en un mismo plano. En este plano un valor más alto en cualquiera de las entropías,  $H_{BP}$ ,  $H_{BPW}$  o  $H_{hist}$ , implica una mayor uniformidad de la PDF implicada. El punto (1, 1) representa el caso ideal con histograma uniforme y distribución uniforme de los patrones de orden. Mostramos algunos puntos relevantes como ejemplo.

El ruido aleatorio blanco ideal con distribución uniforme da un punto en  $(H_{hist}, H_{BP}) = (1, 1)$  representado por un círculo azul, un círculo rojo en la misma posición muestra los resultados cuando se incluyen las contribuciones de

amplitud  $(H_{hist}, H_{BPW}) = (1, 1)$ . Si ordenamos el vector ideal con distribución uniforme de forma ascendente, los puntos resultantes se muestran con un cuadrado azul  $(H_{hist}, H_{BP}) = (1, 0)$  y un cuadrado rojo  $(H_{hist}, H_{BPW}) = (0, 1)$ , este ejemplo ilustra la complementariedad de  $H_{hist}$  y  $H_{BP}$ .

Las estrellas azules y rojas muestran  $(H_{hist}, H_{BP})$  y  $(H_{hist}, H_{BPW})$  respectivamente aplicadas a una señal de diente de sierra. Los valores están perfectamente distribuidos en todos los intervalos, pero sólo aparecen unos pocos patrones de orden, esto explica el alto  $H_{hist}$  y bajo  $H_{BP}$ . La frecuencia de aparición de patrones de baja amplitud es mayor que los patrones de alta amplitud, entonces la PDF con contribuciones de amplitud es más uniforme y  $H_{BPW}$  es un poco más alto que  $H_{BP}$ . Cuando la señal de diente de sierra está contaminada con ruido blanco, se incrementan  $H_{BP}$  y  $H_{BPW}$  como se muestra con triángulos azules y rojos. Es evidente que aparecen nuevos patrones de orden y tanto  $H_{BP}$  como  $H_{BPW}$  muestran valores más altos que los casos no contaminados, sin embargo el incremento de  $H_{BPW}$  es menor que  $H_{BP}$  mostrando que la técnica de registrar contribuciones de amplitud añade alguna inmunidad al ruido.

Finalmente, se evaluaron los cuantificadores de una secuencia de un mapa logístico que converge a un punto fijo, en todos los casos la longitud del vector de datos permanece constante y la longitud de transitorio es variable. Los resultados obtenidos sin las contribuciones de amplitud se representan en puntos azules, convergen a  $(H_{hist}, H_{BP}) = (0, 0)$  a medida que la longitud de transitorio se hace más corta, sin embargo  $H_{BPW}$  (puntos rojos) permanece constante para todos los casos. El último punto en  $(H_{hist}, H_{BP}) = (0, 0)$  corresponde a un vector de ceros, en este caso el histograma de patrones de orden con contribuciones de amplitud es también un vector nulo y  $H_{BPW}$  no se puede calcular. A través de este último ejemplo, mostramos que la convergencia a un punto fijo puede ser detectada por la información conjunta de  $H_{BP}$  y  $H_{BPW}$ .

En la figura ?? se muestra el plano causal  $H_{BP} \times C_{BP}$ . Podemos ver que no toda la región  $0 < H_{BP} < 1, 0 < C_{BP} < 1$  es alcanzable, de hecho, para cualquier PDF los pares  $(H, C)$  de valores posibles caen entre dos curvas extremas en el plano  $H_{BP} \times C_{BP}$  cite Anteneodo1996. Los mapas caóticos tienen entropía intermedia  $H_{BP}$ , mientras que su complejidad  $C_{BP}$  alcanza valores mayores, muy cercanos a los del límite de complejidad superior [?, ?]. Para procesos regulares, la entropía y la complejidad tienen valores pequeños, cercanos a cero. Los procesos estocásticos no correlacionados se ubican en la localización planar asociada con  $H_{BP}$  cerca de uno y  $C_{BP}$  cerca de cero. Los sistemas aleatorios ideales que tienen un Bandt & Pompe PDF uniforme, están representados por el punto  $(1, 0)$  citeGonzalez2005 y una PDF tipo delta corresponde al punto  $(0, 0)$ .

En la figura ?? mostramos  $H_{BP} \times C_{BP}$  con y sin contribuciones de amplitud. Se muestran los mismos puntos de muestra para ilustrar las posiciones planas para diferentes vectores de datos.

En ambos planos de información  $H_{BP} \times H_{hist}$  en la Fig. 2.13 y  $H_{BP} \times C_{BP}$  en Fig. 2.14, los datos estocásticos, caóticos y deterministas están claramente localizados en diferentes posiciones planares.

También usamos el número de patrones perdidos MP como un cuantificador [?]. Como mostraron recientemente Amigó y colaboradores [?, ?, ?, ?], en el caso de mapas deterministas, no todos los patrones de orden posibles pueden materializarse efectivamente en órbitas. De hecho, la existencia de estos patrones de orden faltantes se convierte en un hecho persistente que puede considerarse

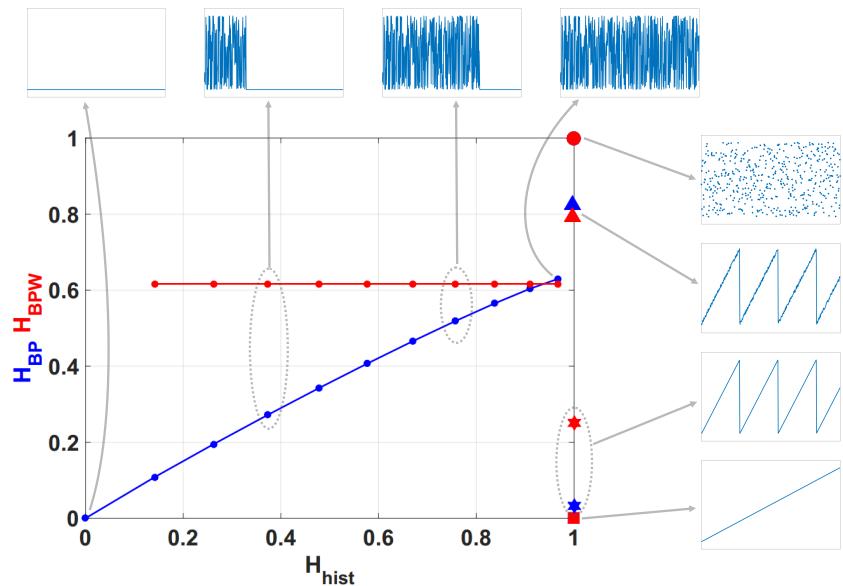


Figura 2.13: Causal-Non causal Entropy plane.

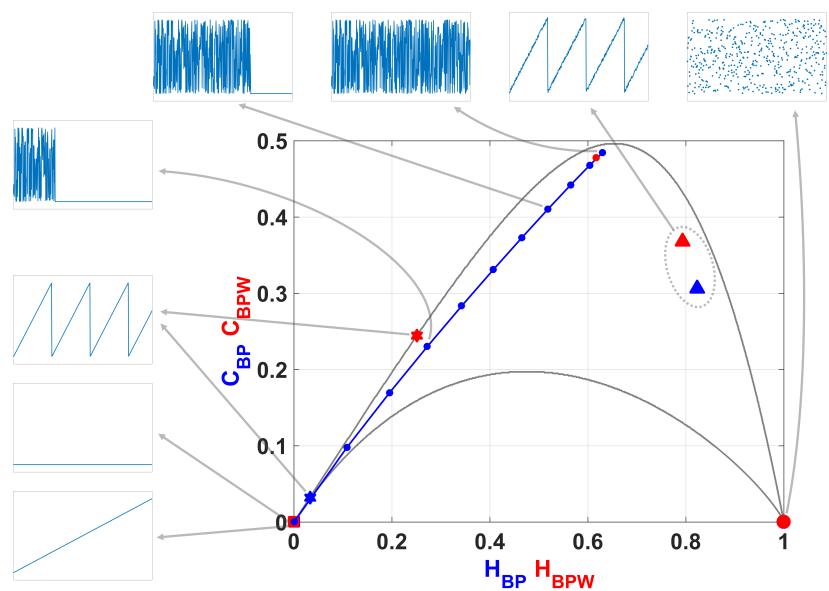


Figura 2.14: Causal Entropy-Complexity plane.

como una nueva propiedad dinámica. Por lo tanto, para una longitud de patrón fija (dimensión de embedding  $D$ ) el número de patrones perdidos de una serie temporal (patrones no observados) es independiente de la longitud de la serie  $N$ . Obsérvese que esta independencia no caracteriza otras propiedades de la serie como la proximidad y la correlación [?, ?].

Existen fuentes bibliográficas para una discusión completa sobre la conveniencia de usar estos cuantificadores [?, ?, ?, ?, ?, ?, ?, ?].

### Entropías diferenciales

La entropía de Shannon  $S(P)$  es el punto de partida para otros cuantificadores:

1. Entropía normalizada  $H(P)$ : es la entropía de Shannon dividida por su valor máximo. Por ejemplo, si usamos  $S_2$  (ver arriba), se obtiene la entropía máxima para equiprobabilidad entre dos símbolos. Su valor es  $S_{max} = -1/2\log(1/2) - 1/2\log(1/2) = \log(2) = 1$ ; entonces, la entropía normalizada es  $H_2 = S_2$ . Si usamos  $S_W$  la equiprobabilidad entre las  $2^W$  posibles palabras (números decimales de  $W$ -bits) produce  $S_{max} = W$  y  $H_W = S_W/W$ . Finalmente, para  $S_{BP}^{(D)}$  la equiprobabilidad entre los  $D!$  patrones de orden produce  $S_{max} = \log(D!)$  y  $H_{BP}^{(D)} = S_{BP}^D/\log(D!)$ .
2. Entropía diferencial o condicional  $h$  y  $h^*$  son:

$$h = S_{W+1} - S_W \quad (2.14)$$

$$h^* = S_{BP}^{(D+1)} - S_{BP}^{(D)} \quad (2.15)$$

En las expresiones de arriba  $W = 1, 2, \dots$  y  $D = 2, 3, \dots$ ,  $S_0 = 0$  y  $S_{BP}^{(1)} = 0$ . Esta entropía diferencial o condicional da la cantidad promedio de información requerida para predecir el símbolo  $(W + 1)$  (o  $(D + 1)$ ), dado los  $W$  (o  $D$ ) símbolos precedentes.

3. Finalmente, las *rate entropies*  $h_0$  y  $h_0^*$  [?, ?] son dadas por:

$$h_0 = \lim_{W \rightarrow \infty} h = \lim_{W \rightarrow \infty} S_W/W \quad (2.16)$$

$$h_0^* = \lim_{D \rightarrow \infty} h^* = \lim_{D \rightarrow \infty} S_{BP}^{(D)}/(D - 1) \quad (2.17)$$

## 2.4. Cuantificador de entropías implementado en FPGA

En esta sección se describe la implementación de un sistema de medición de entropías. El diseño fue optimizado para ser implementado en un microcontrolador simple y pequeño, conservando una precisión aceptable. El sistema permite medir entropías a señales generadas internamente por código y a señales externas analógicas muestreadas. Se utilizó la placa de desarrollo *M1AFS-embedded kit* de ACTEL. En la FPGA (*Field Programmable Gate Array*) se instanció un microcontrolador 8051 al que se programó en lenguaje C. Se detalla el diseño del *hardware* y *software* y los resultados obtenidos.

Este trabajo se enmarca en un proyecto más ambicioso, que se propone el desarrollo e implementación en *hardware* de herramientas para el análisis de sistemas alineales. Contar con estas herramientas supondrá un avance significativo en el campo de la implementación de los sistemas no lineales. Permitiría comprender y describir con mayor precisión el comportamiento de la versión digital de este tipo de sistemas. El paquete completo de herramientas que nos proponemos implementar consta de:

- funcionales de la distribución de probabilidad: entropía de Shannon, desequilibrio estadístico y complejidad estadística;
- cuantificadores de la serie temporal, en especial exponentes de Lyapunov, autocorrelación, correlación cruzada y dimensiones fractales;
- operador de Perron Frobenius, y cuantificadores de diagramas de recurrencias;
- tests estadísticos propuestos en los bancos estandarizados para el estudio de generadores de números aleatorios (Marsaglia, NIST, etc.).

Al momento hay muy poca bibliografía sobre implementaciones en *hardware* de estas herramientas[?].

En el caso particular de la entropía es empleada en diversas aplicaciones, como por ejemplo, en la detección de anomalías en flujos de datos IP [?, ?]. En [?] se presentó un diseño y simulación en FPGA de un cuantificador de entropía, sin embargo actualmente no hay disponibles implementaciones en *hardware* de este cuantificador.

Dentro del proyecto mencionado, en este trabajo se implementa un sistema que calcula la entropía para la distribución de probabilidades (*PDF*) asociada a una serie de datos. Se analizan *PDF*'s causales y no causales. Los datos pueden tener un origen digital (generados mediante códigos), o bien provenir del muestreo de señales analógicas. Se utilizó la placa de desarrollo *M1AFS-embedded kit*, basado en el chip *M1AFS1500* que se destaca por tener un bloque analógico embebido en el mismo encapsulado de la FPGA. Luego, se verifica la exactitud numérica del cuantificador implementado comparando sus resultados con un programa patrón. A partir del máximo error detectado se determina la exactitud numérica del sistema.

#### 2.4.1. *Hardware* Implementado

El diseño del *hardware* se basó en el que provee ACTEL en [?], basado en el microcontrolador 8051, interfaces y periféricos. Fue realizado con el paquete de programas *Libero Soc v11.3<sup>®</sup>* de ACTEL. Se utilizó la placa de desarrollo *M1AFS-EMBEDDED-KIT* que contiene una FPGA *M1AFS1500* de ACTEL y periféricos [?]. El chip *M1AFS1500* contiene embebido un bloque analógico que consiste en nueve adaptadores direccionables de cuatro entradas cada uno, un multiplexor analógico de 32 entradas y un conversor analógico-digital configurable.

El sistema implementado puede dividirse en tres etapas principales como se muestra en la Fig. 2.15: una primer etapa de Adquisición de datos, que convierte a palabras digitales las señales del mundo analógico, una Lógica de Cálculo, que se vale de la memoria SRAM para llevar a cabo los cálculos y coordinar las

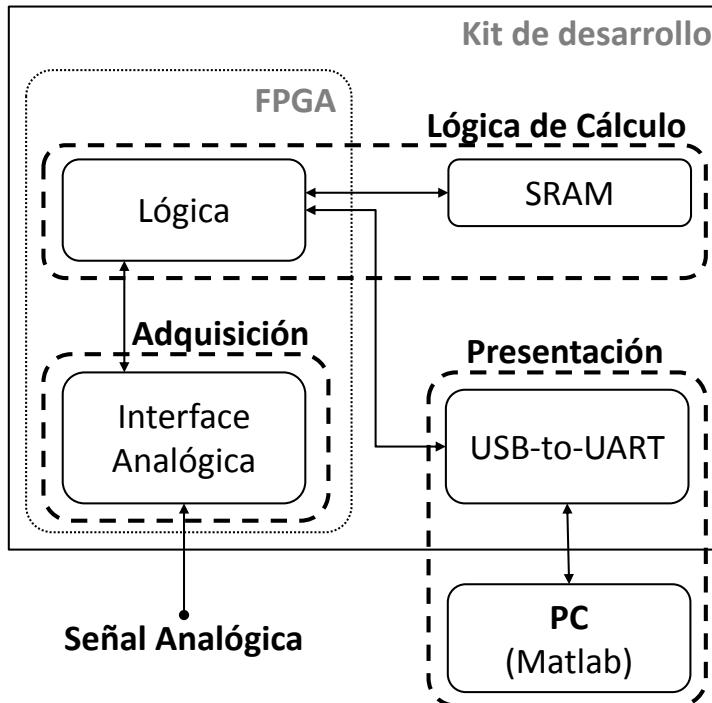


Figura 2.15: Esquema del sistema completo.

interfaces y una etapa de Presentación de resultados, que envía los resultados de la medición a una computadora a través de la interfaz *USB-to-UART*.

#### Etapa de Adquisición

Para ingresar los datos analógicos a ser evaluados utilizamos la entrada de tensión *AV2* del *Analog Quad 2* del bloque analógico. Se encuentra direccionada en el canal siete del multiplexor analógico y fue configurada para un rango de tensiones de entrada de 0 V a 4 V. El conversor analógico-digital se configuró con una resolución de 12 bits. En este primer prototipo la frecuencia de muestreo máxima alcanzada fue de 16 ks/s limitada por el retardo necesario en el procesamiento de la lógica.

#### Lógica de cálculo

En esta etapa se realizan los cálculos y la sincronización entre periféricos. En la Fig. 2.16 pueden verse los bloques principales que la componen.

El núcleo de la implementación es un *Core 8051* que provee ACTEL en su catálogo de librerías. Se trata de un microcontrolador que contiene la lógica principal del microprocesador 8051 de Intel, sin sus periféricos. Este micro tiene una arquitectura Von Newman con un bus de direcciones de 16 bits, lo que limita nuestro diseño a 64 KB de memoria de código y 64 KB de memoria de datos.

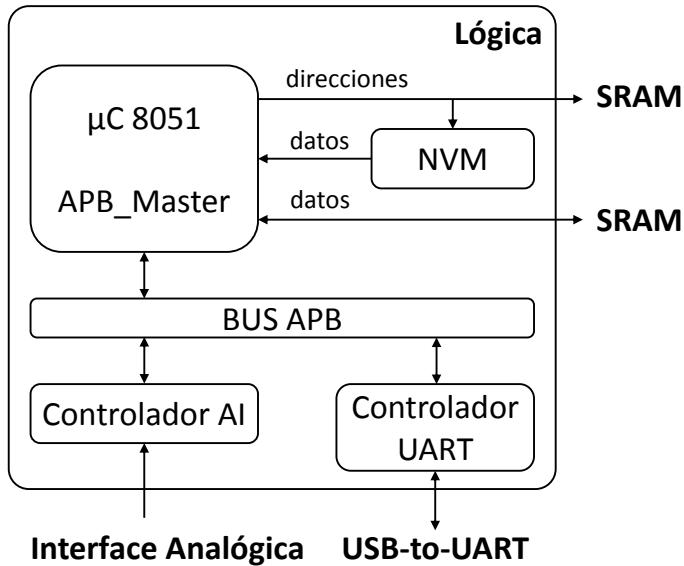


Figura 2.16: Detalle de la lógica de cálculo.

Sobre este microcontrolador corre el programa que realiza los cálculos presentados en la sección ???. Se encarga de, a partir de los datos de entrada, obtener las PDFs (*BP* e *hist*) y de realizar los cálculos para la obtención de las entropías, según la ec. ???. El *software* implementado se describe más detalladamente en la sección 2.4.2.

La memoria de código es una memoria no volátil (NVM) implementada con los bloques flash internos de la FPGA. Ocupa las direcciones desde 0x0000 hasta 0xFFFF y se escribe con el contenido de un archivo en formato hexadecimal durante la compilación.

Las funcionalidades del sistema son ampliadas mediante la conexión de periféricos a través de la interfaz APB.

Para realizar la comunicación con la PC utilizamos el *Controlador UART*. La salida de este bloque es dirigida hacia afuera de la FPGA y se conecta a un chip *USB-to-UART* que se encuentra soldado a la placa del kit de desarrollo.

El bloque analógico es controlado por el *Controlador AI*, que direcciona y sincroniza sus entradas.

### Presentación

La etapa de Presentación de los datos involucra al chip adaptador *USB-to-UART* que se encuentra en la placa de desarrollo y es manejado tanto por el programa que corre en la FPGA como por el *software* que corre sobre la PC. El chip adaptador *USB-to-UART* es el responsable de adaptar la entrada-salida UART de la lógica a una entrada-salida USB estándar mediante la cual es posible interactuar con la PC. Por otra parte el programa que corre en la PC se encarga de la interfaz con el usuario y es descripto en detalle en la siguiente sección.

### 2.4.2. *Software* Implementado

El funcionamiento del sistema se logra mediante la interacción de dos programas. Uno corriendo en la PC y otro en el microcontrolador implementado en la FPGA. Puede verse un diagrama de flujo de ambos programas y la interacción entre ellos en la Fig. 2.17.

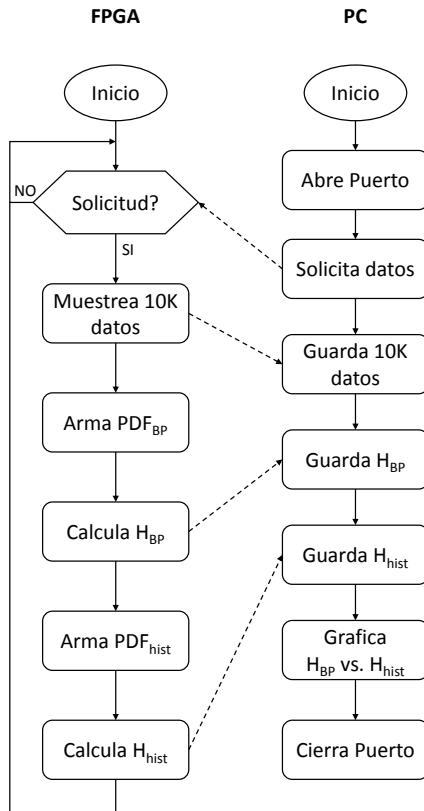


Figura 2.17: Diagrama de flujo del *software* implementado.

En la PC corre un *script* de *Matlab*<sup>©</sup> que se encarga de abrir el puerto serie en donde se encuentra mapeado el USB, solicitar los datos, tomar los resultados del mismo puerto, graficarlos en un plano  $H_{BP}$  vs.  $H_{hist}$  y cerrar el puerto.

Sobre el microcontrolador en la FPGA corre un programa escrito en lenguaje C y compilado para el microcontrolador 8051 utilizando la herramienta *SoftConsole IDE v3.4*<sup>©</sup>. El firmware es una modificación del usado en [?]. Cuando se presenta una solicitud de datos por el puerto UART, se guardan los datos muestreados de la entrada analógica. Luego, se recorre este vector generando las  $PDF_{hist}$  y  $PDF_{BP}$ , a las que se les calcula sus respectivas entropías  $H_{hist}$  y  $H_{BP}$ . Estos resultados son enviados a la PC mediante el mismo puerto.

Con el fin de validar el sistema, el programa en la FPGA envía a *Matlab*<sup>©</sup> el vector de datos muestreados, para que se puedan calcular en la PC sus entropías y compararlas con los resultados del sistema implementado.

| Generador  | Origen    | Error $H_{BP}$ | Error $H_{hist}$ |
|------------|-----------|----------------|------------------|
| Rand       | Digital   | $1,7421E^{-6}$ | $2,6977E^{-6}$   |
| Logístico  | Digital   | $0,4256E^{-6}$ | $94,693E^{-6}$   |
| Triangular | Analógico | $6,3445E^{-6}$ | $2,0028EE^{-6}$  |
| Senoidal   | Analógico | $6,3151E^{-6}$ | $5,6506E^{-6}$   |
| Cuadrada   | Analógico | $0,1797E^{-6}$ | $1,9930EE^{-6}$  |
| Rampa      | Analógico | $245,00E^{-6}$ | $1,0876E^{-6}$   |

Cuadro 2.1: Error de los cuantificadores evaluados en la FPGA con respecto a los resultados calculados por el programa patrón.

### 2.4.3. Resultados

Como se dijo, para testear el sistema se compararon los resultados obtenidos por el sistema implementado y por un programa patrón que corre en la PC. Para esto, se generaron 10 000 muestras de señales con distintas formas de onda tanto externas (analógicas) como internas (digitales).

Las señales digitales fueron generadas por código en el microcontrolador, una corresponde a la función rand() de C y la otra al mapa caótico logístico con parámetro  $r=4$ .

Las señales analógicas fueron generadas con el generador de funciones *HP33120A*. Tienen una amplitud de 4 Vpp y un nivel de continua de 2 V de forma de aprovechar todo el rango del conversor analógico-digital y aumentar la relación señal-ruido. En los cuatro casos la frecuencia de las señales fue de 100 Hz y la velocidad de muestreo de 16 ks/s.

El cuadro 2.1 muestra el error absoluto entre los resultados de los cuantificadores calculados en la FPGA comparados con los resultados calculados con el programa patrón sobre los mismos datos.

La Fig. 2.18 muestra los valores entregados por la FPGA en el plano  $H_{BP}$  vs.  $H_{hist}$ .

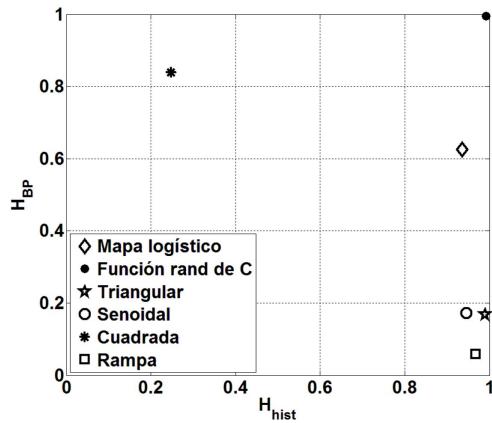


Figura 2.18: Resultados de las mediciones.

Los resultados de la compilación nos permite conocer los recursos de la FPGA utilizados por el sistema completo y la cantidad de memoria ocupada por el *software* que corre en el microcontrolador. Recordemos que esta es una

implementación de *hardware* rígida, es decir primero se arma el circuito en la FPGA (microcontrolador, periféricos, etc.) y luego se carga el *software* sobre él.

El reporte de la compilación de *hardware* devuelto el *Place and Route* se muestra en la Fig. 2.19. Podemos ver que la implementación utiliza un 19 % de los recursos lógicos de la FPGA, el 21 % de las celdas de entrada-salida y el 28 % de los bloques de memoria.

El reporte de la compilación de *software* se muestra en la Fig. 2.20. Podemos ver que la memoria FLASH no volátil se encuentra ocupada al 15,4 %. Por otro lado, de las 65536 direcciones la memoria SRAM tenemos disponibles 61440 dado que parte de esta memoria es utilizada por el bus APB, por lo que se utiliza el 76,7 % de la memoria disponible.

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

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

Figura 2.19: Recursos empleados por el *hardware* del sistema.

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

Figura 2.20: Recursos empleados por el *software* del sistema.

#### 2.4.4. Discusión

El programa debió ser adaptado al microcontrolador instanciado en la FPGA. Estas modificaciones hacen que la salida del sistema implementado no sea igual a la de un programa que corre en la PC, al cual tomamos como programa o patrón. Por esto se testeó el error cometido, para tener una cota y determinar si los resultados de los cuantificadores son correctos. El programa patrón utiliza aritmética de 64 bits en punto flotante norma IEEE754-64 bits y emplea la librería math.h [?]. Para el algoritmo en la FPGA se disminuyó la aritmética a 32 bits de punto flotante norma IEEE754-32 bits. También se requirió el cálculo de la función logaritmo, que se implementó mediante un algoritmo de CORDIC. En el cuadro 2.1 se ve que el error absoluto no supera los  $245E^{-6}$ . Esto indica que se detecta diferencia recién a partir del quinto dígito decimal.

En la Fig. 2.18 puede verse como los cuantificadores  $H_{BP}$  y  $H_{hist}$  diferencian claramente las propiedades estadísticas de las series de datos analizadas. Las señales Senoidal, Rampa y Triangular presentan un valor alto de  $H_{hist}$  porque tienen casi todos los valores que es capaz de generar el conversor Analógico-Digital. Sin embargo, la mezcla de estos datos es mala por tratarse de una señales periódicas totalmente predecibles, esto se ve en el bajo valor de  $H_{BP}$ . Un caso interesante de analizar es la señal Cuadrada. El efecto del ruido aditivo es especialmente notable en las zonas en donde el valor de la señal debería ser constante. Se generan dos Gaussianas muy finas en torno a los valores ideales en

la  $PDF_{hist}$ , esto no afecta demasiado el valor calculado  $H_{hist}$ , sin embargo para la  $PDF_{BP}$ , se calcula el patrón de orden directamente a la señal ruidosa, por lo que el valor de  $H_{BP}$  es más alto que el esperado. La señal generada mediante la función rand de C, presenta las mejores propiedades estadísticas ubicándose en el punto  $\sim (1, 1)$ .

#### 2.4.5. Conclusiones y trabajo futuro

Se desarrolló e implementó un sistema que permite medir con buena precisión las entropías causal y no-causal de señales analógicas provenientes del exterior de la FPGA y también internas generadas por código.

Se logró medir señales y realizar cálculos complejos con un microcontrolador modesto como el 8051 instanciado en la FPGA AFS1500 de ACTEL. Este primer prototipo cumple con las especificaciones de precisión y cantidad de recursos requeridos establecidas en el diseño, el próximo paso será optimizar el sistema en cuanto a frecuencia de operación e inmunidad al ruido.

Se prevé que el sistema permita modificar, en tiempo de ejecución, la frecuencia de muestreo, de forma de que sea adaptable a la señal de entrada, con el límite superior de 500 Ks/s fijado por el ADC.

Deberá agregarse también un umbral a partir del cual un valor es considerado distinto de otro, de esta forma se solucionaría el problema que presenta el ruido aditivo en el cálculo de  $H_{BP}$ .

El código de este sistema ocupa el 15,4 % del total de la memoria flash del micro instanciado, por lo que será posible agregar *software* para implementar otros cuantificadores y funcionalidades. En cuanto a los recursos disponibles en la FPGA se utilizaron 7349 celdas lógicas, quedando casi el 80 % de los recursos de *hardware* disponibles para implementar los sistemas bajo prueba en forma concurrente.

### 2.5. Dinámica de los ITQ's con AWGN y banda limitada

En esta sección exploramos la respuesta de un sistema de medición de entropías en presencia de ruido aditivo y señales filtradas. Esta inquietud surge como resultado de la implementación detallada en la sección 2.4. El filtrado es inherente al ancho de banda del sistema de medición y las señales a medir siempre están contaminadas con ruido, por lo tanto es necesario caracterizar la respuesta de nuestro sistema de medición ante estos dos procesos. Este trabajo es complementario al desarrollo de un sistema de medición de entropías implementado en FPGA.

#### 2.5.1. Filtrado digital

Ya sea en la elección de un filtro como en cualquier problema de diseño en ingeniería, generalmente no es posible dar una respuesta posible acerca de cual es al mejor solución. Se discute la posibilidad de la implementación de distintos filtros porque no hay un solo método de diseño ni un solo tipo de filtro mejor para todas las circunstancias. La elección del tipo de filtro depende de la importancia de sus ventajas aplicadas a cada problema.

Un filtro ideal es aquel en el que la respuesta en frecuencia es unitaria en el rango de las frecuencias de paso, cero en la banda de rechazo y no posee banda de transición. Dada la inherente periodicidad de la respuesta en frecuencia para tiempo discreto esta tiene la apariencia de un tren rectangular en frecuencias, sin embargo en este trabajo solo se muestra la frecuencia normalizada en el intervalo  $(0; 1)$ . Entonces la transferencia de un pasabajos ideal en frecuencia normalizada quedaría:

$$H_{LP} = \begin{cases} 1, & |f - 0,5| > f_c \\ 0, & |f - 0,5| < f_c \end{cases} \quad (2.18)$$

Ecuación definida en el intervalo de frecuencias normalizadas  $f \in (0, 1)$ .

El hecho de que no podemos contar con series de valores infinitamente largas para ser filtradas, equivale a decir que disponemos de una serie de muestras enventanada. Como el producto en el dominio del tiempo equivale a una convolución en el dominio de la frecuencia, podemos estudiar el efecto que este enventanado tiene sobre la respuesta frecuencial del filtro. Consideremos la ventana mas sencilla; la ventana rectangular. Supongamos que la aplicamos sobre una versión retardada de la respuesta ideal, su efecto en el dominio de la frecuencia será la convolución entre la respuesta de nuestro filtro ideal y la transformada esta ventana rectangular, es decir una función *sinc* de período  $1/N$  en donde  $N$  es la cantidad de muestras que entran en la ventana.

El efecto de enventanado o truncamiento de la respuesta es doble: por una parte, la anchura del lóbulo principal está relacionada con la aparición de una banda de transición en el filtro. Por otra, la presencia de lóbulos laterales (secundarios) lleva a la aparición de un ripple u oscilaciones en la respuesta en frecuencia, en ambas bandas, (más apreciable en la banda no pasante). La aparición de los lóbulos secundarios se debe a que la ventana rectangular presenta una discontinuidad abrupta que, al pasar al dominio de la frecuencia, conlleva un reparto de la energía por todo el espectro a causa del aliasing.

Una opción que se plantea es generalizar el concepto de ventana y emplear ventanas más suaves que la rectangular para realizar el truncamiento de la respuesta deseada, esta técnica es una de las formas de realizar un filtro FIR. Sin embargo, si analizamos la transformada de la ventana cuadrada vemos que presenta valores nulos cada  $1/N$ , que son los mismos lugares en donde aparecen las componentes espectrales de la DFT. Esto significa que los efectos de la ventana rectangular aparecen al convertir la respuesta de este filtro a tiempo continuo.

Otra opción sería diseñar un filtro analógico y transformar su respuesta a frecuencia discreta. Para esto contamos con fórmulas cerradas de diseño, por lo que podemos satisfacer cualquier especificación preestablecida. La utilización de esta técnica da como resultado un filtro IIR. Comparado con un FIR, un filtro IIR requiere un orden mucho menor para cumplir las especificaciones de diseño.

Aquí analizamos un sistema en el cual la respuesta es analizada en el dominio digital. Además, es necesario filtrar componentes espectrales de a una, lo que requiere una banda de transición muy estrecha, esto reduce el conjunto de filtros posibles. Por el lado del IIR probamos un filtro elíptico, este filtro presenta una banda de transición muy estrecha en sacrificio de un ripple que aparece tanto en la banda de paso como en la de rechazo. Por el lado del FIR probamos un filtro ideal con una ventana rectangular que abarca toda la serie de valores, en

la sección 4.2.3 se detalla como fue implementado.

### 2.5.2. Resultados

Para representar la dinámica en función del filtrado, se eligieron dos señales representativas (cuadrada y senoidal) y se les calcularon los cuantificadores descritos en la sección ?? luego de ser filtrados por los filtros elegidos en la sección 2.5.1. Por otro lado, se calculan los mismos cuantificadores a una señal de ruido blanco gaussiano.

En la figura 2.21 se muestra el procedimiento utilizado. Primero se generó un vector de ruido blanco gaussiano de  $N = 50E3$  muestras, la desviación estándar  $\sigma$  es variable y se logra multiplicando al vector inicial de  $\sigma = 1$  por la desviación estándar elegida. Luego se genera la señal determinística de  $N = 50E3$  muestras, período  $T = 100$  muestras y amplitud unitaria, que se sumó el ruido para lograr la señal contaminada. La señal resultante se filtra para luego calcular cuantificadores. Como se explicó más arriba, para calcular la entropía de valores se genera el histograma de valores y se lo normaliza para calcular la Función Densidad de Probabilidad de valores  $PDF_{hist}$  a la que se le calcula la entropía de Shannon normalizada que da como resultado la entropía de valores normalizada  $H_{hist}$ . Para calcular la entropía de patrones de orden se utiliza el histograma de patrones de orden que cuando se normaliza se consigue la función densidad de probabilidad de patrones de orden  $PDF_{BP}$ , a la que se le calcula la entropía de Shannon normalizada para conseguir la entropía de patrones de orden  $H_{BP}$ .

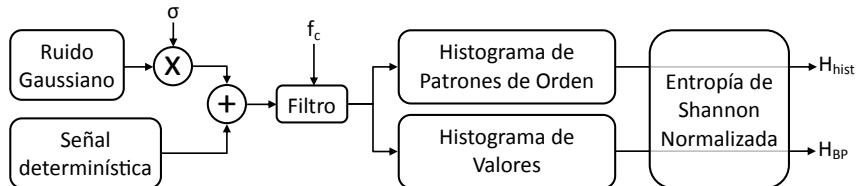


Figura 2.21: Diagrama de flujo del experimento.

Para evaluar la contribución de cada componente espectral a las entropías, se evaluaron dos filtros. Primero se aplicó un filtro elíptico de orden 10 con ripple pasabanda de  $0,5dB$ , ripple en la banda de rechazo de  $100dB$  y frecuencia de corte variable  $f_c$ , en la figura 2.22 se muestra su respuesta en ganancia (fig. 2.22b) y fase (fig. 2.22c) para el caso de  $f_c = 0,5$ . De esta forma se logra un filtrado lo suficientemente abrupto como para considerar que a medida que se barren distintas frecuencias de corte se eliminan componentes espectrales individualmente. Los resultados de este filtrado se compararon con los resultados de un filtro ideal (fig. 2.23), que consiste en una máscara aplicada a la transformada de fourier de la señal a filtrar, de esta manera se consigue el espectro de la señal filtrada, el cual es antitransformado para recuperar la versión filtrada en las muestras. El diagrama de este filtro puede verse en la figura 2.23a. Este procedimiento equivale a un filtrado ideal sin retardo, por lo que el bode de amplitud es  $0dB$  en la banda de paso y  $-\infty dB$  en la banda de rechazo (fig. 2.23a); la fase  $\omega\tau = 0$  es lineal con pendiente nula (fig. 2.23c).

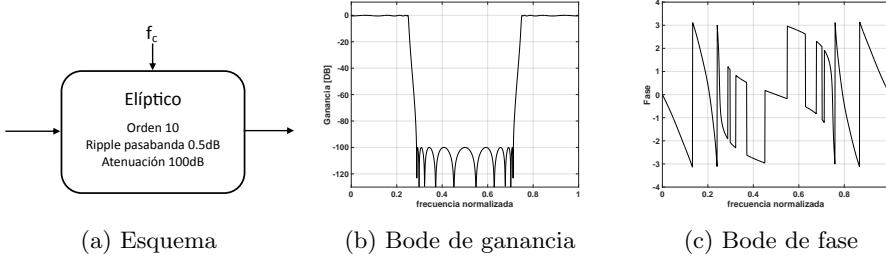


Figura 2.22: Filtro elíptico.

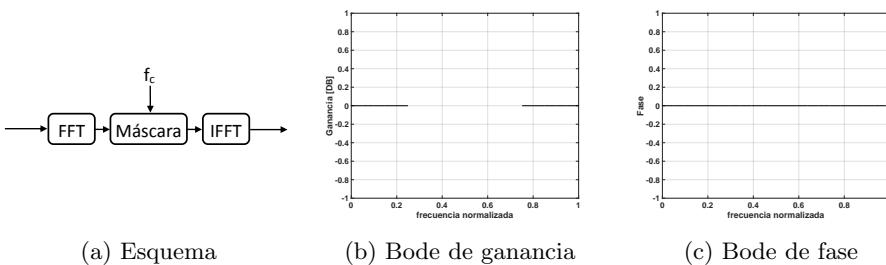
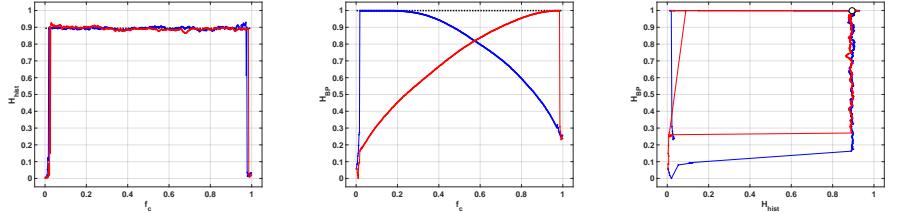


Figura 2.23: Filtro ideal.

Primero se aplicó una señal de ruido blanco gaussiano, es decir que la señal determinística es cero y la desviación estándar de la gaussiana unitaria. En la figura 2.24 se muestra el resultado de los cuantificadores a medida que se va barriendo la frecuencia de corte del filtro elíptico. En la figura 2.24a se muestra la entropía del histograma de valores  $H_{hist}$ , puede verse que su valor se mantiene constante alrededor de 0,9 tanto para el filtro pasa-bajos (roja) como el pasa-altos (azul), este valor es el mismo que resulta de calcular la entropía del histograma de valores a la señal sin filtrar (resultado que se muestra con una línea punteada negra en el mismo plot). También puede verse que cuando la frecuencia de corte del filtro elíptico se acerca a los extremos el valor del cuantificador cae, en estas frecuencias el método numérico que calcula el vector filtrado diverge debido a la precisión finita. En la figura 2.24b se muestra la entropía de los patrones de orden,  $H_{BP}$  se mantiene en valores bajos cuando el filtro (pasa-altos en azul y pasa-bajos en rojo) deja pasar pocas componentes espectrales. Luego, a medida que la frecuencia de corte deja pasar más componentes espectrales, el cuantificador tiende a 1, que es justamente el valor que arroja cuando se ingresa con la señal sin filtrar (este valor se marca con una línea punteada negra). El cuantificador detecta los cambios en la forma de la señal a medida que es filtrada. Por último, en el plano  $H_{hist} - H_{BP}$  de la figura 2.24c se compacta la información de ambos cuantificadores, aunque se pierde la noción de la frecuencia de corte.

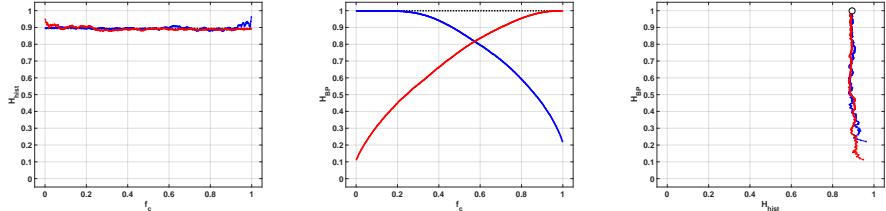
En la figura 2.25 se muestran los resultados del mismo procedimiento pero cuando se aplica un filtro ideal. El comportamiento de los cuantificadores es igual al del filtro elíptico en todos los casos con la diferencia que el método no diverge cuando  $f_c \rightarrow 1$  o  $f_c \rightarrow 0$ . Pueden verse por lo tanto los valores que arrojan los cuantificadores en los extremos de la frecuencia de corte. La entropía no causal



(a) Entropía de valores normalizada      (b) Entropía de patrones de orden normalizada      (c) Plano doble entropía

Figura 2.24: Cuantificadores calculados sobre la salida del filtro elíptico cuando se ingresa con ruido blanco gaussiano.

de la figura 2.25a aumenta levemente en los extremos, en donde el histograma de valores deja de tener una distribución gaussiana y se aplana levemente. También puede verse en 2.25b que la entropía de valores  $H_{BP} \rightarrow 0,15$  cuando  $f_c \rightarrow 0$  para el pasa-bajos (rojo) y para el pasa-altos ( $f_c \rightarrow 1$ ) para el pasa-altos (azul)  $H_{BP} \rightarrow 0,22$ . En este caso es fácil comparar la sensibilidad al filtrado de ambos cuantificadores, en el plano doble entropía de la figura 2.25c. El círculo blanco muestra la posición en este plano cuando ningún filtro es aplicado, podemos ver que el apartamiento en el eje vertical aumenta a medida que la serie es filtrada, mientras que no se aparta en el sentido horizontal. Esto muestra que la sensibilidad al filtrado de  $H_{BP}$  es mucho mayor que la de  $H_{hist}$ .



(a) Entropía de valores normalizada      (b) Entropía de patrones de orden normalizada      (c) Plano doble entropía

Figura 2.25: Cuantificadores calculados sobre la salida del filtro ideal cuando se ingresa con ruido blanco gaussiano.

Para el sistema planteado no se necesita volver al dominio continuo analógico, por lo que las dificultades mencionadas en la sección 2.5.1 respecto al filtrado ideal (como ripple en las bandas de paso y rechazo) no aplican a este caso. Por este motivo para esta serie de pruebas elegimos el filtro ideal, dado que presenta mejores resultados que el elíptico.

La primer señal determinística que se muestra es una senoidal de amplitud unitaria con período de 100 muestras, los resultados pueden verse en la figura 2.26. Mientras la única componente espectral no es filtrada, el valor de la entropía de valores es  $H_{hist} \approx 0,57$  en la figura 2.26a y la entropía de patrones de orden  $H_{BP} \approx 0,16$  en la figura 2.26b. Ambos cuantificadores caen a cero cuando la única componente espectral es filtrada, ya sea por el filtro pasa-bajos (azul) o por el pasa-altos (rojo). El plano doble entropía muestra un punto en

(0, 57; 0, 16) para la senoidal sin filtrar y otro en (0; 0) cuando la única componente espectral es filtrada.

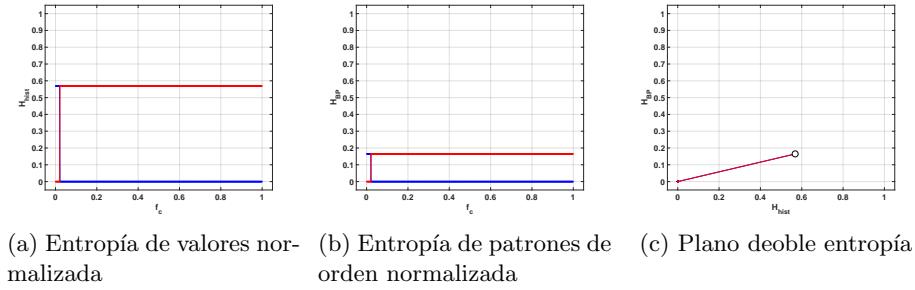


Figura 2.26: Cuantificadores calculados sobre la salida del filtro ideal cuando se ingresa con una senoidal limpia.

La salida de los cuantificadores cuando esta señal es contaminada con ruido gaussiano aditivo con  $\sigma = 0,2$  puede verse en la figura 2.27. Vemos en la figura 2.27a que la entropía de valores aumenta cuando el filtrado no elimina la componente espectral, dando valores incluso sobre el valor de la entropía de la gaussiana. Esto se debe a que la PDF de la senoidal es complementaria con la de la gaussiana, entonces la PDF de la resultante es más parecida a la del ruido uniforme. Para los patrones de orden de la figura 2.27b, el pasa-altos no deja ver un cambio significativo debido a que la componente espectral de la senoidal es eliminada en la zona en la que su entropía es alta. El pasa-bajos en cambio muestra que mientras esta componente está presente el valor de la entropía es asintótico a  $H_{BP} \rightarrow 0,16$  a medida que la frecuencia de corte baja. Recordemos que  $H_{BP} \approx 0,16$  es el valor de la entropía de patrones de orden de la senoidal limpia. En el plano doble entropía (figura 2.27c) se ve que ambos cuantificadores son complementarios, en el sentido que la entropía de valores detecta la presencia o no de la señal determinística mientras que la entropía de patrones de orden detecta el filtrado sobre la señal de ruido.

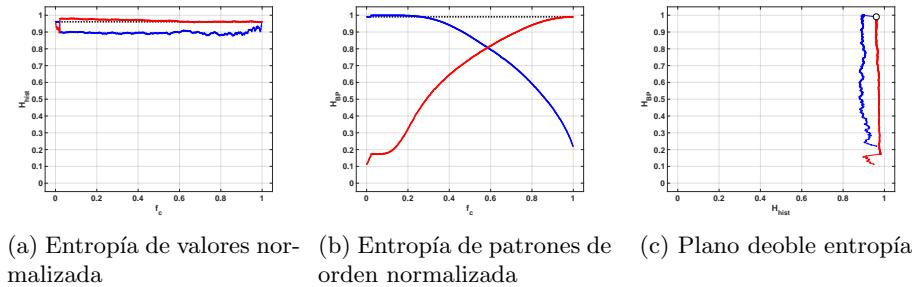
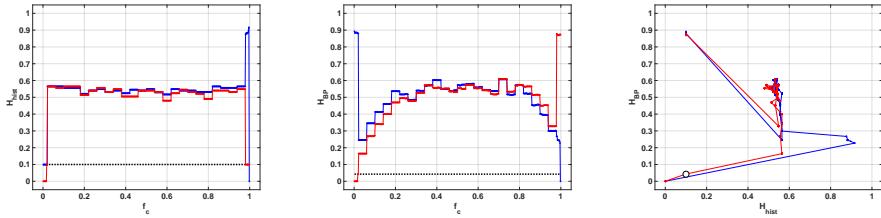


Figura 2.27: Cuantificadores calculados sobre la salida del filtro ideal cuando se ingresa con una senoidal ruidosa.

En la figura 2.28 se muestran los resultados cuando la señal determinística es una cuadrada sin ruido de amplitud unitaria y período de 100 muestras. Tanto la entropía de valores  $H_{hist}$  como la entropía de patrones de orden  $H_{BP}$  presentan

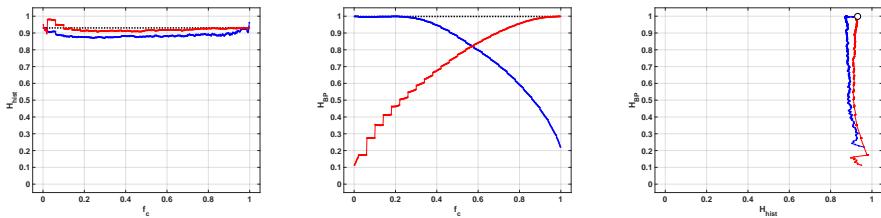
una forma escalonada, sus valores se mantienen constantes a medida que se barre la frecuencia de corte de los filtros hasta que la siguiente componente espectral es filtrada. También se ve que en ambos casos los valores resultantes se mantienen bastante lejos del valor sin filtrar, que se muestra con una linea negra punteada.



(a) Entropía de valores normalizados  
(b) Entropía de patrones de orden normalizada  
(c) Plano doble entropía

Figura 2.28: Cuantificadores calculados sobre la salida del filtro ideal cuando se ingresa con una cuadrada limpia.

El caso contaminado con ruido (figura 2.29) cambia respecto del caso sin contaminar. En la figura 2.29a se ve que para el pasabajos (rojo)  $H_{hist}$  se mantiene alrededor del valor sin filtrar (linea punteada), excepto con las tres frecuencias más bajas, en donde su valor aumenta un levemente por las mismas razones que aumentaba con la senoidal contaminada. Algo parecido sucede con  $H_{BP}$  en la figura 2.29b. Cuando la cuadrada se contamina con ruido su valor se mantiene cercano al del ruido gaussiano, esto es por que en las regiones en las que la cuadrada es plana su contribución al patrón de orden es nula. Para el pasa bajos se ve un escalonado en la posición de cada componente espectral que se hace más notorio para las frecuencias más bajas, en donde la contribución del ruido ya es bastante baja y a la vez se encuentran las componentes espectrales de mayor peso. Esto no es tan notorio en el pasa-altos, en este caso cuando la contribución del ruido es de baja amplitud también lo es la de la determinística, enmascarando este fenómeno.



(a) Entropía de valores normalizados  
(b) Entropía de patrones de orden normalizada  
(c) Plano doble entropía

Figura 2.29: Cuantificadores calculados sobre la salida del filtro ideal cuando se ingresa con una cuadrada ruidosa.

Para caracterizar el comportamiento de los cuantificadores frente a la amplitud de ruido, se generaron cuadradas contaminadas con AWGN de dos amplitudes y se filtraron para calcular cuantificadores. En la figura 2.30 se muestran

ambos cuantificadores cuando se hace variar el ruido con valores de la desviación estándar  $\sigma = [0 \ 0,1 \ 1]$ . Cuando comparamos las figuras 2.30a, 2.30b y 2.30c vemos un cambio significativo cuando pasamos de la señal limpia de 2.30a a la contaminada con bajos niveles de ruido de la 2.30b, sin embargo cuando pasamos del bajo nivel de ruido de 2.30b al de la figura 2.30c el cambio es mucho más sutil. De modo similar, entre las figuras 2.30d y 2.30e hay muy poco parecido, mientras que las figuras 2.30e y 2.30f son bastante similares. En este segundo caso es más evidente la diferencia cuando cambia el nivel de ruido, con bajos niveles puede verse el escalonado que aparece cada vez que una frecuencia es filtrada, mientras que cuando la amplitud de ruido es mayor este escalonado aparece solo en el pasabajo para las tres primeras frecuencias, que resultan ser las de mayor peso.

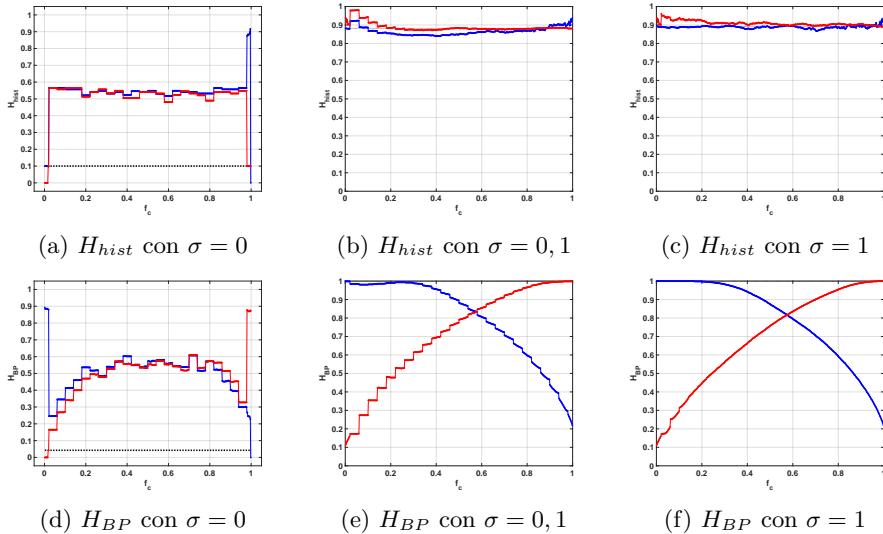


Figura 2.30: Cuantificadores calculados sobre la salida del filtro ideal cuando se ingresa con cuadradas contaminadas con AWGN de con amplitudes de ruido  $\sigma = [0 \ 0,1 \ 1]$ .

### 2.5.3. Discusión

Este trabajo fue necesario para explorar las fuentes de error en un medidor de entropías implementado en FPGA.

Para este primer análisis evaluamos que sucede aplicando un filtro abrupto, es por esto que elegimos para comparar un filtro elíptico y uno ideal. Las respuestas del filtro elíptico y del ideal fueron muy similares en el rango de frecuencias en los que el elíptico tiene un buen comportamiento, sin embargo cuando la frecuencia de corte del elíptico se acerca a los extremos (es decir cuando  $f_c \rightarrow 0$  o  $f_c \rightarrow 1$ ) la salida del filtro diverge. El problema se debe a que el método numérico utilizado para calcular la salida del filtro diverge por la precisión finita utilizada. Como no necesitamos volver a la frecuencia continua nos quedamos con los resultados del ideal para hacer las pruebas, sin tener que preocuparnos por el ripple que aparece en las bandas de paso y rechazo cuando

pasamos al mundo analógico.

Cuando comparamos las respuestas de los cuantificadores con y sin ruido, vemos que las señales limpias tienen mesetas, es decir que se mantienen constantes hasta que el filtrado elimina la siguiente componente espectral. Sin embargo, cuando son contaminadas con ruido los cuantificadores cambian para parecerse más a los resultados que arroja el ruido blanco gaussiano sin ninguna señal determinística. En todos los casos se vio que estos cuantificadores son muy sensibles a la presencia de ruido, los que nos permite vincular a este hecho los errores en la medición.

También vimos que los valores cambian a medida que se filtra la señal sin contaminar, lo que agrega una segunda fuente de error dada por el ancho de banda finito del sistema.

Para continuar con este proyecto faltaría, por un lado caracterizar el sistema de medición en cuanto a su ancho de banda y su rechazo al ruido aditivo, y por otro lado probar con otros cuantificadores (como complejidad, desequilibrio, entropía diferencial, rate entropy, etc) o con variantes de los presentados aquí (Bandt & Pompe pesada, amplitud promedio en el emmbedding, etc).

# Capítulo 3

## Sistemas caóticos

### 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 convexión 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 categorías:

- 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.

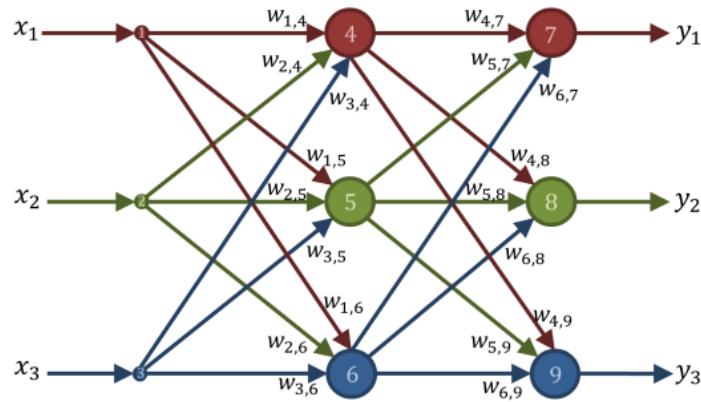
El perceptrón multicapa es una RNA formada por capas de neuronas. Las neuronas pueden pertenecer a la capa de entrada, capas ocultas o capa de salida. Estas neuronas no incorporan memoria por lo que su salida depende del estado de sus entradas en el instante actual (no tienen retardo), además, como el 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 saltándose capas. En la figura 3.1 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 asociativa; 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 asociativa 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:

- Su modelo incluye realimentaciones, que son básicas en su modo de funcionamiento.



|             |                                       |                                       |                     |
|-------------|---------------------------------------|---------------------------------------|---------------------|
| $0 \ 0 \ 0$ | $w_{1,4} \quad w_{1,5} \quad w_{1,6}$ | $0 \ 0 \ 0$                           | Vienen de la capa 1 |
| $0 \ 0 \ 0$ | $w_{2,4} \quad w_{2,5} \quad w_{2,6}$ | $0 \ 0 \ 0$                           |                     |
| $0 \ 0 \ 0$ | $w_{3,4} \quad w_{3,5} \quad w_{3,6}$ | $0 \ 0 \ 0$                           |                     |
| $0 \ 0 \ 0$ | $0 \ 0 \ 0$                           | $w_{4,7} \quad w_{4,8} \quad w_{4,9}$ | Vienen de la capa 2 |
| $0 \ 0 \ 0$ | $0 \ 0 \ 0$                           | $w_{5,7} \quad w_{5,8} \quad w_{5,9}$ |                     |
| $0 \ 0 \ 0$ | $0 \ 0 \ 0$                           | $w_{6,7} \quad w_{6,8} \quad w_{6,9}$ |                     |
| $0 \ 0 \ 0$ | $0 \ 0 \ 0$                           | $0 \ 0 \ 0$                           | Vienen de la capa 3 |
| $0 \ 0 \ 0$ | $0 \ 0 \ 0$                           | $0 \ 0 \ 0$                           |                     |
| $0 \ 0 \ 0$ | $0 \ 0 \ 0$                           | $0 \ 0 \ 0$                           |                     |
| $0 \ 0 \ 0$ | $0 \ 0 \ 0$                           | $0 \ 0 \ 0$                           |                     |

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.

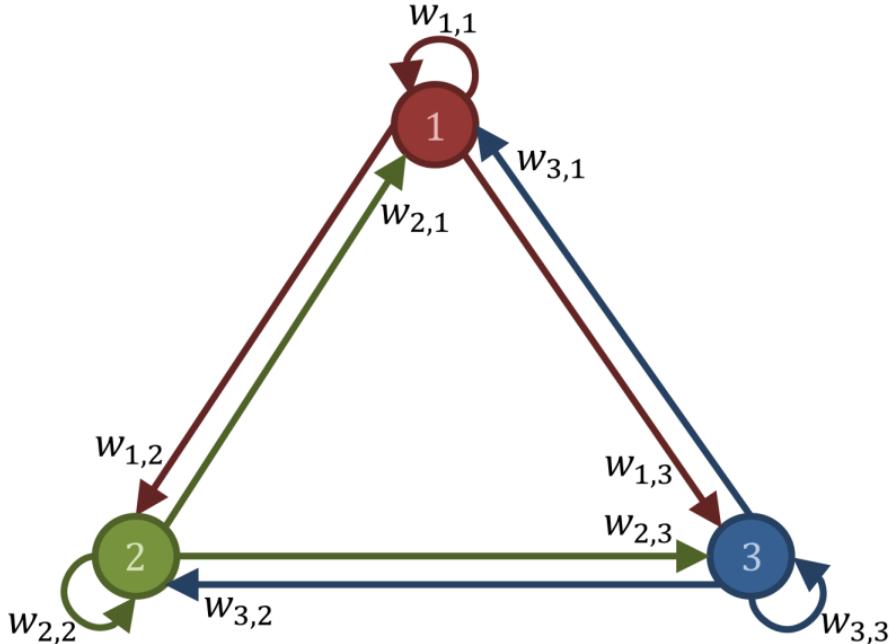


Figura 3.2: Red de Hopfield. Ahora, la matriz de pesos tiene todos sus valores permitidos.

- 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.

### 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 3.2. 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 descrito 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 3.3 se muestra nuestra nueva red neuronal en tiempo discreto. Sus coeficientes dependen del paso de tiempo. Este sistema se aproxima 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 3.4 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 frente 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 3.5.

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 3.6 puede verse este efecto.

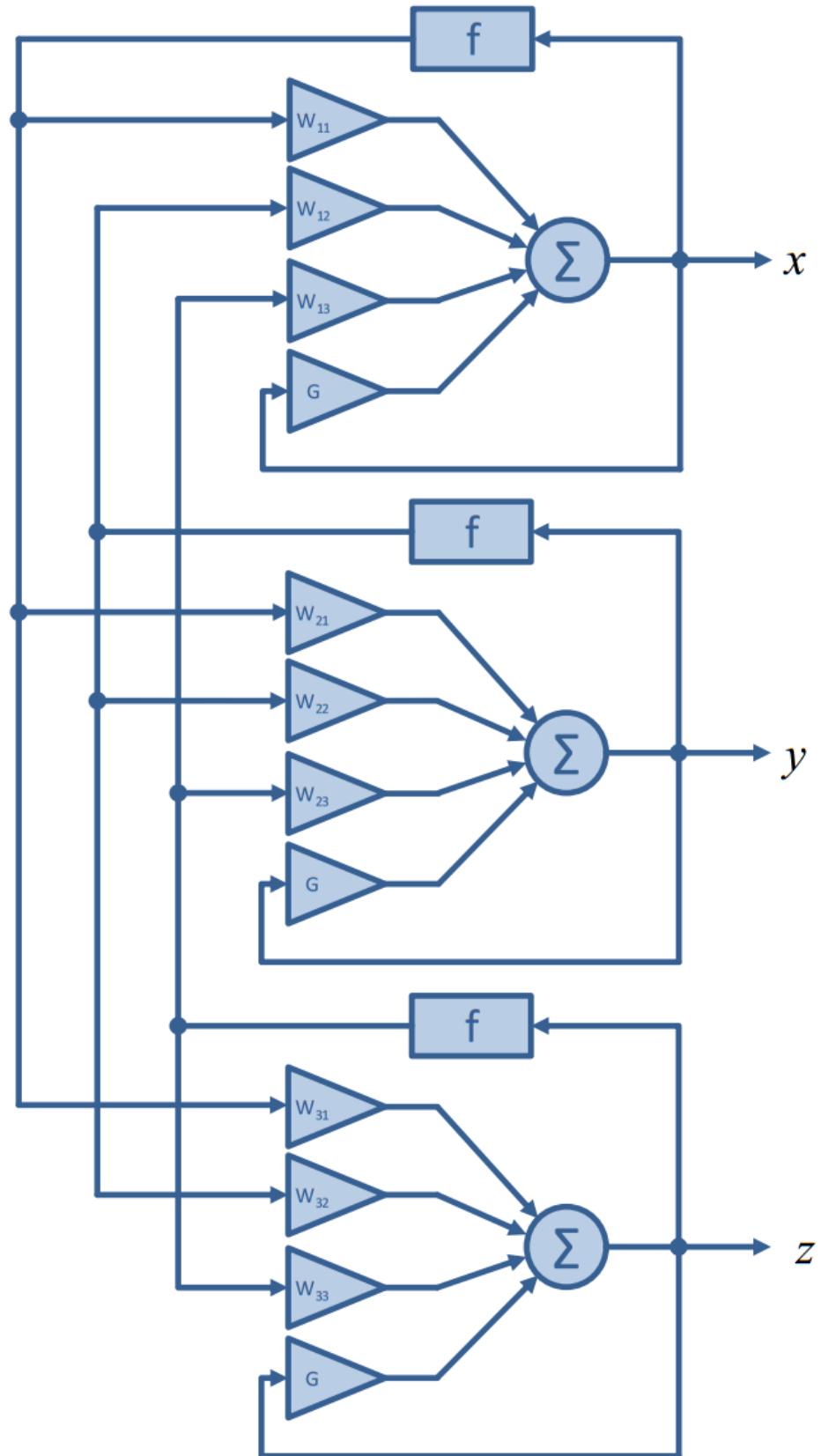


Figura 3.3: Red utilizada. Se trata de una red de Hopfield tridimensional de memoria diferencial, el diseño está orientado a una posterior implementación en FPGA.

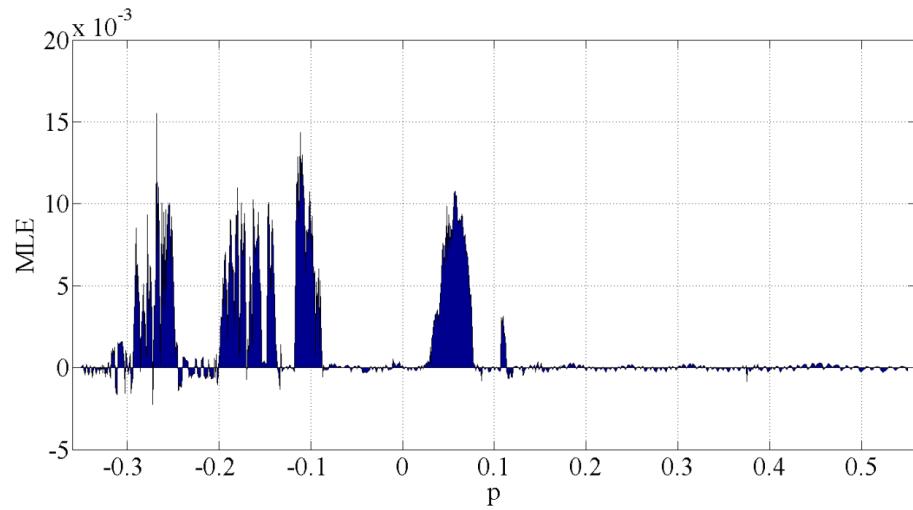


Figura 3.4: Exponente de Lyapunov en función del parámetro  $p$ . Existe caos en toda la zona en la que es positiva.

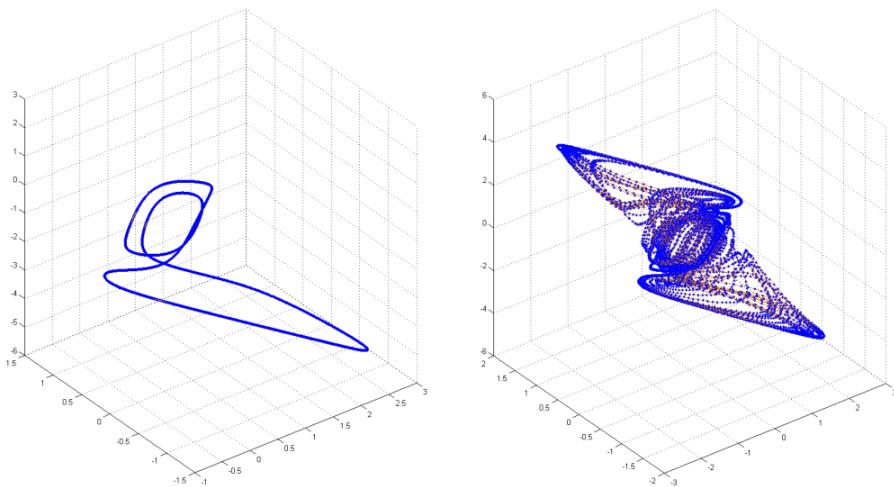


Figura 3.5: Dos trayectorias características del sistema en el espacio de fases. La trayectoria de la izquierda se corresponde con un  $MLE < 0$  y la positiva con un  $MLE > 0$ .

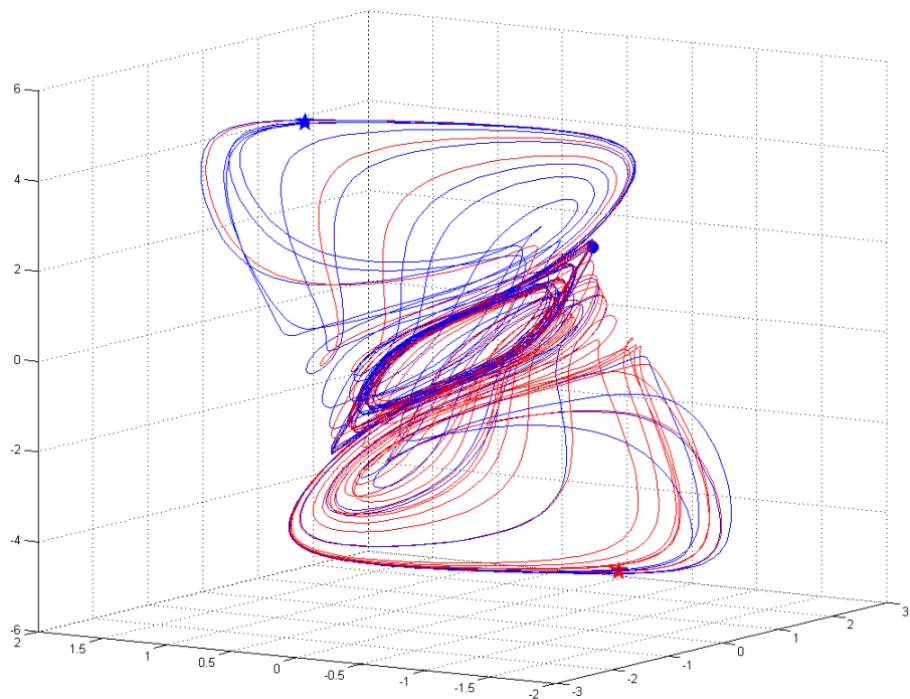


Figura 3.6: Dos trayectorias de la red de Hopfield para condiciones iniciales próximas. Las condiciones iniciales están marcadas con dos puntos grandes cerca del centro del atractor y los valores después de  $\Delta t = 3s$  con estrellas en ambos extremos de la figura.



## 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.

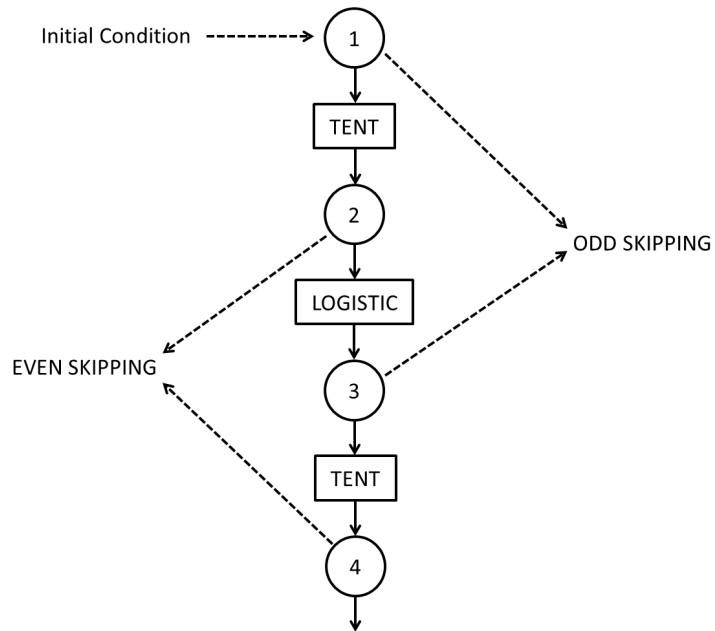


Figura 4.1: ZONA CH REHACER

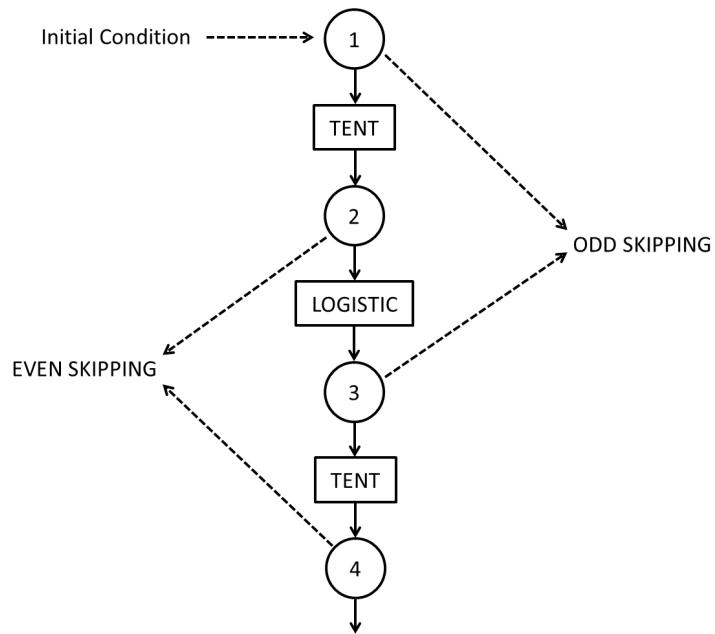


Figura 4.2: Sequential switching between Tent and Logistic maps. In the figure are also shown even and odd skipping strategies

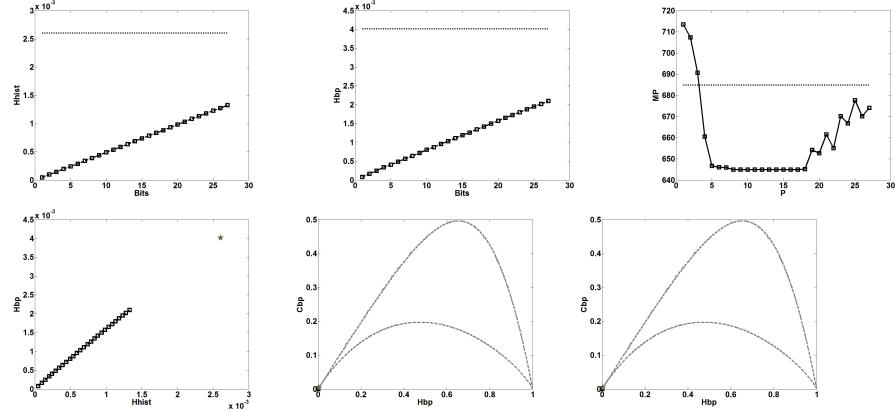


Figura 4.3: Statistical properties of the Tent map using different numerical representations. Figures (a) to (e) correspond to decimal representation: (a)  $H_{hist}$  vs  $P$  (b)  $H_{BP}$  vs  $P$  (c) Number of missing ordering patterns  $MP$  vs  $P$ . In Figures (a) to (c) dashed line correspond 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.

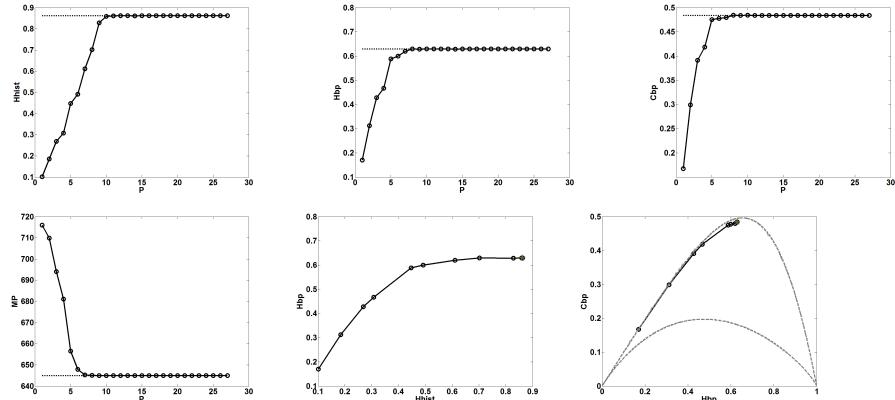


Figura 4.4: Statistical properties of the LOG map using different numerical representations. Figures (a) to (d) correspond to decimal representation: (a)  $H_{hist}$  vs  $P$  (b)  $H_{BP}$  vs  $P$  (c)  $C_{BP}$  vs  $P$  (d) Number of missing ordering patterns  $MP$  vs  $P$ . In Figures (a) to (d) 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.

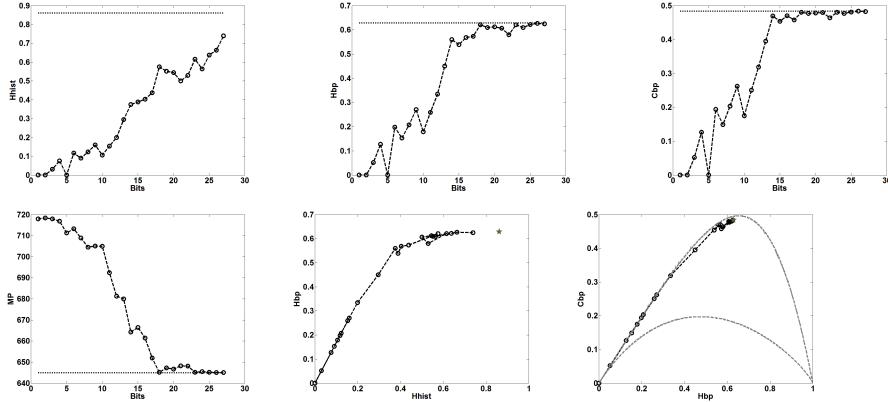


Figura 4.5: Statistical properties of the LOG map using binary representation: (a)  $H_{hist}$  vs  $P$  (b)  $H_{BP}$  vs  $P$  (c)  $C_{BP}$  vs  $P$  (d) Number of missing ordering patterns  $MP$  vs  $P$ . In Figures (a) to (d) dashed line correspond to floating point numbers. (e) representation 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.

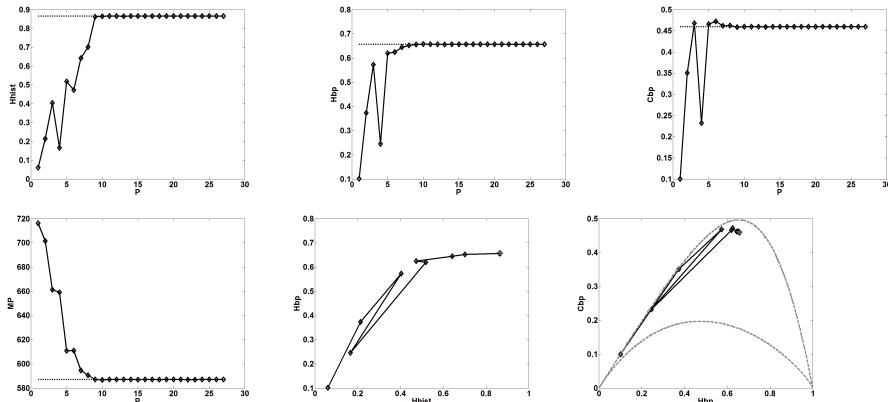


Figura 4.6: Statistical properties of the SWITCH map using decimal representation: (a)  $H_{hist}$  vs  $P$  (b)  $H_{BP}$  vs  $P$  (c)  $C_{BP}$  vs  $P$  (d) Number of missing ordering patterns  $MP$  vs  $P$ . In Figures (a) to (d) dashed line correspond to floating point numbers. (e) representation 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).

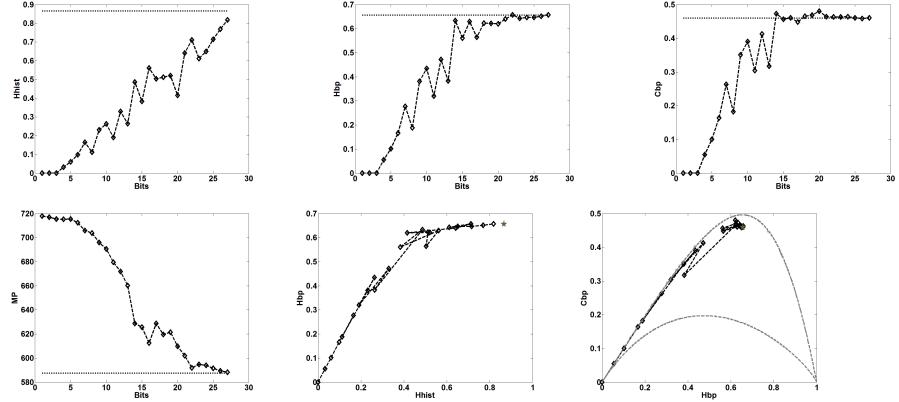


Figura 4.7: Statistical properties of the SWITCH map using binary representation: (a)  $H_{hist}$  vs  $P$  (b)  $H_{BP}$  vs  $P$  (c)  $C_{BP}$  vs  $P$  (d) Number of missing ordering patterns  $MP$  vs  $P$ . In Figures (a) to (d) dashed line correspond to floating point numbers. (e) representation 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).

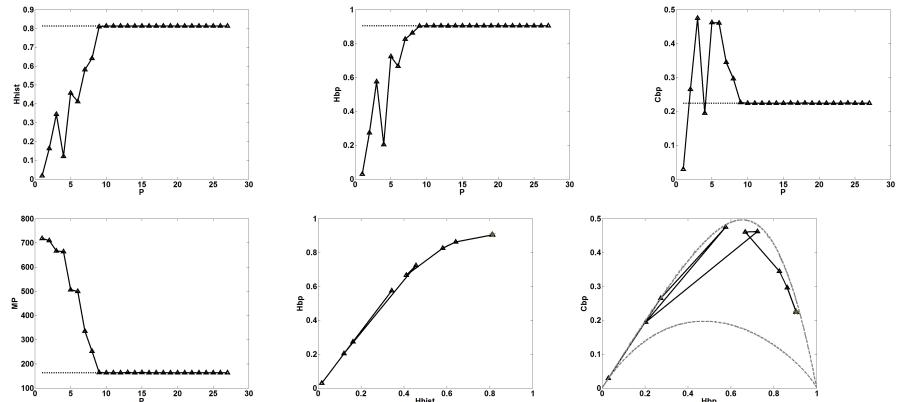


Figura 4.8: Statistical properties of EVEN, obtained by skipping the values in the odd position of the time series of SWITCH, using decimal representation: (a)  $H_{hist}$  vs  $P$  (b)  $H_{BP}$  vs  $P$  (c)  $C_{BP}$  vs  $P$  (d) Number of missing ordering patterns  $MP$  vs  $P$ . In Figures (a) 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.

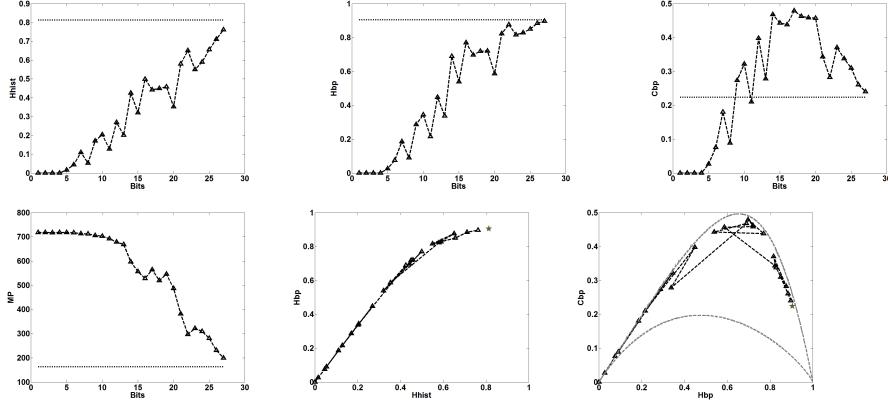


Figura 4.9: Statistical properties of EVEN, obtained by skipping the values in the odd position of the time series of SWITCH, using binary representation: (a)  $H_{hist}$  vs  $P$  (b)  $H_{BP}$  vs  $P$  (c)  $C_{BP}$  vs  $P$  (d) Number of missing ordering patterns  $MP$  vs  $P$ . In Figures (a) 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.

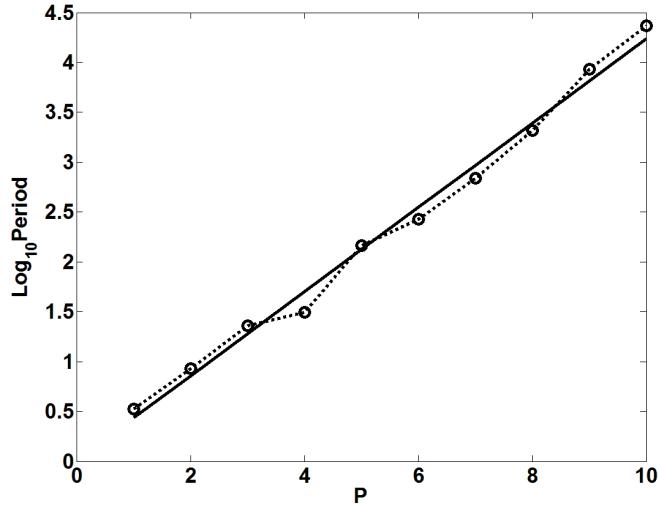


Figura 4.10: Period  $T$  as a function of de number of decimal digits  $P$  for the LOG map.



## Capítulo 5

# Generadores de TRNG usando ROs en FPGA

### 5.1. Introduction

El *Jitter* es cualquier desviación leve del período medio de una señal presumamente periódica. Hay muchos ejemplos físicos donde la inestabilidad es relevante. Algunos ejemplos de diferentes áreas son: (a) Stalberg *et. al* [?] encontraron que el intervalo de tiempo entre los dos potenciales de acción de las fibras de dos fibras musculares, que pertenecen a la misma unidad motora en los músculos humanos normales, muestra una variabilidad o inestabilidad; (b) Mecozzi *et. al* [?] detectaron jitter de temporal y variaciones de amplitud en enlaces ópticos utilizando transmisión de pulso altamente disperso; (c) Derickson *et. al* [?] realizó una comparación completa de la fluctuación de tiempo en el caso de los láseres semiconductores en modo bloqueado; (d) el California y Carnegie Planet Search en el Observatorio Keck [?] informó la inestabilidad de las estrellas en las velocidades radiales; (e) Roberts & Guillemin estudiaron los retardos debidos a las colas en etapas de *upstream multiplexing*, en una red de Modo de transferencia asíncrono (ATM); (f) Baron et al [?] consideró la calidad de la señal del *bunch clock* del *Large Hadron Collider* (LHC), en términos de inestabilidad, un problema fundamental porque sincroniza todos los sistemas electrónicos en el detector; (g) Marsalek *et. al* analizaron la relación entre la entrada sináptica y la fluctuación de fase de salida pico en neuronas individuales [?], etc.

Además, los instrumentos digitales se utilizan en cualquier experimento moderno y la inestabilidad inevitable en los sistemas de adquisición de datos produce incertidumbres en el tiempo y, por consiguiente, en cualquier determinación del espectro.

Este capítulo está dedicado a los osciladores de anillos (*RO*). Recalcemos que en esta aplicación particular, el *jitter* no siempre es indeseable. El *jitter* no es deseado en aplicaciones que usan un *RO* como generador de reloj [?, ?, ?, ?, ?]. Por el contrario, los generadores de números aleatorios *RNG* basados en *RO*'s, usan *jitter* como fuente de aleatoriedad, [?, ?]. El *jitter* también mejora la compatibilidad electromagnética para distribuir la frecuencia del reloj sobre una banda, mejorando la Compatibilidad Electromagnética (EMC) [?].

La determinación del *jitter* de fase en *RO* se ha estudiado en varios artículos: en [?] se presentó el estudio de tres medidas relevantes del *jitter* en el dominio del tiempo. En [?] se propuso un modelo para la generación y distribución del *jitter* en *RO*. En este artículo, los autores separan las fuentes de inestabilidad en deterministas y aleatorias (gaussianas); además, cada fuente se clasifica adicionalmente en local o global. Demuestran que las contribuciones más importantes son la inestabilidad gaussiana local y la inestabilidad determinística global y solo la primera debe usarse como una fuente de aleatoriedad de generadores de números aleatorios verdaderos (*TRNG*). El mismo enfoque se usó en [?, ?, ?, ?]. En Lubicz *et. al* se describe un método práctico y eficiente para estimar la tasa de entropía de un *TRNG* basado en osciladores libres; enfatizaron que su método no requiere extraer las señales del dispositivo y analizarlas con equipos externos [?] (una metodología que introduce fluctuación y distorsión extra en la señal medida debido a la cadena de adquisición de datos).

Por lo general, *jitter determinista* es el nombre que se le da a cualquier *jitter no gaussiano*. Está limitado y se caracteriza por su valor máximo de  $\Delta_{pp}$ . *Jitter aleatorio* es el nombre utilizado para la *jitter gaussiano* y se caracteriza por su valor RMS. A veces aparece *jitter periódico determinístico*. Tiene un *período* que es el intervalo entre dos tiempos el efecto máximo (mínimo); el inverso del período de tiempo es *la frecuencia del jitter*. El *jitter* periódico con una frecuencia de fluctuación inferior a  $10\text{Hz}$  usualmente se denomina *wander* y el nombre *jitter* está reservada solo a la fluctuación periódica con frecuencias en o por encima de  $10\text{Hz}$ . En comunicaciones, *jitter total* es  $T = \Delta_{pp} + 2nR_{rms}$  donde  $n$  es un número entre 6 y 8 relacionado con la tasa de error de bit (*BER*).

Los *ROs* son uno de los principales componentes de los circuitos integrados analógicos y digitales y se han utilizado ampliamente como osciladores *on-chip* para generar relojes en circuitos de alta velocidad. Además, los *ROs* se pueden implementar fácilmente en circuitos digitales programables como *FPGAs*. Las principales ventajas de los osciladores integrados *RO* sobre los *LC* son su área de chip más pequeña, su rango de funcionamiento más amplio (que puede ser sintonizado eléctricamente) y su menor consumo de energía.

Ya sea que se quiera usar o eliminarlo, el *jitter* en *ROs* debe medirse, lo que no es una tarea simple. La principal contribución de este trabajo es proporcionar una técnica de medición del *jitter* basada en cuantificadores de la teoría de la información (*ITQ*). Utilizamos un modelo estocástico cuya aleatoriedad está relacionada con la amplitud de la inestabilidad. Cada *ITQ* propuesto utilizado en este trabajo se basa en una entropía, es decir, una función de Shannon de la función de distribución de probabilidad (*PDF*) asignada a la serie de tiempo del proceso estocástico. También se pueden usar desequilibrios y complejidades [?, ?], pero no representan una mejora en nuestro caso. En trabajos anteriores [?, ?] mostramos que muchas *PDFs* diferentes pueden asignarse a la misma cadena de datos. La mejor opción depende de la aplicación específica. En este caso utilizamos dos opciones para *PDF*: el *histograma normalizado* y el *histograma de patrones de orden*. Se usa un plano de representación para comparar diferentes situaciones. Una vez que se elige la *PDF*, la Entropía de Shannon es la función básica que cuantifica la uniformidad de la *PDF*. Las *entropías normalizadas*, *entropías diferenciales* y *tasa entropía* son las otras *ITQs* evaluadas. En nuestro caso las *entropías diferenciales* obtienen los mejores resultados y se utiliza un *plano de entropías diferenciales* para comparar su sensibilidad como medida de *jitter*.

## 5.2. Determinación del *jitter* en *RO's*

Hay dos situaciones diferentes en lo que concierne al *jitter* en *ROs*: (a) en algunas aplicaciones es suficiente con asegurar que el *jitter* no perturba a la señal por encima de un límite aceptable. En este caso la señal se observa en un osciloscopio con un amáscara sobre la pantalla, lo que es suficiente para verificar que la señal se mantiene dentro de los márgenes de tolerancia; (b) en otros casos se precisa una determinación exacta del *jitter*. Entre esos casos está la caracterización de *ROs* considerada en este trabajo.

Los *ROs* ideales están compuestos por un numero impar de inversores. Cada inversor tiene un tiempo de propagación y por lo tanto los flancos de subida y bajada separados por medio período viajan a través de los inversores. Si todos los tiempos de propagación son constantes, la salida de este *RO* ideal es una señal cuadrada con un espectro de frecuencia discreto. Pero los tiempos de propagación no son constantes, por lo tanto hay *jitter*. El *jitter* distorsiona el espectro de potencia ensanchando cada delta en un máximo con cierta anchura.

Supongamos que  $T/2$  es el medio período de un *RO* ideal. Entonces está dado por:

$$\frac{T}{2} = k \sum_{i=1}^k d_i \quad (5.1)$$

En donde  $k$  es el número de inversores y  $d_i$  es el tiempo de propagación a través del  $i$ -ésimo inversor. Cuando hay *jitter*,  $d_i$  es una variable aleatoria que modelamos como:

$$d_i = D_i + \Delta d_i \quad (5.2)$$

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:

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  $\sigma_T = m * \sigma_s$ .

### 5.3. 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. 5.1). 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. 5.2).
- 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  $W$ ;

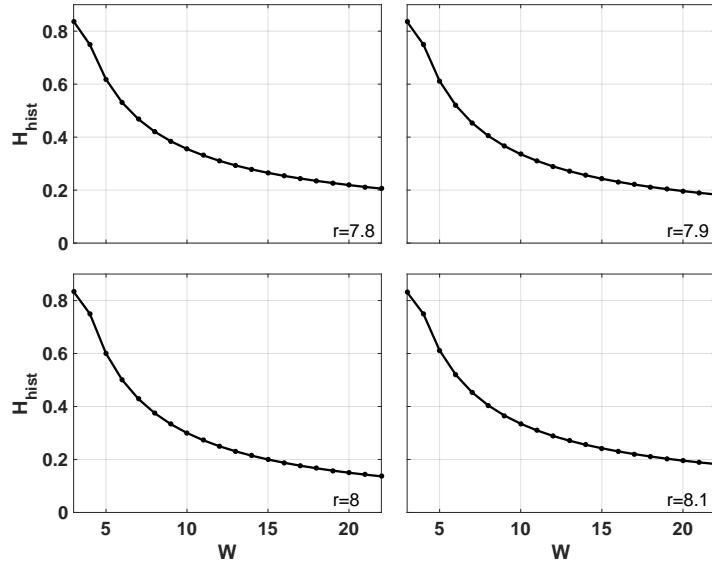


Figura 5.1: Normalized entropy  $H_W$  as a function of  $W$  for a jitter-less  $RO$  sampled with different values of  $r$ .

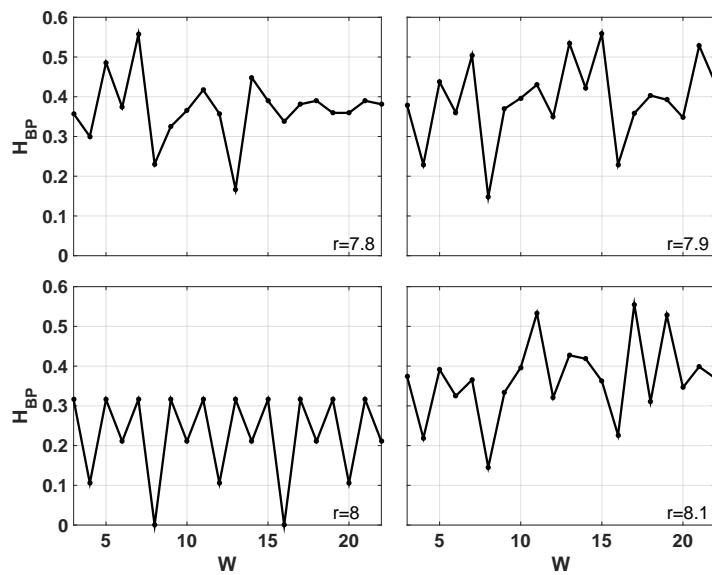


Figura 5.2:  $H_{BP}^{(D)}$  as a function of  $W$  for a jitter-less  $RO$  sampled with different values of  $r$ . Calculations are made without superposition of words

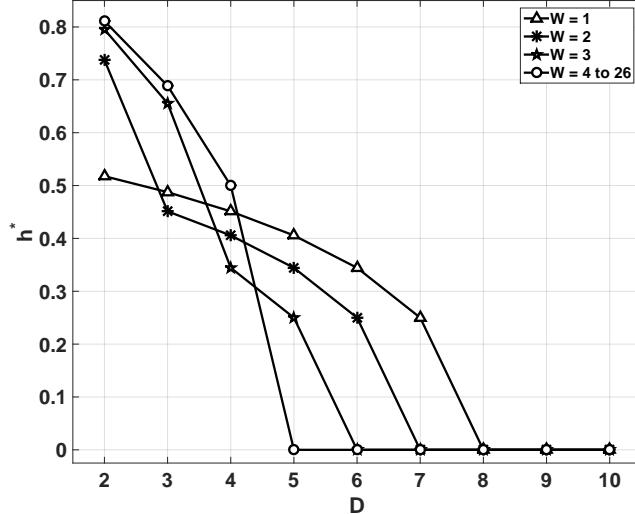


Figura 5.3:  $h^*$  as a function of  $D$  for a jitter-less  $RO$  sampled with  $r = 8$ .

- (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 5.3 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. 5.3 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 5.4, where  $h^*$  is plotted as a function of  $D$  with  $\sigma_T$  as a parameter. The values considered are  $\sigma_T = \{0(\text{no jitter}), 0,001, 0,002, 0,003, 0,004, 0,005, 0,007, 0,01, 0,02, 0,02, 0,04, 0,05, 0,07, 0,1\}$ . The inset of Fig. 5.4 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. 5.5 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. 5.6 shows jitter-less case,  $h$  is almost independent of  $W$  for  $W \geq 4$ . In this paper, we adopted  $W = 6$ . Figure 5.7 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. 5.8 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.

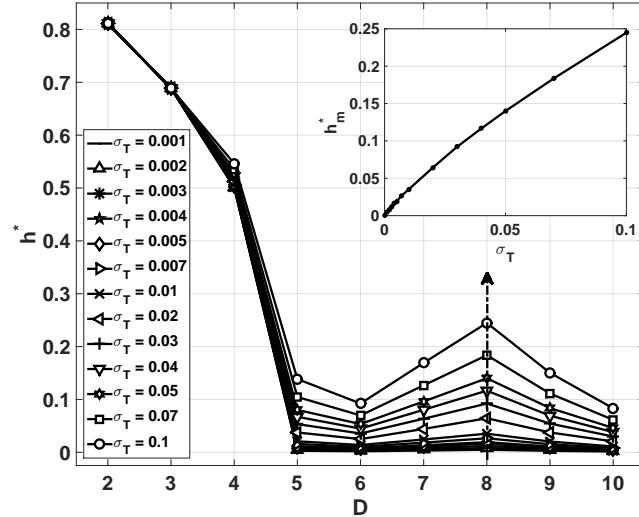


Figura 5.4:  $h^*$  as a function of  $D$  for a *RO* sampled with  $r = 8$  for a world length  $W = 6$  for jitter with several variances. The inset shows  $h^*$  as a function of  $\sigma_T$  for  $r = 8$ ,  $W = 6$  and  $D = 8$ .

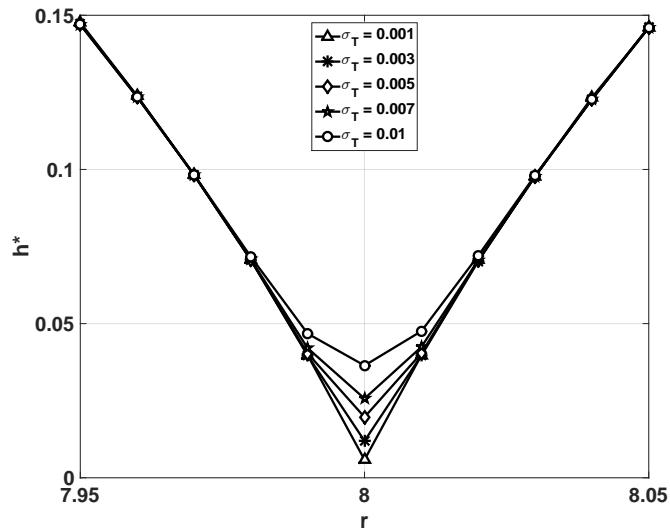


Figura 5.5:  $h^*$  as a function of  $r$  for  $r \in [7.95, 8.05]$ , with several  $\sigma_T$ ,  $W = 6$  and  $D = 8$ . The curve has a minimum at the correct value  $r = 8$ .

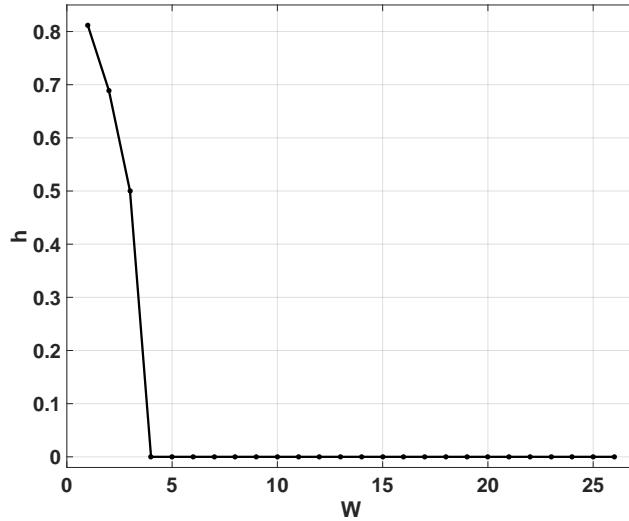


Figura 5.6:  $h$  as a function of  $W$  for a jitter-less RO sampled with  $r = 8$ .

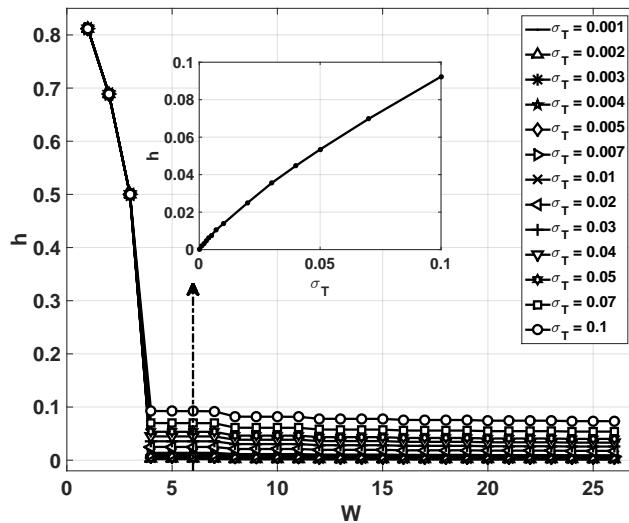


Figura 5.7:  $h$  as a function of  $W$  for a RO sampled with  $r = 8$ , for jitter with several variances. The inset shows  $h$  as a function of  $\sigma_T$  for  $r = 8$  and  $W = 6$ .

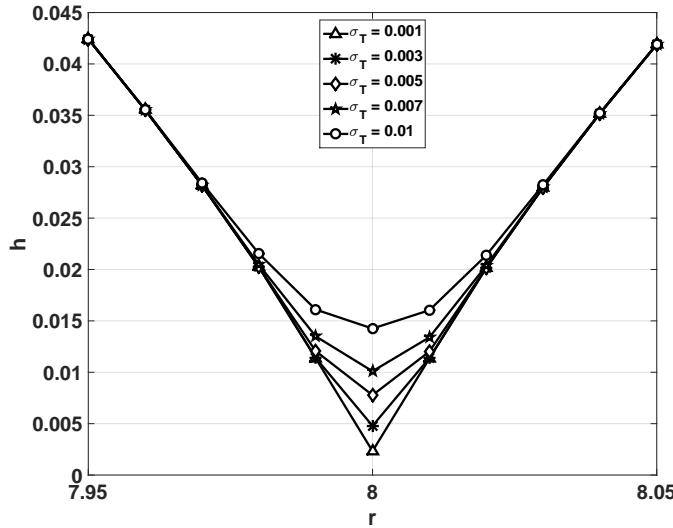
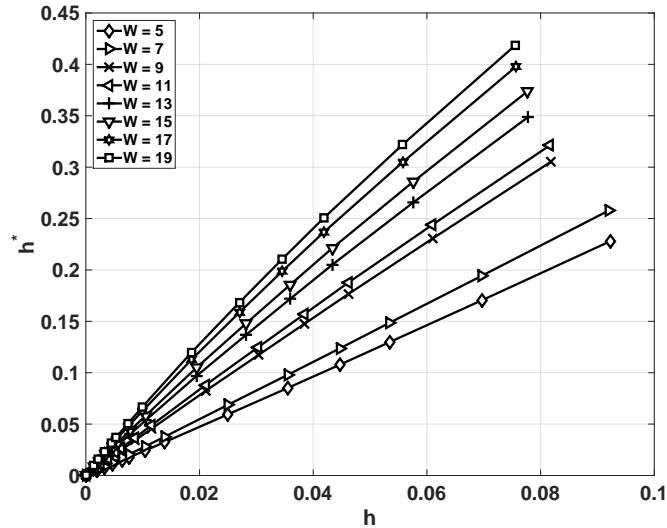
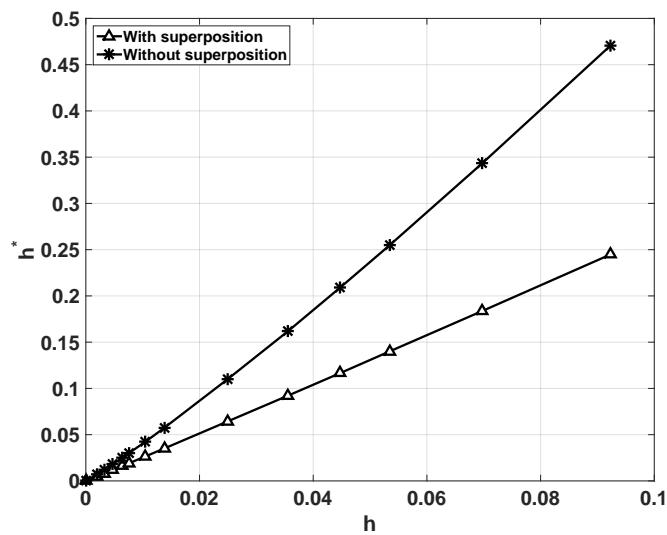


Figura 5.8:  $h$  as a function of  $r$  for  $r \in [7.95, 8.05]$ , with several  $\sigma_T$  and  $W = 6$ . The curve has a minimum at the optimum value  $r = 8$ .

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 5.9. 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. 5.4, and 5.7. 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.

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. 5.10 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.

Figura 5.9:  $h^*$  as a function of  $h$  for  $r = 8$ ,  $D = 8$  and different values of  $W$ .Figura 5.10:  $h^*$  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).

## 5.4. 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 5.3 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 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 6**

## **Conclusiones**



## **Apéndice A**

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

Cosas que distraen en la tesis.



# Bibliografía