Universidad de Valladolid

Escuela Técnica Superior de Ingenieros de Telecomunicación

Universidad de Valladolid

TRABAJO DE FIN DE GRADO

GRADO EN INGENIERÍA DE TECNOLOGÍAS DE TELECOMUNICACIÓN

Generación de números aleatorios en sistemas de fibra óptica

Autor: D. Noé Rodríguez Rivas

Tutor: D. Juan Carlos García Escartín

Valladolid, julio de 2019

TÍTULO: **Generación de números aleatorios en sistemas de fibra óptica**

AUTOR: **D. Noé Rodríguez Rivas**

TUTOR: **D. Juan Carlos García Escartín**

DEPARTAMENTO: **Teoría de la Señal y Comunicaciones e Ingeniería Telemática**

TRIBUNAL

PRESIDENTE: **D. Pedro Chamorro Posada**

VOCAL: **Dña. María Jesús González Morales**

SECRETARIO: **D. Julio Sánchez Curto**

SUPLENTE 1: **D. Juan Carlos García Escartín**

SUPLENTE 2: **D. Luis Miguel San José Revuelta**

FECHA: 26 de julio de 2019

CALIFICACIÓN:

Resumen

El mundo de los sistemas de seguridad se encuentra en constante desarrollo, siendo preciso explorar nuevos mecanismos que permitan encriptar la información de una forma fiable. Debido a ello, la criptografía precisa de una investigación continua a la hora de obtener generadores de números aleatorios de gran calidad, encontrando en los sistemas de fibra óptica un campo muy interesante para seguir evolucionando. Aprovechando este hecho, el presente Trabajo de Fin de Grado tendrá como objetivo lograr un generador de números aleatorios, diseñando para ello un sistema de fibra óptica fiable cuyos fundamentos se puedan comprender fácilmente. Se aprovechará el principio de realimentación de un láser óptico para generar niveles de potencia con variaciones lo suficientemente importantes de modo que puedan generar números aleatorios tras un proceso de adquisición y procesamiento. Durante el transcurso del Trabajo, se abordará tanto el estudio del cierto comportamiento caótico que puedan presentar las dinámicas del láser realimentado como su calidad de generación de números aleatorios. Por último, se destaca la gran versatilidad en cuanto a ciertos parámetros que rigen el funcionamiento del generador, adaptándose de este modo a las limitaciones impuestas por la existencia finita de recursos.

Palabras clave: generador de números aleatorios, caos determinístico, láser óptico, fibra óptica, osciloscopio, extractor, aleatoriedad, realimentación óptica.

Abstract

The world of security systems is in constant development, being necessary to explore new mechanisms that allow information to be encrypted in a reliable way. Due to this, cryptography requires continuous research when it comes to obtaining high quality random numbers generators, finding in optical fiber systems a very interesting field to achieve its purpose. Taking advantage of this fact, this Final Degree Project will aim to achieve a random numbers generator, designing a reliable fiber optic system, whose foundations can be easily understood. The feedback principle of an optical laser will be used to generate power levels with variations important enough so that they can produce random numbers after an acquisition and processing process. Along this Project, it will be studied the certain chaotic behavior that the dynamics of the feedback laser could present, as well as, its quality of generation random numbers. Finally, it highlights the great versatility in terms of certain parameters that govern the operation of the generator, adapting in this way to the limitations imposed by the finite existence of resources.

Keywords: random number generator, deterministic chaos, optical laser, optical fiber, oscilloscope, extractor, randomness, optical feedback.

Agradecimientos

En primer lugar, no podría comenzar de otra forma que no fuese agradeciendo el esfuerzo realizado por mi madre y mi padre durante todos estos años de carrera, siendo las primeras personas con las que he compartido tanto mis alegrías como decepciones. Al final los desvelos que les haya podido causar han merecido la pena. Tampoco me puedo olvidar de mis familiares más cercanos como son mi abuela, tíos y primos (grandes y pequeños) por su apoyo constante.

Todos estos años han sido de lo más intensos, tanto fuera como dentro de las aulas, y sin lugar a dudas no hubiesen sido lo mismo sin todas las personas cercanas que me han acompañado en un momento u otro. Intentando no olvidarme de nadie, solo puedo agradecer entre otras cosas por su cariño a Alejandro, Álvaro, Andrea, Bustillo, Garazi, Henar, Jorge y Lucía. Mención especial también merece mi amiga y compañera de carrera Marta por todos estos años trabajando codo a codo.

Por último, no puedo pasar por alto el esmero y dedicación de mi tutor Juan Carlos, ayudándome en todo lo necesario para que las prácticas y el proyecto hayan salido adelante de la mejor forma posible.

Índice

Índice de figuras y tablas

1. Introducción

El presente Trabajo de Fin de Grado estará centrado en el diseño y fabricación de un generador de números aleatorios mediante un sistema de fibra óptica que tenga como fuente emisora un láser que pueda mostrar un comportamiento caótico bajo ciertas circunstancias. La elección de este tipo de sistema para generar números aleatorios se basa en las altas tasas de bits que es capaz de proporcionar, siendo realmente útil en el mundo de la criptografía actual [1]. A lo largo del proceso de investigación llevado a cabo, es necesario comprender ciertos comportamientos caóticos que presentará nuestro láser gracias al fenómeno de la realimentación, siendo esta característica una de las principales a analizar en el desarrollo del trabajo. Mediante diferentes herramientas se analizará la calidad de los números aleatorios obtenidos, del mismo modo que se emplearán diversos mecanismos para intentar cuantificar los indicios de fenómenos caóticos que se están produciendo.

Las primeras etapas que se llevarán a cabo se extienden desde el diseño inicial de nuestro sistema de fibra óptica realimentado hasta la elección de los componentes adecuados, realizando diversos experimentos intermedios para poder obtener los mejores resultados posibles de cara a obtener números aleatorios de la máxima calidad. Las limitaciones impuestas en los recursos disponibles en el laboratorio también influirán a la hora de poder llevar a cabo los experimentos, ya que el abanico de circuitos de realimentación implementables se ve reducido. Una vez encontradas pruebas evidentes de estar ante un circuito de realimentación que proporcione buenos resultados, el trabajo se centrará describir sus características, evidenciando los motivos por los que ha sido escogido. Del mismo modo se comentará la influencia de los parámetros que rigen su funcionamiento, pudiendo identificar de esta forma cuales son las mejores condiciones para la generación de números aleatorios.

Aparte de lograr el objetivo de obtener un sistema robusto cuyos resultados sean capaces de superar los tests propuestos para el análisis de aleatoriedad, también resulta imprescindible lograr la correcta integración y aprovechamiento del equipo de medición empleado como es el osciloscopio digital. Debido a ello también hay que destacar la necesidad de una comprensión exhaustiva de su funcionamiento y mecanismos de adquisición de datos, ya que un error en todo el proceso puede llevar a modificaciones equivocadas en el propio circuito óptico. La posibilidad de automatizar las tareas de este equipo también resulta fundamental para que todos los estudios realizados puedan ser realizados de la forma más cómoda posible, posibilitando contar con un amplio abanico de datos que poder analizar.

2. Objetivos

Resulta de vital importancia fijar una serie de objetivos que resulten útiles a la hora de lograr el generador de números aleatorios deseado. Estos objetivos serán progresivos, de modo que cada uno de ellos se encuentra relacionado con el inmediatamente anterior. De esta forma será más sencillo comprender la secuenciación de las diferentes etapas recogidas en los capítulos de este trabajo. A continuación se enumeran de forma ordenada los objetivos marcados:

- Determinar un esquema que permita fijar un porcentaje de potencia de retorno al láser con un cierto retardo en su camino de realimentación para que se pueda generar un comportamiento caótico en las dinámicas del láser.
- Mediante los diferentes elementos existentes en el laboratorio, encontrar un sistema de realimentación robusto que pueda asegurar el porcentaje de realimentación descrito en el anterior objetivo.
- Lograr un sistema de adquisición de muestras que permita configurar parámetros como la tasa de muestreo.
- Justificar mediante histogramas el uso del circuito de realimentación para generar números aleatorios.
- Comprobar mediante diversas herramientas de análisis de caos si el comportamiento de las muestras posee indicios de ser caótico.
- Determinar un criterio de decisión en el que se vean involucradas las muestras obtenidas para generar números aleatorios.
- Analizar la calidad de los números aleatorios obtenidos.
- Encontrar un proceso de extracción que permite mejorar la calidad de los números aleatorios obtenidos.

3. Generación de comportamientos caóticos en el láser

A la hora de conseguir un generador de números aleatorios, habrá que tener en cuenta ciertos comportamientos del láser empleado. Entre estos comportamientos sin lugar a dudas destacada el caótico. La existencia de diversas inestabilidades en la salida del láser es un claro objeto de estudio, del mismo modo que conocer las perturbaciones externas que favorecen esas inestabilidades. Si su comportamiento se encuentra derivado de diferentes reglas procedentes de las dinámicas del láser, se puede hablar de la existencia de caos determinístico, pudiéndose diferenciar de otros fenómenos como el ruido estocástico o cuántico. En el caso que comprende al funcionamiento del láser planteado, será preciso tener en cuenta la existencia de ruido shot en el análisis. También hay que destacar como el caos determinístico puede ser descrito mediante una serie de ecuaciones matemáticas que ayudan a comprender un fenómeno que un primer momento puede poseer una apariencia impredecible.

Una de las principales características del caos determinístico se sustenta en una gran dependencia respecto a las condiciones iniciales, tal y como demostró Lorentz en 1963 [2]. Esto supone que si dos secuencias caóticas temporales tienen como punto de partida unas condiciones iniciales cercanas pero con ligeras diferencias, será posible predecir sus trayectorias al principio, pero con el paso del tiempo tenderán a divergir de forma muy rápida y ya no podrán presentar el mismo comportamiento que a su inicio. Aquí es donde entran en juego los diferentes mecanismos para intentar predecir su comportamiento inicial y cómo de "caótico" será su comportamiento futuro. Tanto la dimensión de correlación como los exponentes de Lyapunov y el exponente de Hurst serán de lo más útiles a la hora de analizar los datos obtenidos a través del generador de números aleatorios, siendo conscientes de si se está accediendo a un sistema caótico su generación [3].

Ahora bien, en primer lugar hay que describir de una forma clara el mecanismo físico que provoca este comportamiento. A través del empleo de un láser semiconductor y un circuito óptico, será posible explicar la alteración sufrida en la salida del láser si el mencionado circuito se emplea para que un porcentaje de la luz emitida regrese a su origen. Cuando la luz que viaja de regreso incide sobre la cavidad de emisión del láser se produce en ella una interacción entre los fotones emitidos y los reflejados. La perturbación producida entre los fotones afectará al tamaño de la cavidad del láser, teniendo en cuenta que no contribuye toda la luz reflejada ya que hay una parte de ella que puede haberse dado en la superficie de la cavidad o en el propio borde de la fibra óptica [4].

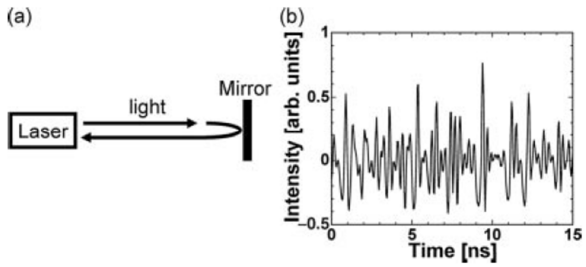


Figura 1a. Modelo básico de un láser semiconductor realimentado. Figura 1b. Intensidad de emisión de un láser semiconductor realimentado [4].

En el caso que concierne a la generación de números aleatorios o la distribución de claves seguras este hecho será positivo, mientras que en otro tipo de aplicaciones se verá como un inconveniente, ya que requerirán un nivel de amplitud continua en el flujo de salida del láser. De ahí que se empleen medidas como asegurar un espectro óptico muy estrecho para evitar que este fenómeno pueda tener lugar o minimizar la potencia reflejada de los conectores que unen los diferentes elementos del circuito. Precisamente en el siguiente capítulo del trabajo se abordará el porcentaje de potencia de realimentación necesario para que se pueda a empezar a considerar que las variaciones del flujo de salida pueden ser producidas debido a la existencia de un comportamiento caótico de las dinámicas del láser [4].

Aunque en este trabajo el comportamiento caótico se enfocará hacia la generación de números aleatorios, su estudio también está presente en otras numerosas aplicaciones. Una de las más importantes es la de los sensores remotos, es decir, la de adquirir información sin necesidad de contacto físico con el sistema que se está analizando. En este caso el tipo de sensores remotos en los que se emplea es en los Lidar (*Light Detection and Ranging)*, teniendo gran utilidad en tareas como el reconocimiento automático de un objetivo o *target detection* [4].

También hay que destacar otras aplicaciones que puede tener el análisis del caos orientado en una dirección opuesta: el estudio de cómo controlar los comportamientos caóticos. Esta línea de desarrollo ha crecido enormemente a partir de los 90, en parte gracias a aplicaciones como la separación ciega de señales o BSS (*Blind Signal Separation)*. Este campo trata de separar señales que se encuentran mezcladas sin necesidad de conocer gran información a mayores de las señales por separado, siendo realmente frecuente en tareas de multiplexación y demultiplexación. Otra aplicación destacada en esta dirección se basa en la obtención de láseres ultraestables, siendo fuentes de emisión continua que no presentan ningún tipo de fluctuación. Por último, hay que mencionar una tercera vía de investigación de

los comportamientos caóticos relacionados con la sincronización del caos y su empleo en diferentes sistemas de comunicación que requieren gran precisión [4].

4. Componentes empleados y su caracterización

El presente capítulo se centrará en describir y analizar el comportamiento de los componentes empleados en nuestro sistema generador de números aleatorios. A través de diversas pruebas comprobaremos las similitudes existentes entre las pruebas realizadas y lo indicado en las hojas de especificaciones de los diversos componentes, tratando de caracterizar al máximo su funcionamiento y la posible existencia de pérdidas adicionales respecto a las señaladas por el fabricante. De esta forma será posible conocer mejor el funcionamiento de nuestro sistema y minimizar posibles errores derivados de la pérdida de potencia. A continuación se exponen los diversos componentes así como los conectores empleados en sus conexiones junto con los equipos de medida, teniendo lugar una descripción de sus características destacadas y análisis de su linealidad en la transmisión de potencia.

4.1 Osciloscopio digital PicoScope 9200

El osciloscopio digital PicoScope es el equipo empleado tanto a la hora de medir los niveles de potencia del sistema como la realización de la captura de datos y su visualización con herramientas como el histograma. Con diferentes formatos de visualización que permiten unir las muestras obtenidas, resulta de gran utilidad para poder hacerse una idea de la señal que tenemos en nuestro sistema. Merece la pena destacar que las medidas proporcionadas por el osciloscopio se encuentran en voltios, mientras que por lo general se trabaja siempre con términos de potencia en lo referido a las cantidades presentes en las hojas de especificaciones.

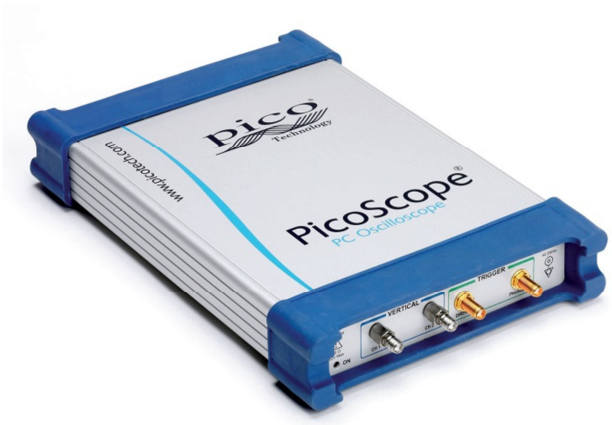


Figura 2. Osciloscopio digital PicoScope 9200.

Como características principales se puede destacar un factor de escala comprendido entre 2 mV/div a 500 mV/div, bases temporales de 10 ps/div a 50 ms/div, resolución del ADC

de 16 bits, longitud del registro de datos de 32 a 4096 puntos, histograma vertical u horizontal, tensión máxima de entrada del disparador de seguridad de ±2 V, conector de entrada del disparador SMA (F), máxima potencia pico de entrada de +7 dBm (1310 nm), potencia máxima de 1,9 A y rango de temperatura de funcionamiento+5 °C a +35 °C (+15 °C a +25 °C para una precisión establecida) [5].

Una de las funciones más destacadas de este osciloscopio es la posibilidad de mostrar un histograma obtenido a partir de los datos capturados, incluyendo en él una serie de medidas automáticas como la media, amplitud pico a pico, la mediana, la desviación estándar o el valor máximo de la señal en la ventana de tiempo seleccionada. Precisamente la desviación estándar será de gran utilidad a la hora de conocer si las muestras se encuentran distribuidas cercanas a la media o si se encuentran en un rango mucho más amplio, algo que será de interés a la hora de realizar modificaciones en el sistema para lograr la generación de números aleatorios. En la *Figura 3* se puede comprobar cómo se visualiza el histograma y estas medidas asociadas.

Otra de las características importantes de este osciloscopio es la posibilidad de programarlo en Python a través de una serie de comandos. Esto resultará de gran ayuda a la hora de automatizar el proceso de captura de datos, al mismo tiempo de incluir la calibración por defecto o bien controlar el número de muestras que se desea tomar o su longitud. Del mismo modo, también existen comandos asociados a la visualización de los datos, tales como la escala de tiempos o el factor de escala de la amplitud.

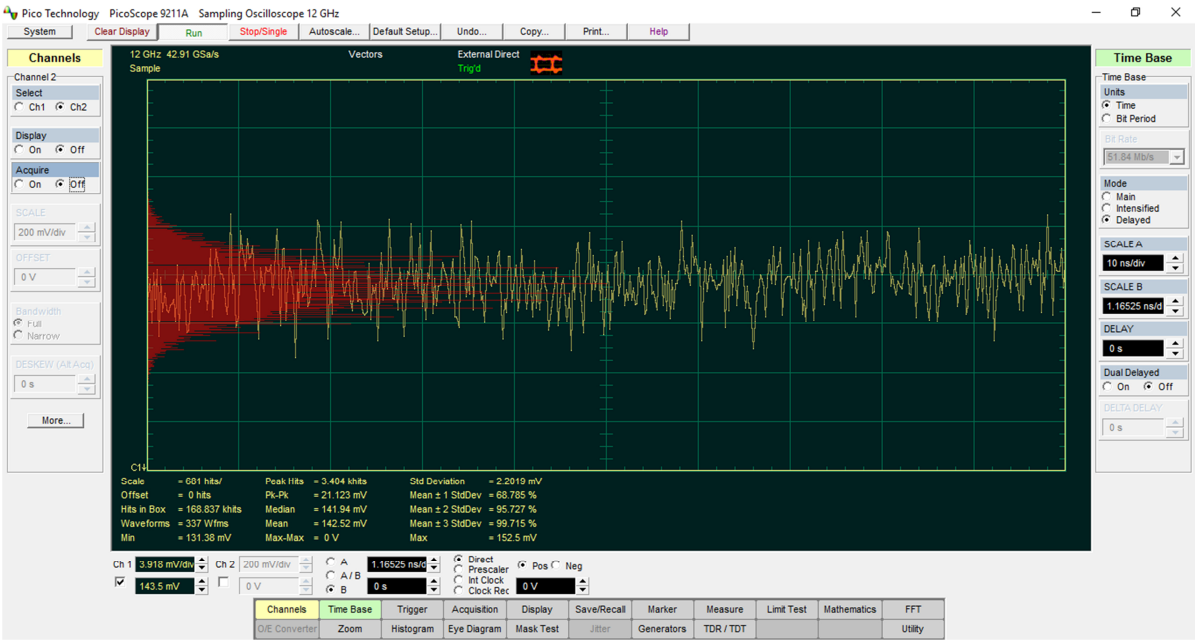


Figura 3. Captura del histograma realizado por el osciloscopio PicoScope 9200 A.

4.2 Detector de silicio (DET025AFC(/M))

Al estar trabajando en fibra óptica, las señales que se transmiten son luminosas, de modo que para realizar la conversión de la señal luminosa a una señal eléctrica de alta frecuencia es preciso emplear un detector. En este caso se emplea un detector de silicio de fibra acoplada fabricado por ThorLabs denominado DET025AFC(/M), capaz de trabajar en una longitud de onda comprendida entre 400 y 1110 nm. Su justificación a la hora de emplearlo para realizar medidas de potencia radica en que el láser con el que se trabajará lo hará en el espectro del rojo, en torno a los 638 nm, siendo este el más adecuado y el que mayor precisión proporcionará entre los disponibles en el laboratorio.

Figura 4. Detector de silicio (DET025AFC(/M)).

Este detector proporciona sus medidas usando fotodiodo, además de presentar una R_{LOAD} de 50 Ω, una señal de salida SMA y un voltaje máximo de salida de 2V. Teniendo en cuenta que las especificaciones del resto de componentes suelen venir en unidades de potencia óptica (W), será preciso emplear la ecuación *voltaje (V) = responsividad (A/W) x potencia óptica (W) x* R_{LOAD} *(Ω)* para conocer su valor. Como el detector se encuentra trabajando a 638 nm, su sensibilidad en las hojas de especificaciones nos indica que es de 0,3673 (A/W). El detector incluye una pila A23 de 12V remplazable, proporcionando una fuente de alimentación de bajo ruido [6].

En la *Figura 5* se muestra el comportamiento del detector ante diferentes niveles de entrada de potencia óptica. Hay que tener en cuenta que en la obtención de la recta influye tanto la responsividad presente del detector para la longitud de onda del láser en la que se encuentra trabajando, como la precisión que presenta el láser en relación a la potencia que se espera lograr y la corriente necesaria para ello. Hay que destacar también el nivel mínimo de potencia en torno a 2 mW que es posible lograr en el láser, dado que existe un umbral de corriente de aproximadamente 72 mA que se debe de superar para que el láser comience a emitir. Del mismo modo, el límite de potencia a la entrada del detector no podrá exceder los 18mW para no dañarlo, por lo que será preciso introducir un atenuador a su entrada si se

quiere trabajar con potencias más elevadas. El empleado para obtener tanto esta gráfica como las sucesivas ha sido de 10 dB.

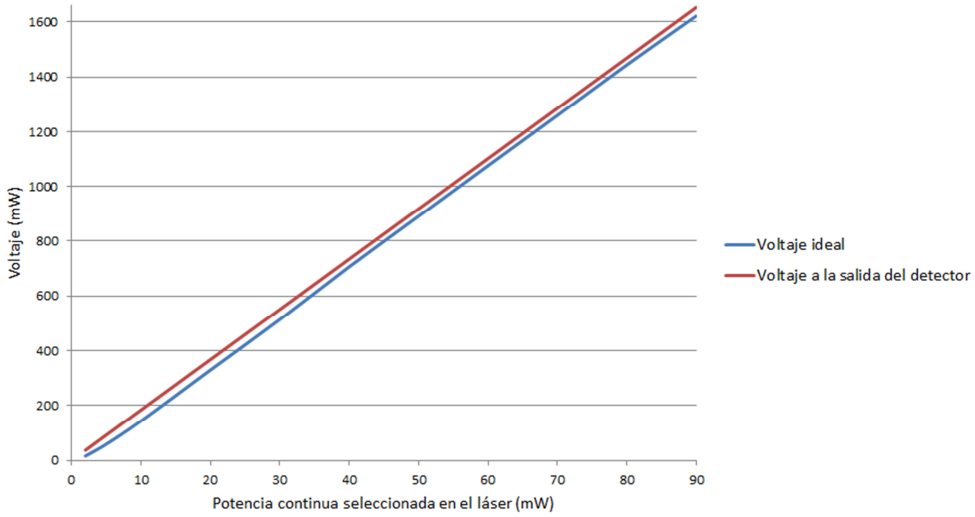


Figura 5. Comparativa del voltaje a la salida del detector DET025AFC(/M) con el ideal obtenido a partir de la fórmula de conversión.

4.3 Láser Cobolt MLD 638 nm

El sistema precisa de una fuente, que en este caso será un láser emisor de señales luminosas medidas en potencia óptica. El elemento encargado de ello será un láser, de modo que las propias señales luminosas transmitidas por él servirán para su realimentación. El láser empleado en el sistema trabajará en el espectro del rojo (618-780 nm), concretamente a 638 nm de longitud de onda, ya que en los experimentos realizados a través de diferentes circuitos de realimentación planteados los resultados respecto al porcentaje de potencia realimentada se adecuaban a lo esperado. Por otra parte, el láser en rojo cuenta con un disipador Cobolt HS-03 que permite que las oscilaciones de temperatura sean mínimas, de modo que será beneficioso para obtener una mayor estabilidad en las señales ópticas emitidas.

Figura 6. Láser Cobolt MLD 638 nm.

Como principales características de este láser se encuentran una alimentación de 5V, una precisión de la longitud de onda de trabajo de ±5 nm, cociente de extinción de la polarización 100:1 vertical, trabajo óptimo a temperatura ambiente de 10 ℃ a 40 ℃ y rango de potencias de salida comprendido entre 2 y 90 mW. Merece la pena destacar como se puede trabajar en dos modos: potencia constante y corriente constante. En el modo de corriente constante, el fotodiodo que controla la potencia del láser funciona de una forma similar a una fuente de corriente, manteniendo la corriente constante a través de sus terminales mientras que la potencia varía acorde a las condiciones de la resistencia de carga para lograr precisamente que la corriente que circule a través de ella no varíe. Sin embargo, en el modo de potencia constante ocurre precisamente lo contrario, ya que la potencia, se mantiene constante a costa de evitar que su nivel no baje. Esto se consigue gracias a contar con una pequeña realimentación de la potencia emitida. En este caso se trabajará en el modo de corriente constante, ya que si se trabajase en modo de potencia constante, las propias fluctuaciones de la potencia el láser debido a la realimentación podrían ser compensadas por los mecanismos del fotodiodo, intentando que la potencia no variase.

Cierto es que en todo momento resulta más interesante trabajar en unidades de potencia, ya que en las hojas de especificaciones de los latiguillos de fibra óptica viene indicada la potencia máxima que son capaces de soportar, del mismo modo que las pérdidas se calculan a partir de la potencia. Sin embargo esto no resulta impedimento para poder regular el funcionamiento del láser a través de la corriente de alimentación del fotodiodo. Al igual que ocurría en el osciloscopio PicoScope, este láser posee un conjunto amplio de comandos en Python que permiten controlar las características del mismo [7].

4.4. Latiguillo de fibra óptica P5-630A-PCAPC-1

Uno de los principales problemas que se encuentran a la hora de minimizar las pérdidas del sistema es que la fibra de salida del láser Cobolt presenta una conexión APC FC, mientras que el resto de elementos del sistema presentan conexiones PC FC. Debido a ello y para evitar que principalmente las pérdidas por reflexión sean muy elevadas, es precisa una conversión de APC a PC, necesitando un latiguillo que se encargue de ello. Esta tarea se llevará a cabo con el latiguillo P5-630A-PCAPC-1 fabricado por Thorlabs. Como características principales se puede destacar su rango de trabajo entre 633 y 780 nm, una atenuación máxima de ≤15 dB/km, pérdidas de retorno típicas de 50 dB y mínimas de 40 dB para conectores PC (60 dB mínimas para APC) y 2 dB de pérdidas de inserción para 633 nm [8].

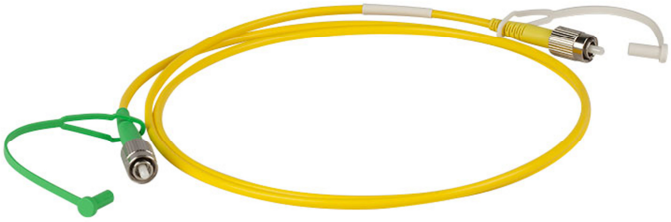


Figura 7. Latiguillo de fibra óptica P5-630A-PCAPC-1.

A continuación se comprueba como transmitiendo a diferencias potencias continuas en 638 nm y con un conector ADAFC3 entre la conexión del láser y del latiguillo P5-630A-PCAPC-1, las pérdidas que se tienden a incrementar de forma constante cuanto mayor sea la potencia transmitida. Nótese que para calcular la potencia de entrada es necesario la visualización en voltaje en el osciloscopio digital del láser conectado directamente al detector y posteriormente realizar su conversión a potencia a través de la ya mencionada fórmula *voltaje (V) = responsividad (A/W) x potencia óptica (W) x* R_{LOAD} *(Ω).* Como se puede observar, las pérdidas resultan ligeramente superiores cuanta mayor potencia deseamos transmitir, aunque si bien es cierto el incremento no resulta muy destacado, pudiendo trabajar en prácticamente todos los casos con unas pérdidas cercanas a 0,8 dB, quedándose lejos de los 2 dB de pérdidas tan solo de inserción que se aseguran como máximo en las especificaciones. Por lo tanto se puede deducir que las conexiones realizadas son buenas.

En la transmisión de bajas potencias hay que tener también en cuenta que la existencia de diferentes tipos de ruido, como puede ser el ruido *shot*, alcanzan niveles menores pero algo significativos respecto a los de la propia señal transmitida. Debido a ello, el nivel de potencia en la salida del P5-630A-PCAPC-1 no decae tanto ya que es el propio ruido el que lo mantiene, asemejándolo a la salida en el detector y encontrándonos con unas pérdidas ligeramente inferiores.

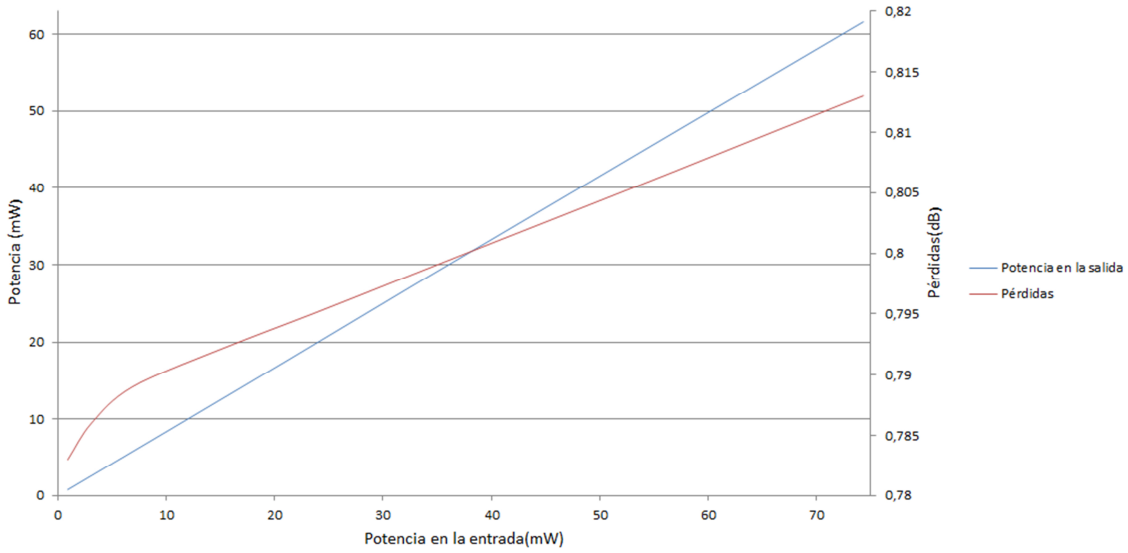


Figura 8. Análisis de la potencia y pérdidas a la salida del P5-630A-PCAPC-1.

4.5 Acoplador FC632-50B-FC - 2x2

En el sistema de generación de números aleatorios es necesario un camino de vuelta para que el láser pueda ser realimentado. Debido a ello se va a emplear un acoplador que no es más que un dispositivo capaz de separar en varios caminos la potencia transmitida en sus puertos de entrada. A través del parámetro relación de acoplo, podremos saber cuál es el porcentaje de potencia que se transmite a través de cada uno de sus puertos de salida. De este modo, si tuviésemos una relación de acoplo de 30:70, se transmitiría por uno de los puertos de salida el 30% de la potencia que llega a un determinado puerto de entrada del acoplador, mientras que por el otro puerto de salida se transmitiría el 70%. Conectando dos puertos de salida se obtendrá la realimentación, ya que cada puerto puede transmitir potencia en ambos sentidos.

Figura 9. Acoplador FC632-50B-FC - 2x2.

En este caso, el acoplador disponible en el laboratorio tendrá una relación de acoplo 50:50, utilizando uno de cuatro puertos capaz de trabajar a 632 nm de longitud de onda con un margen de ±15 nm. Las características proporcionadas por el fabricante nos indican que las pérdidas por inserción son de 3,31 dB de desde el puerto 1 al puerto 3, 3,47 dB desde el puerto 1 al puerto 4, 3,42 dB desde el puerto 2 al puerto 3 y de 3,32 dB desde el puerto 2 al puerto 4. Merece la pena destacar que en teoría los caminos son bidireccionales, por lo que si se invierten las entradas por las salidas las pérdidas se deberían ajustar a estos datos. También se indica que las pérdidas por exceso son de 0,7 dB, pudiendo trabajar en un rango de temperatura comprendido entre -40 ℃ y 85 ℃ [9].

A continuación se analiza la relación de acoplo presente entre los puertos, pudiendo identificar en la *Figura 10* como la relación de acoplo entre los puertos 1 y 3 es la mayor de todas, permaneciendo prácticamente constante salvo para potencias bajas. Estos datos coinciden también con las pérdidas indicadas por el fabricante, ya que las existentes en el camino entre los puertos 1 y 3 son las menores. Al igual que se explicó en el apartado del P5-630A-PCAPC-1, a bajas potencias predominan diferentes tipos de ruido que alterarán las relaciones de acoplo, ya que la potencia transmitida en realidad resulta la propia del ruido. Para conectar la salida del láser con los puertos del acoplador, se ha empleado un conector ADAFC1.

A la hora de obtener las medidas con las que se ha elaborado la gráfica, merece la pena destacar que los valores mostrados representan el promedio de varias pruebas, ya que cada vez que se conecta y desconectan los puertos de un dispositivo, al ser un proceso manual, influyen factores como su alineación, la curvatura de los caminos de fibra o el propio roscado de los conectores. A pesar de ser variaciones no muy pronunciadas, hay que tenerlas en cuenta para poder entender como no es posible obtener con total exactitud las mismas medidas en dos pruebas consecutivas.

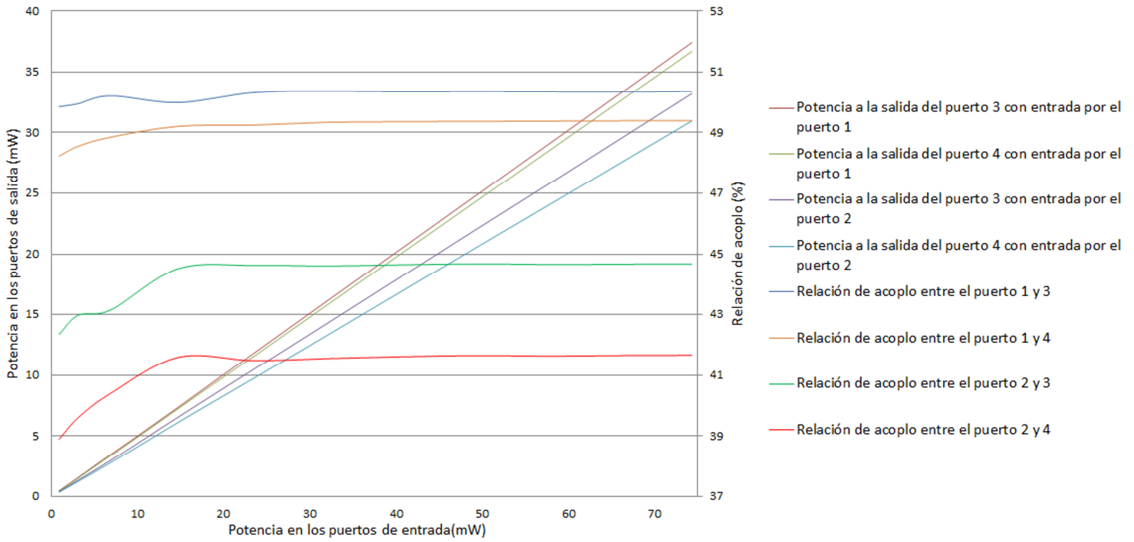


Figura 10. Análisis de potencia y pérdidas en la salida de los puertos del FC632-50B-FC - 2x2.

4.6 Atenuador variable VOA630-FC

En el camino de realimentación del láser será preciso introducir un atenuador variable, ya que la respuesta del láser no es la misma según los diferentes niveles de potencia que reciba. Debido a ello, es preciso controlar la potencia que regresa a él de forma precisa para poder alcanzar un mayor comportamiento caótico. En este caso se emplea el atenuador óptico variable VOA630-FC de Thorlabs, que no es más que un latiguillo de fibra que en un determinado punto posee una especie de obturador controlable que permite tapar total o parcialmente el camino físico de las señales luminosas, pudiéndose controlar a través de un tornillo. Como principales características de este atenuador, se encuentran un rango de operación entre 620 y 650 nm, un rango de atenuación de 3 a 50 dB y pérdidas por inserción menores de 3 dB [10].

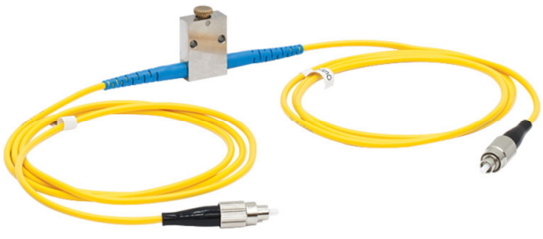


Figura 11. Atenuador variable VOA630-FC.

En los análisis realizados en cuanto a la variabilidad de potencia lograda, comprobamos que es posible moverse entre la completa atenuación de potencia hasta un valor mínimo, presentando en la *Figura 12* la potencia a la salida del puerto output y las

pérdidas totales cuando la atenuación es mínima, es decir, cuando no hemos girado el tornillo. Hay que destacar que el atenuador teóricamente debería ser totalmente bidireccional, pero sin embargo presenta una ligera mayor transmisión de potencia si empleamos como entrada el puerto marcado como output. Al igual que en apartados anteriores, la pérdidas a bajas potencias resultan menores ya que hay niveles de ruido ligeramente equiparables a los la señal transmitida.

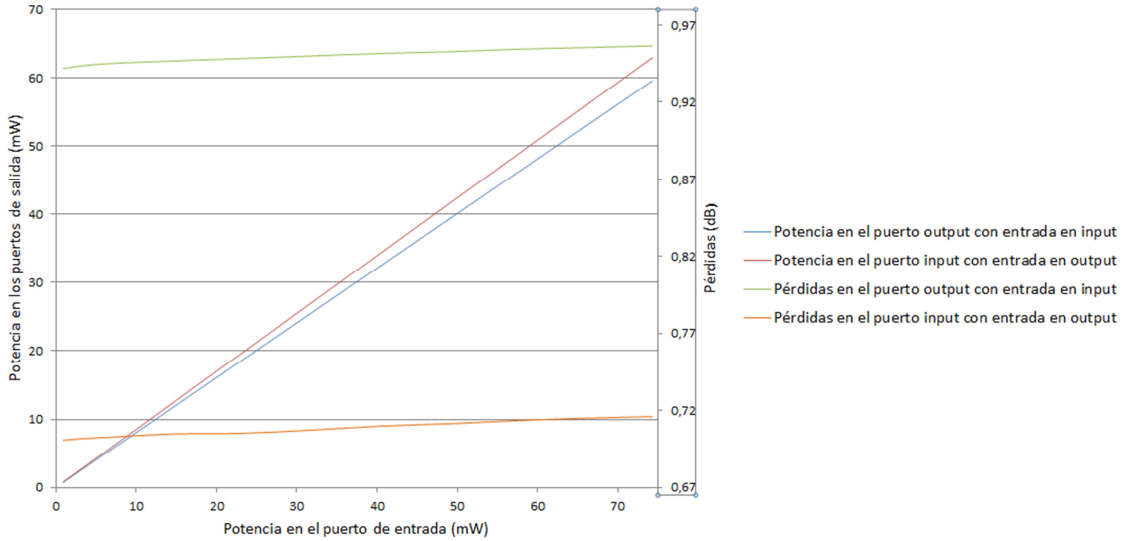


Figura 12. Análisis de potencia y pérdidas en las puertas del VOA630-FC.

4.7. Latiguillo de fibra óptica P1-630A-FC-2

En el camino de realimentación es preciso que exista un retardo lo suficientemente alto para que la realimentación resulte efectiva. Para probar diferentes retardos será necesario añadir cualquier latiguillo con una longitud aceptable en el camino de vuelta, empleando en esta ocasión uno de dos metros que pueda trabajar a la longitud de onda deseada. Merece la pena destacar como este latiguillo al final no ha sido empleado en el circuito de realimentación final ya que nos excedíamos en el retardo logrado, obteniendo resultados no satisfactorios. A pesar de ello en esta pruebas optamos por el P1-630A-FC-2 fabricado por ThorLabs, encontrándonos con características principales una banda de operación comprendida entre 633 y 780 nm, una longitud de onda de corte entre 500 y 600 nm, una atenuación de ≤15 dB/km operando en 633 nm, unas pérdidas por inserción típicas de 2dB a 633 nm, unas pérdidas de retorno típicas de 50dB (40 dB de mínimo) y dos extremos conectores FC/PC [11].

Figura 13. Latiguillo de fibra óptica P1-630A-FC-2.

A continuación se analizan las pérdidas presentes en los diversos experimentos, pudiendo comprobar como en la *Figura 14* las pérdidas permanecen bastante constantes una vez superada la transmisión de los 2mW de potencia en la salida del láser. A bajas potencias de nuevo cobran protagonismo diferentes tipos de ruido que aumentan la potencia total en la salida y haga que las pérdidas sean menores.

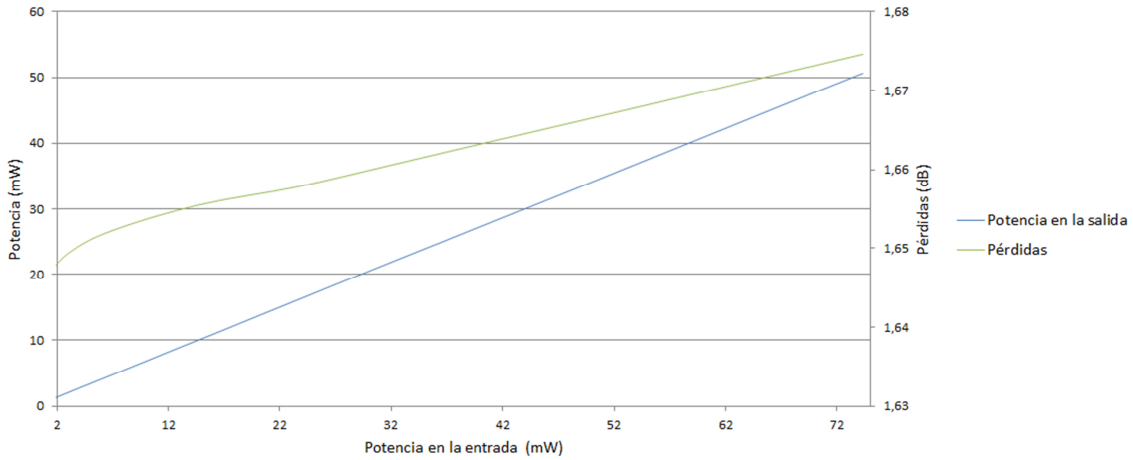


Figura 14. Análisis de potencia y pérdidas del P1-630A-FC-2.

4.8 Conectores ADAFC3 y ADAFC1

Resulta imprescindible conectar entre sí los puertos de los diferentes componentes. Esta tarea se llevará a cabo a través de los conectores, unos elementos metálicos que unen y alinean físicamente los caminos de las señales luminosas. En este caso hemos empleado el conector ADAFC3 para conectar puertos que sean APC, mientras que el conector ADAFC1 se ha empleado para los puertos PC. Estos conectores presentan unas pérdidas de inserción menores de 0,5 dB a 635 nm y en el caso de que se empleasen en fibras ópticas que mantienen la polarización, serían mayores de 1 dB. Sin embargo en nuestro caso, este segundo apartado no se contempla [12].

Figura 15a. Conector ADAFC3. Figura 16b. Conector ADAFC1.

5. Circuito de realimentación

El diseño del circuito de realimentación se ha llevado a cabo teniendo en cuenta la necesidad de que regrese un nivel mínimo de potencia a la cavidad del láser. A pesar de no existir una regla fija ni método de cálculo para determinar el porcentaje de realimentación necesario respecto al nivel de potencia emitido por el láser, diversos artículos como el publicado por J. Ohtsubo [13] distinguen diferentes regímenes según el porcentaje de realimentación recibido. Debido a ello se toma como referencia la última de ellos, la número V, ya que indica como las cavidades tanto externas como internas del láser oscilan en modo único, produciendo un comportamiento caótico importante. Para trabajar en este régimen, el porcentaje de potencia en la realimentación debe de estar en torno al 10%. De este modo, todos los esfuerzos se concentrarán en conseguir una fuerte realimentación que nos permita rondar este umbral. Merece la pena destacar, que la potencia real que llega en la realimentación al láser resulta realmente desconocida, ya que no puede ser medida, pero sí que se podrá hacer una buena estimación teniendo en cuenta las pérdidas existentes en los componentes que separan el punto de la medición y la salida del láser.

Por su parte, en otros artículos como el de N. Oliver, M. C. Soriano, I. Fischer y D. W. Sukow [14] realizan sus pruebas con porcentajes de potencia de realimentación muy superiores, por encima del 20,2% en todos los casos. Sin embargo, los resultados desfavorables obtenidos en los tests de comprobación de aleatoriedad de NIST en algunos casos no se deben a problemas relacionados con una realimentación elevada, sino a la forma de adquirir los datos relacionaos con el número de bits. Hay que tener en cuenta que porcentajes de realimentación muy altos pueden dañar el láser, por lo que el objetivo deberá ser obtener indicios de comportamiento caótico con una realimentación menor.

Dentro de todos los diferentes circuitos que se pueden proponer en la realimentación, podemos destacar varios. En todos ellos habría que conectar la salida del láser a uno de los puertos de entrada de un divisor 2x2, disponiendo de este modo de puertos suficientes para realizar las conexiones con otros elementos y poder tomar medidas en uno de ellos. En un primer escenario posible, uno de los puertos de salida de este divisor se conectaría a un espejo de Faraday, capaz de redistribuir la polarización de la luz que llega a él. Otra propuesta posible consistiría que uno de los puertos de salida del divisor se encuentre conectado a un latiguillo reflectante para que pueda reflejar de vuelta un porcentaje de potencia del 50%. A mayores, otra propuesta podría ser la de emplear un rotador en uno de los puertos de salida del divisor, conectando su salida a un latiguillo que mantiene la polarización. Posteriormente la salida de

este latiguillo estaría conectada al otro puerto del divisor 2x2 para que regrese la luz con el cambio de polarización. Para poder medir la potencia que retorna, sería preciso conectar el detector al puerto de entrada libre del divisor en todos los casos.

Todos estos experimentos propuestos no proporcionan el porcentaje mínimo de realimentación necesario, optando por emplear un método más sencillo y eficiente consistente en emplear un divisor 2x2 con sus dos puertos de salida conectados con un latiguillo que en ese caso será un atenuador variable, logrando de este modo un lazo de realimentación. En este circuito presentado, para una atenuación mínima obtenemos una realimentación aproximada del 12,5%. Hay que tener en cuenta que este porcentaje ha sido obtenido mediante una medición en el brazo inferior izquierdo del divisor, por lo que la potencia real que llegaría al láser sería menor. Mediante una estimación consistente en 1,1 dBs de pérdidas adicionales debido a la existencia de un conector entre la fibra del láser y el latiguillo conversor, a mayores de las pérdidas de ambos latiguillos, el porcentaje de potencia realimentado real sería aproximadamente de un 9,7%. Lograr al menos este porcentaje de realimentación es posible solo con los elementos disponibles en el laboratorio para el láser que emite a 638 nm, por lo que se escogió esta fuente en lugar del láser trabajando a 1550 nm.

Otro de los parámetros importantes a la hora de lograr un circuito de realimentación útil para nuestro propósito es el de lograr un retardo lo suficientemente alto para que las dinámicas del láser puedan presentar un comportamiento caótico. Por lo general, estos tiempos se suelen contabilizar como el total correspondiente a lo que tarda en la potencia emitida del láser en regresar, conociéndose como el *round trip delay time*. En artículos como el realizado por N. Oliver, M. C. Soriano, I. Fischer y D. W. Sukow [14] el tiempo indicado es de 90,9 ns. Sin embargo el retardo logrado en nuestro circuito acabará siendo menor, ya que al introducir algún latiguillo a mayores se observa como las pérdidas adiciones introducidas por los conectores ponían en peligro el alcanzar el porcentaje mínimo de realimentación. En total el circuito final presenta un retardo de 45,75 ns, correspondiente al tiempo que tarda la luz en recorrer 9,35 m de fibra óptica. También es importante destacar que en este tiempo también contribuye un latiguillo conversor APC FC a PC FC, ya que la fibra de salida del láser empleado es APC y el resto del sistema es PC FC.

En la *Figura 16* se puede comprobar el montaje final del sistema, siendo preciso indicar como la realimentación del láser sigue dos caminos. Por un lado la luz del brazo superior izquierdo del divisor según se observa en la *Figura 16*, pasa a través del brazo superior del atenuador variable, regresando al láser a través del brazo inferior del mencionado atenuador.

También ocurre lo contrario, es decir, la luz también atraviesa el brazo inferior del atenuador y regresa al láser a través de su brazo superior, ya que estamos ante un atenuador variable bidireccional, tal y como se ha comprobado en el *Capítulo 4*. El hecho de no existir ningún aislador en el laboratorio que pueda trabajar a 638 nm, provoca que haya dos caminos posibles de realimentación, produciéndose una interferencia en el centro del divisor.

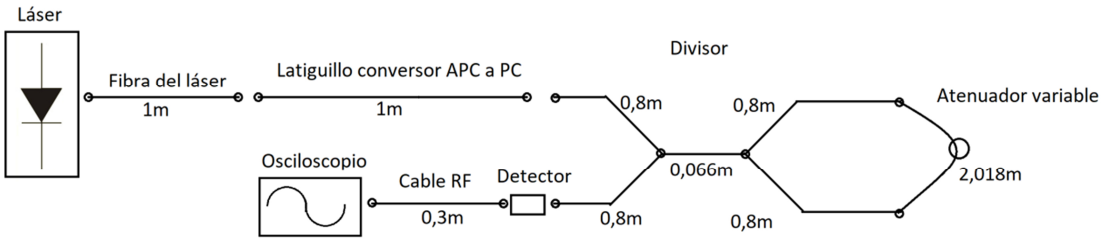


Figura 16. Configuración del circuito de realimentación.

6. Atenuación variable en el circuito de realimentación

En el camino de realimentación se ha introducido un atenuador variable con el objetivo de comprobar cómo afecta al comportamiento caótico del láser la modificación de la cantidad de potencia que retorna a él. Por lo tanto, el objetivo de este capítulo es poder encontrar un determinado valor de atenuación que provoque que el láser presente una mayor emisión de potencia respecto a otros valores de atenuación.

Ahora bien, resulta fundamental poder cuantificar la atenuación introducida, ya que el atenuador variable empleado simplemente presenta un pequeño tornillo manual que se mueve entre una posición de atenuación mínima y una posición de atenuación máxima. Para ello se llevará a cabo un experimento donde es necesario contar con un generador de funciones que pueda modular la salida del láser, consiguiendo de esta forma una señal con forma de pulso.

En primer lugar será necesario fijar un valor máximo de potencia a través de la emisión de una señal continua con atenuación mínima, es decir, girando el tornillo del atenuador hacia uno de sus topes. Esta será la referencia que se tomará para medir la variación de potencia emitida por el láser según la realimentación. Posteriormente, sin girar el tornillo, se emitirá en el láser un pulso de un período y tiempo en alta adecuados, de forma que se pueda observar en el osciloscopio el valor del voltaje de dicho pulso.

A la hora de fijar el período y el tiempo en alta de dicho pulso, habrá que tener en cuenta el tiempo que tarda el pulso desde que es emitido por el láser hasta que regresa a él. Esta cantidad de tiempo se puede obtener midiendo la distancia que recorrerá el pulso hallando su relación temporal, correspondiéndose a 45,75 ns. De esta forma, un pulso de período 80 ns con un tiempo en alta de 15 ns, será válido para su correcta observación en el osciloscopio, logrando que no se solape el pulso emitido por el láser y el de retorno a él. Obteniendo el cociente respecto a estos dos pulsos podremos conocer la atenuación introducida, expresada en el eje de abscisas de la *Figura 17*. Por su parte, si se emite una señal de potencia continua y se realiza su cociente respecto a la señal continua de atenuación mínima, se obtendrá la variación de potencia emitida por el láser según la realimentación, correspondiéndose al eje de las ordenadas de la gráfica.

Si se continúa realizando el proceso, es decir, se sigue girando el tornillo hacia el tope opuesto de forma gradual, se tiene un nuevo nivel de continua que se comparará de nuevo con el máximo fijado. Para conocer la atenuación correspondiente a este nuevo giro del

tornillo, se repetirá de nuevo el proceso de emisión de pulsos de período 80 ns con un tiempo en alta de 15 ns, comparando el emitido y el reflejado. Repitiendo el proceso a través de las diversas posiciones del tornillo hasta llegar a la atenuación máxima, se obtiene una gráfica como la mostrada en la *Figura 17*.

Hay que tener en cuenta que en el proceso de emisión de pulsos a través de la modulación externa del láser, el osciloscopio necesita una señal de trigger sincronizada con la señal de modulación para poder observarla de forma correcta. En la *Figura 17* se puede observar como para una atenuación en la realimentación aproximadamente de 2,5 dB, la potencia del láser aumenta sensiblemente, pudiendo determinar que esta es la zona en la que se intuye que puede haber un mayor comportamiento caótico.

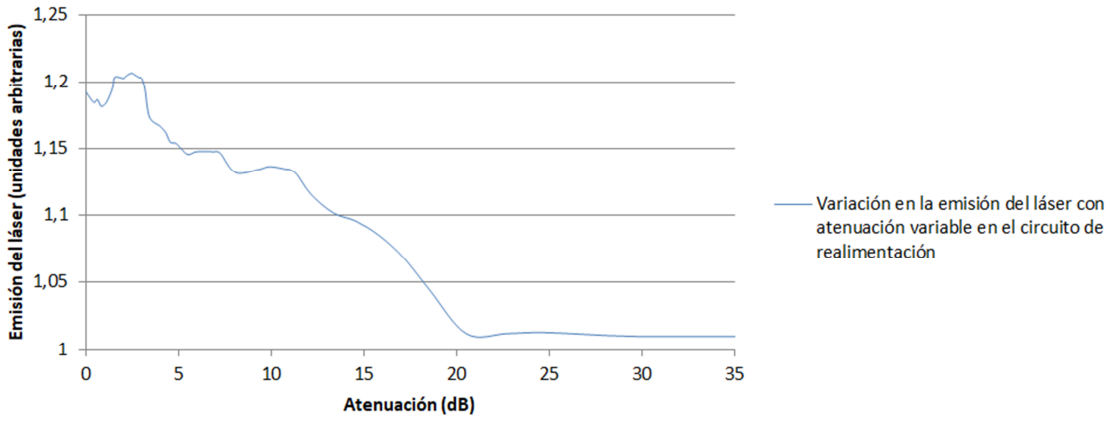


Figura 17. Potencia emitida por el láser según diferentes niveles de atenuación en su realimentación.

7. Adquisición de muestras

Para lograr que las muestras adquiridas relativas a las variaciones del voltaje capturadas con el osciloscopio sean el correcto punto de partida del generador de número aleatorios, es preciso realizar un proceso secuencial mediante el cual se puedan controlar todos los parámetros configurables del osciloscopio. Debido a ello este capítulo se centrará en la elaboración de un código en Python que ayude a implementar esta tarea, reduciendo al mínimo la iteración del usuario mientras el equipo se encuentra adquiriendo datos.

Dentro de este proceso habrá que tener muy en cuenta la correcta calibración del osciloscopio, ya que ello podría llevarnos a medidas erróneas del voltaje. En este proceso de calibración influye un factor fundamental que es la temperatura, de modo que si esta varía un mínimo de 3 ºC será preciso volver a calibrar el equipo. Del mismo modo, la función de autocalibración que posee el propio osciloscopio, precisa de un intervalo de tiempo en el que no tiene que haber ninguna señal conectada a su panel frontal, por lo que este será el punto de partida del código de adquisición. Como se puede comprobar en el fragmento de código de la Figura 18, el usuario deberá indicar al sistema cuando ha desconectado la entrada y cuando la ha vuelto a conectar para que se inicie de forma automática el proceso de captura de muestras.

```
print("Desconecte las entradas del panel frontal para iniciar la calibración.
Introduzca 1 cuando lo haya hecho" +"\n")
ok = input()
if not ok=="1":
 sys.exit(0)
COMRCW = win32com.client.Dispatch("PicoScope9000.COMRC")
COMRCW.ExecCommand("Gui:RemoteLocal")
COMRCW.ExecCommand("Ch1:Acquire 1")
COMRCW.ExecCommand("Ch2:Acquire 0")
COMRCW.ExecCommand("Gui:RemoteLocal")
COMRCW.ExecCommand("Flash:Calibr:AutocalTB")
COMRCW.ExecCommand("Flash:Calibr:AutocalCh")
resultado=COMRCW.ExecCommand("Flash:Calibr:AutocalResult?")
resultado=resultado[28]
time.sleep(5)
if not resultado == "0":
  sys.exit(0)
if resultado == "0":
 print("Por favor, conecta la entrada. Introduzca 1 cuando lo haya hecho" +"\n")
 ok = input()
 if not ok=="1":
  sys.exit(0)
```

Figura 18. Código correspondiente a la autocalibración del sistema e interacción con el usuario.

Un aspecto importante a tener en cuenta en la adquisición de datos es conocer el límite del período de muestreo mínimo con el que es posible trabajar. Este límite viene impuesto con el conversor analógico digital del osciloscopio, necesitando recurrir a su hoja de especificaciones y consultar la velocidad de adquisición, que en este caso se corresponde a

200KS/s. A partir del inverso de esta cantidad que es de 5µs podremos muestrear a tiempos superiores, siendo capaces de ralizar de esta forma experimentos a diversas tasas de muestreo y comprobar que es lo que ocurre. Como se puede comprobar en el cálculo de la tasa de muestreo $T_{muestreo}$ *= 10 unidades horizontales * base de tiempos / longitud del buffer,* influyen diversos parámetros configurables del osciloscopio que tenemos que tener en cuenta.

Una vez fijado el tiempo de muestreo es momento de describir el mecanismo por el cual las muestras que se van adquirieron se almacenan en un fichero único para su posterior procesado. En el caso de nuestro osciloscopio es preciso destacar cómo va adquiriendo muestras, exportándolas varios ficheros (en este caso se llama *SenalGrabadax*, donde *x* representa el número de fichero generado), cada uno de los cuales se corresponde a una captura de tamaño la longitud del buffer fijado. Debido a ello será necesario fijar el número de iteraciones durante las cuales se producirá este proceso, de modo que el número de muestras que se deseen obtener en total, se corresponderá al número de iteraciones multiplicado por la longitud del buffer fijada.

También es importante destacar la necesidad de activar diversas opciones de visualización como solo mostrar el canal 1 que es donde está configurada la adquisición, al mismo tiempo de desactivar el canal 2 y mostrar la salida en formato vector para que la información no se muestre discontinua. Por último, es importante destacar que la adquisición tiene que ser de tipo “single” para que no exista ningún tipo de promediado entre muestras adyacentes.

```
COMRCW.ExecCommand("TB:ScaleA 10e-4")
COMRCW.ExecCommand("Ch1:Display 1")
COMRCW.ExecCommand("Ch2:Display 0")
COMRCW.ExecCommand("Acq:FitTo Single")
COMRCW.ExecCommand("Display:Ch1:Style Vectors")
i = 1
cont = 1;
COMRCW.ExecCommand("Acq:FitTo Single")
COMRCW.ExecCommand("Acq:Ch1:RecLen 1024")
COMRCW.ExecCommand("Save:Memo:Source Ch1")
COMRCW.ExecCommand("Save:Disk:NameMode Manual")
COMRCW.ExecCommand("Save:Disk:FileFormat YOnly")
while i <=1000:
 COMRCW.ExecCommand("Save:Disk:FileName D:datos
 \muestras\SenalGrabada"+str(i)+".txty
 COMRCW.ExecCommand("Save:Disk:Save")
 i += 1
```

Figura 19. Código correspondiente a la adquisición y grabado de la señal en diferentes ficheros.

La última fase de este proceso será la concatenación de los diferentes ficheros generados en uno solo, de modo que pueda ser mucho más manejable el posterior manejo de las diferentes muestras de voltaje obtenidas. Hay que destacar que en este proceso de

concatenación, es importante introducir un control de excepciones que nos permita seguir concatenando ficheros aunque alguno de ellos por el motivo que sea no esté presente o esté corrompido. De esta forma este fichero dañado o ausente de descartará, pudiendo seguir el proceso de concatenación. A mayores se puede añadir un mensaje informativo por pantalla explicando cual es el fichero que no se ha podido concatenar para tenerlo en cuenta.

```
s2 = r'D:datos\datos.txt'
f2 = open(s2,'w')
while i <=1000:
 try:
  s1 = r"D:datos\muestras\SENALGRABADA"+str(i)+".txty"
  f1= open(s1,'r')
 except OSError:
  i += 1
 else:
  f2 = open (s2,'a')
  contenido=f1.read()
  f2.write(contenido)
  i += 1
f2.close()
f1.close()
```

Figura 20. Código correspondiente a la concatenación de ficheros de una misma captura.

8. Análisis de datos: histograma

Una vez adquiridos los datos, llega el momento de comprobar cómo estos se encuentran distribuidos, obteniendo información muy valiosa acerca de cuáles pueden ser las amplitudes de voltaje más frecuentes del mismo modo que poder obtener información acerca de la media y varianza que poseen nuestros datos. Para ello se va a emplear el análisis mediante histogramas. Durante los primeros experimentos realizados, se ha comprobado que los datos podían seguir una distribución gaussiana, por lo que hemos ajustado los datos de nuestro histograma a los propios que presentaría uno que siguiese una distribución gaussiana según la media y varianza obtenida a partir de los datos experimentales. Esto se puede lograr fácilmente mediante el comando de Matlab *histfit*, obteniendo unos valores esperados de los parámetros μ y σ de los datos experimentales con el comando *fitdist*.

Uno de los objetivos principales a la hora de emplear esta herramienta es comprobar como varía el histograma cuando el láser se encuentra emitiendo de forma directa, es decir, sin circuito de realimentación, a cuando el láser se encuentra conectado al circuito de realimentación descrito en el *Capítulo 5*. En este análisis tenemos que tener en cuenta que la media de voltaje obtenida a la salida en ambos experimentos debe de encontrarse cercana para que los datos arrojados por la varianza también puedan ser equiparables. En los diversos experimentos llevados a cabo, se han empleado valores de potencia de salida del láser comprendidos entre 35 mW y 80 mW para encontrarnos en una región donde las pérdidas de los elementos empleados resultan constantes, colocando un atenuador a la entrada del detector para no dañarlo. De esta forma, los experimentos realizados nos mostrarán en el osciloscopio una media de unos 60 mV.

En las *Figuras 21 y 22* se comparan dos histogramas con 2560000 muestras cada uno, obtenidas tanto por el láser no realimentado como por él realimentado con un período de muestreo de 39 μs. Se puede comprobar como la varianza crece significativamente cuando el láser resulta realimentado a cuando no lo está, pasando de una varianza de 3,7 mV a una correspondiente a los 11 mV. Otro dato destacado reside en cómo de distribuyen los valores de voltaje en torno a la media, encontrándonos como claramente en el histograma del láser sin realimentar hay más valores cercanos a la media, encontrándonos con bins entorno a la media que superan las 200000 muestras, mientras que en el láser realimentado sus bins no llegan a superar las 160000 muestras. Del mismo modo, si analizamos el número de valores diferentes que contiene el vector de muestras del láser sin realimentar, obtendremos 1210 frente a los 1679 que posee el del láser realimentado.

En el análisis de los histogramas también hay que tener en cuenta la existencia de ruido shot en el láser y ruido térmico en el detector, existiendo la necesidad de comprobar su el aumento de la variancia cuando se realimenta el láser se produce debido a la presencia del mencionado ruido shot. Si fuese así, la desviación estándar, es decir, σ^2 se correspondería con la media, algo que a partir de la información proporcionada por la *Figura 22* se comprueba que no ocurre, obteniendo un valor σ^2 de 0,000014225 frente a 0,0632408 que tiene como valor μ [15].

La información arrojada tanto por el aumento de la varianza y la mayor existencia de diferentes niveles de voltaje en el láser realimentado sirven para justificar el empleo de realimentación para acercarse al propósito de generar números aleatorios y detectar indicios de caos. Si atendemos al ajuste realizado respecto a las gaussianas teóricas, se puede comprobar que la distribución de las muestras se asemeja a la de una gaussiana, por lo que los datos obtenidos respecto a μ y σ son bastante fiables. También hay que tener en cuenta que el histograma propio de una distribución gaussiana se corresponde con el de una variable aleatoria.

Otro aspecto que merece la pena ser objeto de análisis reside en cómo afectan los diferentes períodos de muestreo a la forma del histograma obtenido. Volviendo a recordar según la ecuación del período de muestreo, $T_{muestreo}$ *= 10 unidades horizontales * base de tiempos / longitud del buffer,* como afecta la base de tiempos y la longitud del buffer de las muestras almacenadas al período de muestreo, se han llevado a cabo diferentes experimentos, respetando siempre el período mínimo impuesto por el conversor analógico digital del osciloscopio que es de 5 µs. Los resultados obtenidos mediante múltiples experimentos repetidos para diferentes tasas de muestreo de 19,5 µs, 31,25 µs, 39 µs, 1,22 ms y 3,9 ms muestran resultados similares, de modo que nuestro sistema acepta un amplio rango de períodos de muestreo.

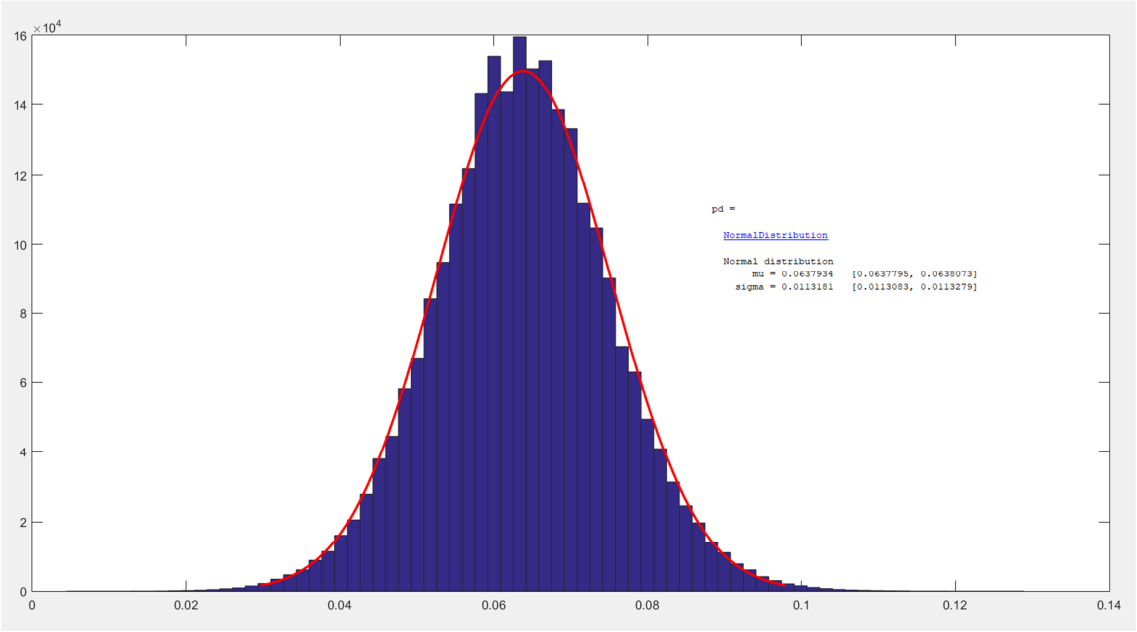


Figura 21. Histograma de 2560000 muestras obtenidas con un período de muestreo de 39 µs con un láser emitiendo sin circuito de realimentación y el ajuste realizado respecto a una gaussiana de parámetros µ=0,0637934 y σ=0,0113181.

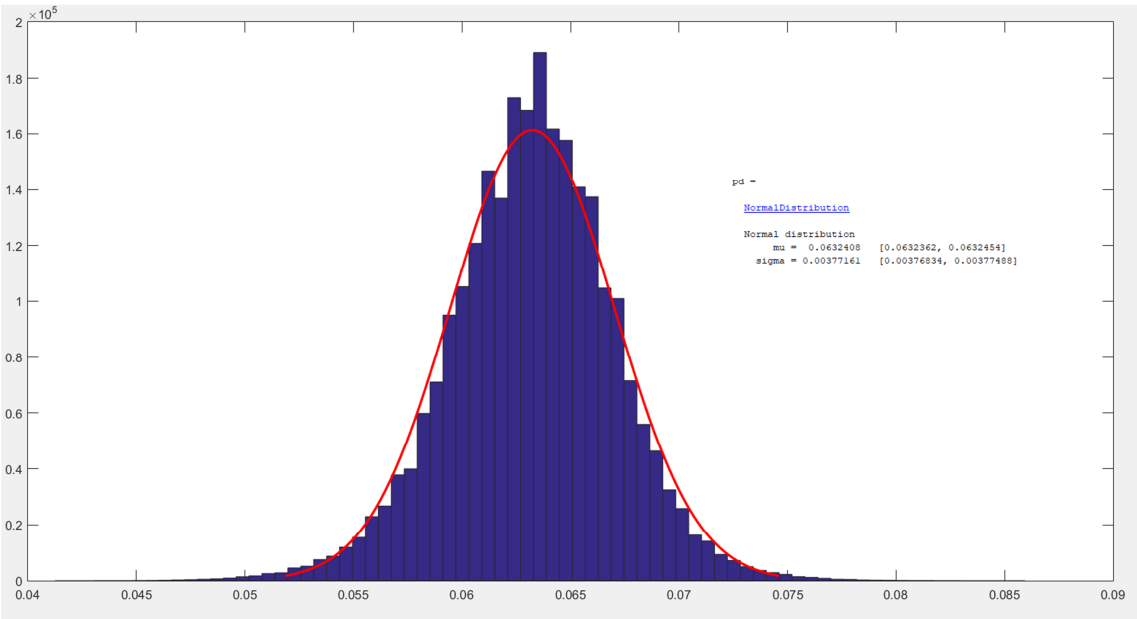


Figura 22. Histograma de 2560000 muestras obtenidas con un período de muestreo de 39 µs con un láser emitiendo sin circuito de realimentación y el ajuste realizado respecto a una gaussiana de parámetros µ=0,0632408 y σ=0,00377161.

9. Análisis de indicios de caos

Una vez se han adquirido y guardado los datos en un fichero único, es momento de estudiar si el circuito realimentado del *Capítulo 3* produce una señal que pueda considerarse caótica o no. Las herramientas que se emplearán para determinar estos indicios de caos serán el exponente de Lyapunov, la dimensión de correlación y el exponente de Hurst, todas ellas obtenidas a partir de la biblioteca de Python *Nolds* [16]. La idea es ir comprobando si los datos obtenidos cuando el láser no cuenta con realimentación, como cuando el láser cuenta con ella, cumplen las condiciones que presuponen los indicios de existencia de caos. De este modo, se explicará brevemente las herramientas empleadas y los resultados obtenidos.

9.1 Exponentes de Lyapunov

En los sistemas periódicos la trayectoria de dos partículas diverge de una forma muy lenta. Sin embargo, si hablamos de trayectoria caóticas la separación entre sus trayectorias crece de una forma muy rápida. La propia media del exponente de crecimiento de esta divergencia que existe en los sistemas que se presuponen caóticos es lo que se conoce como el exponente de Lyapunov. Si este exponente es positivo esto nos indicará que existe una divergencia importante de dos trayectorias cercanas, lo que es lo mismo, se producirá caos. En sistemas disipativos se busca justamente lo contrario, lograr un exponente de Lyapunov negativo que refleje la estabilidad del sistema en un punto fijo [17]. Existen varios algoritmos para calcular el máximo exponente de Lyapunov, encontrando en la herramienta *Nolds* dos de ellos como son el de Rosenstein y el de Eckmann.

Comenzando con el análisis del algoritmo de Rosenstein [18], nos va a proporcionar el máximo exponente de Lyapunov posible, siendo este algoritmo de entrada más sensible a los parámetros introducidos. En este caso el parámetro introducido a mayores de los datos será el de la dimensión de embebimiento, es decir, el número de dimensiones en las que el algoritmo buscará vecinos próximos respecto a cada dato para analizar sus trayectorias. También hay que tener en cuenta que la memoria del equipo empleado a la hora de hacer el experimento es limitada, por lo que se tendrá que ajustar la cantidad de datos a la máxima que nos permita un tiempo de procesamiento razonable. En el caso del equipo usado, se comprueba que tomando aproximadamente 6000 muestras equiespaciadas con una dimensión de embebimiento de 30, el tiempo de procesamiento es razonable, del orden de los 5 minutos. Cuantas más muestras se puedan analizar y más alta sea la dimensión de embebimiento, más preciso será el resultado del análisis.

Ahora bien, si se analizan 6400 muestras obtenidas directamente con el láser sin realimentación bajo las condiciones expuestas anteriormente, se obtiene un exponente de -0.0003027125824632358, confirmando como el comportamiento de las muestras no nos da indicios de caos. Por su parte, si hacemos lo mismo con 6400 muestras del láser con realimentación, el exponente obtenido es de 0.0013515473450215862, lo que serviría para indicar indicios de caos al ser el exponente positivo, aunque el valor absoluto resulta muy bajo.

El segundo algoritmo proporcionado por *Nolds* para determinar el exponente de Lyapunov es el de Eckmann [18]. Este algoritmo nos permite que si el sistema tiene más de una variable libre, el diagrama de fases tiene más de una dimensión, por lo que cada exponente de Lyapunov se corresponderá a cada una de estas dimensiones. Al igual que en el algoritmo anterior, es preciso fijar unas condiciones que permitan al equipo realizar la tarea en un tiempo razonable. En este caso, tomando aproximadamente unas 6000 muestras, empleando una dimensión de embebimiento de 30 y obteniendo 2 exponentes, el tiempo de procesamiento es del orden de los 3 minutos. Hay que precisar que el algoritmo exige que el número de dimensiones embebidas -1 entre el número de exponentes -1 nos proporcione un cociente entero. Bajo estas condiciones, 6400 muestras equiespaciadas, obtenidas directamente del láser sin realimentar, obtienen dos exponentes de -0.00033748 -0.02254014. Por su parte, si hacemos lo mismo con 5100 muestras del láser realimentado, se obtienen unos exponentes de valor -0.0005034 Y -0.0218882, de modo que este algoritmo no verifica nuestra hipótesis de comportamiento caótico del láser realimentado.

9.2 Dimensión de correlación

Como se ha mencionado anteriormente, la evolución de las dinámicas de un sistema caótico manifiestan una fuerte dependencia según las condiciones iniciales. Este comportamiento extraño en el tiempo tiene su origen en la geometría del diagrama de fases, formado por las trayectorias del sistema que se conocen como el atractor [19]. Para poder medir su complejidad, se emplean las dimensiones fractales y dentro de ellas se encuentra la dimensión de correlación, calculada a través del algoritmo Grassberger-Proccacia [20]. Cuanto más grande sea esta dimensión más complejo será el sistema, teniendo en cuenta que un sistema periódico tendrá como dimensión de correlación el valor de la unidad. Sin embargo, para hacerse una idea, uno de los atractores más clásicos es el de Lorentz, contando con una dimensión de correlación de 2,05 [21]. A través de la herramienta de *Nolds* que calcula la dimensión de correlación a través del algoritmo de Grassberger-Procaccia, se podrá saber si nuestros datos poseen un atractor y por lo tanto un cierto comportamiento caótico.

De igual forma que se ha realizado anteriormente, se procede a comparar los datos obtenidos del láser sin realimentación con los del láser realimentado, comprobando si estos resultados se acercan a la dimensión de correlación del atractor de Lorentz. Empleando en ambos casos un número de muestras de 6400 y una dimensión de embebimiento de 3, obtenemos resultados muy similares entre nuestros dos escenarios. La dimensión de correlación del láser realimentado es de 2,4545234320409013 y la del láser sin realimentar es de 2,52412657, lo que nos indica que en principio ambos tendrían atractores.

También hay que tener en cuenta que el ruido contribuye a aumentar la dimensión de correlación, algo que también debería ser tenido en cuenta en las *Figuras 23 y 24*. En ellas podemos comprobar el ajuste lineal de la suma de correlación. Esta suma de correlación representa las fracciones de pares de puntos en el diagrama de fases cuya distancia es menor que r. Como se puede comprobar, el ajuste realizado es más preciso en la gráfica del láser sin realimentar. Por lo tanto, a través de esta herramienta de análisis se arrojan resultados que nos indican como las dinámicas del el láser realimentado como el láser realimentar presentarían en principio un comportamiento caótico.

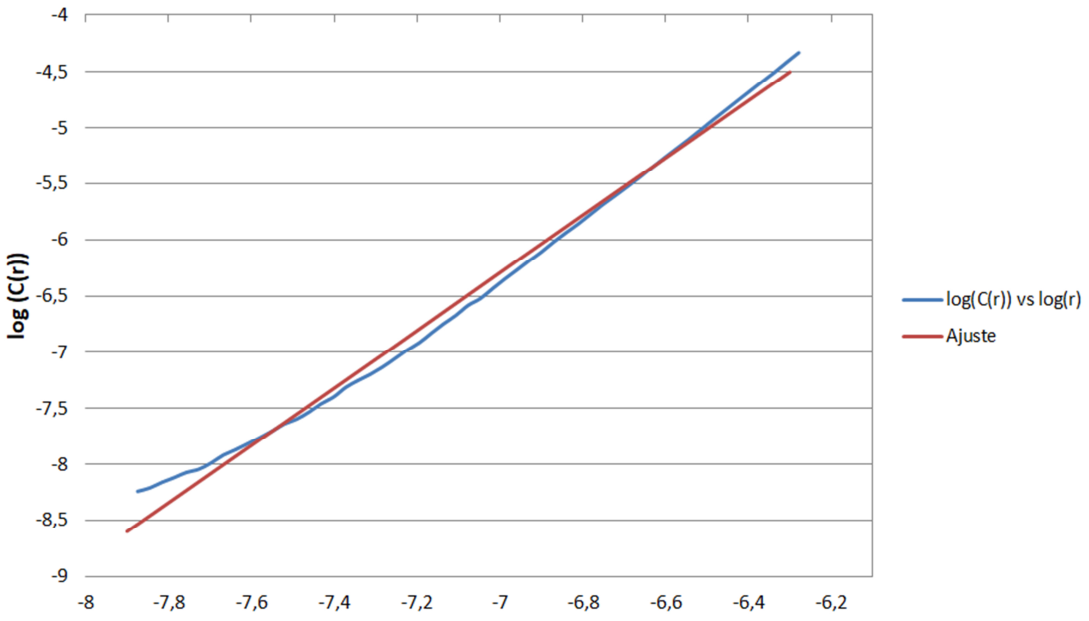


Figura 23. Suma de correlación C(r) obtenida de forma experimental y su ajuste lineal del láser no realimentado.

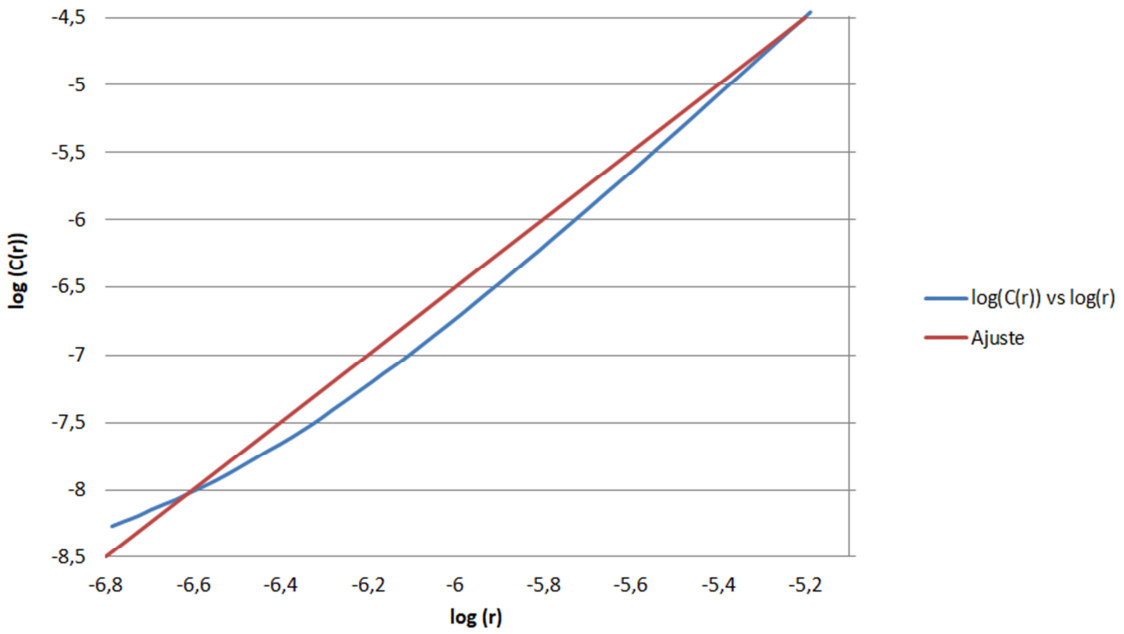


Figura 24. Suma de correlación C(r) obtenida de forma experimental y su ajuste lineal del láser realimentado.

9.3 Exponente de Hurst

El exponente de Hurst es otro potente indicador de existencia de indicios de caos. Este exponente posee un valor entre 0 y 1 e indica una medida de la memoria a largo plazo del sistema. Si este exponente se encuentra entre 0 y 0,5, se puede decir que las series que se extraigan del sistema no muestran dependencia. Merece la pena destacar también como el exponente de Hurst se encuentra relacionado con la dimensión fractal (capacidad) mediante la relación H=2-D [22]. Sometiendo al análisis nuestras muestras, al igual que en las anteriores herramientas de análisis del caos, solo que en este caso pudiendo tomar 64000 muestras debido a la rapidez del algoritmo, obtendremos un exponente de Hurst de 0.5202838466969547 para el láser sin realimentar, mientras que para el láser realimentado obtendremos un exponente de 0.4914437780913453 para el láser realimentado. De esta forma esta herramienta nos ofrece una buena justificación del empleo de un circuito de realimentación.

9.4 Resumen del análisis de indicios de caos

En la *Tabla 1* se puede comprar en conjunto los resultados arrojados por las diferentes herramientas de análisis de indicios de caos sobre los datos del láser sin realimentar y el láser realimentado, encontrando cómo los indicios de caos resultan mucho mayores en el láser realimentado. Solo se encuentra un resultado favorable en el láser sin realimentar, mientras

que en el realimentado solo uno nos aparta de la idea inicial de existencia de comportamientos caóticos. Hay que tener en cuenta que la existencia de ruido no determinístico propicia que tanto en los datos del láser sin realimentar como en los del láser realimentado siempre sean algo más favorables a la existencia de comportamientos caóticos aunque estos realmente no tuviesen lugar. Como ya se ha mencionado anteriormente en este capítulo, no se puede asegurar solo mediante estos resultados la existencia de comportamientos caóticos en las dinámicas del láser ya que sería también interesante contar con una capacidad mayor para analizar un número de muestras más alto y poder contrastar aún más estos resultados.

Herramienta de análisis	Láser sin realimentar	Láser realimentado
Exponente de Lyapunov (algoritmo de Rosenstein)	X	V
Exponente de Lyapunov (algoritmo de Eckmann)	X	X
Dimensión de correlación	V	V
Exponente de Hurst	X	V

Tabla 1. Resumen con los resultados obtenidos en las herramientas de análisis de indicios de caos. El color verde indica que los resultados han sido satisfactorios.

10. Generación de números aleatorios

Una vez logrado un fichero con una cantidad importante de muestras de voltaje del láser realimentado, llega el momento de generar secuencias de bits que sean lo más uniformes posibles. Para llevar a cabo esta tarea, el procedimiento será de lo más sencillo, tomando directamente las muestras y realizando una sencilla operación de resta entre ellas. El algoritmo buscado se basa en tomar dos muestras consecutivas, analizar su diferencia y emplear como criterio de decisión el signo de la diferencia. Si la diferencia es positiva se obtendrá un uno, mientras que si la diferencia es negativa se obtendrá un cero, encontrándonos en la línea número 7 de la Figura 25 el mencionado criterio de decisión. De esta forma tan sencilla se van acumulando los bits hasta lograr bytes que se escriben en el fichero de salida. En la línea número 11 se puede comprobar cómo este proceso de escritura tiene lugar.

```
1  import sys
2  bitn=0
3  byte=0
4  par=0
5  for line in sys.stdin:
6      if par==1:
7         bit=(float(line)-vant)>0
8         byte=byte+int(bit)*2**bitn
9         bitn=bitn+1
10        if bitn==8:
11          sys.stdout.write(chr(byte))
12          bitn=0
13          byte=0
14     par=1-par
15     vant=float(line)
```

Figura 25. Código del algoritmo empleado para la obtención de números aleatorios a partir de las muestras del láser realimentado.

11. Análisis de la calidad del generador

Una vez se ha obtenido un fichero con suficientes bits, llega el momento de verificar que la secuencia obtenida resulta aleatoria según diversos tests. Para ello se van a emplear diversas herramientas consistentes en diversos cálculos matemáticos que podrán determinar cuánto se aproximan las secuencias producidas por el generador a las propiedades de secuencias uniformes. De esta forma se podrán comparar los resultados obtenidos sobre los ficheros generados a partir de las muestras del láser sin realimentar y realimentado. A través de *Ent, Rgntest* y *Dieharder* será posible cuantificar la aleatoriedad de los bits, exponiendo una serie de resultados que precisan de ser analizados para poder sacar conclusiones.

11.1 Ent

Ent es una potente herramienta basada en el análisis de la entropía que poseen diversas secuencias de números [23]. La entropía es una medida de la incertidumbre asociada a una variable aleatoria. Como idea original, este término fue ideado por Claude Shannon, cuantificando el valor esperado de la información contenida en un mensaje [24]. Un fichero con una gran entropía indicará que hay muy pocos patrones repetidos en él, encontrándose comprimida u optimizada la información que contiene. El valor de entropía que *Ent* proporciona está expresado en bits por byte, de modo que el ideal sería 8 bits por byte. Hay que mencionar que esta herramienta proporciona una buena idea acerca de si se puede estar ante un generador de secuencias uniformes pero tampoco sirve para confirmarlo por completo.

Entrando en el análisis de los resultados obtenidos mediante *Ent*, se analizarán dos secuencias de 156 KBytes, correspondientes a los ficheros generados a través de las mismas muestras analizadas en los *Capítulos 8 y 9* con el láser sin realimentar y realimentado, respectivamente. Estos ficheros se han obtenido tomando tanto 2560000 muestras del láser realimentado como sin realimentar con un período de muestreo de 39 µs. Posteriormente se ha aplicado el script que contiene el algoritmo explicado en el *Capítulo 10*.

En la secuencia del láser sin realimentar, se obtiene una entropía de 7,989890 bits por byte, mientras que en la secuencia del láser realimentado la entropía ligeramente superior con 7,998945 bits por byte. Se puede decir que ambos resultados son muy positivos para seguir empleando más herramientas de análisis sobre los ficheros.

Aparte de proporcionar un valor de entropía, *Ent* a su vez se encarga de realizar otras comprobaciones de aleatoriedad. Una de ellas es la de la compresión óptima, ya que si se

ejerciese esta misma sobre un fichero, la reducción de su tamaño debería ser cero si su información fuese aleatoria, no mostrando de este modo patrones repetidos. Pues bien, los resultados arrojados por *Ent* sobre los dos ficheros analizados indican que no se pueden comprimir más.

Otra interesante función de *Ent* es la posibilidad de realizar el test χ^2, proporcionándonos un porcentaje que nos indicará si las secuencias son sospechosas de no ser uniformes. Partiendo de una hipótesis nula en relación a cómo será la distribución de un conjunto de bits, a través de este test se proporcionará una probabilidad acerca de cuanto se aproxima esa hipótesis nula a la distribución que se presupone ser. El test χ^2 implementado en *Ent* lógicamente nos proporcionará información acerca de lo que se parece la secuencia de bits a la distribución normal. De esta forma si el anteriormente mencionado porcentaje es mayor del 99% o menor del 1%, la secuencia es prácticamente no uniforme, mientras que si se encuentra entre el 99% y el 95% o entre el 1% y el 5%, la secuencia sería sospechosa de ser uniforme. Los porcentajes entre el 90% y el 95% o entre el 5% y el 10% indican que la secuencia es casi sospechosa [25]. Por nuestra parte, el fichero del láser sin realimentar arroja un porcentaje del 0,01% mientras que el realimentado es del 75%, proporcionándonos este último información positiva acerca de la uniformidad de nuestra secuencia.

Continuando con la información proporcionada por *Ent*, también se encuentra la media aritmética, que no es más que el resultado de la suma de todos los bytes divididos entre el tamaño del fichero. Si la secuencia es aleatoria, se tendría que acercar este valor a 127,5 [25]. En el análisis realizado, el fichero del láser realimentado se obtiene un valor de 127,5089, mientras que el del fichero no realimentado se obtiene 126,4988.

El penúltimo mecanismo que emplea *Ent* para medir la aleatoriedad es el test de Monte Carlo para obtener el valor pi. En esta ocasión, cada secuencia de 6 bits sucesivos se emplea para construir un cuadrado de coordenadas X e Y. Si la distancia al punto generado aleatoriamente es menor que le radio de un círculo inscrito dentro del cuadrado creado, la secuencia de 6 bits se considera válida. A través del porcentaje de secuencias válidas se obtiene el valor de pi [25]. En el fichero del láser no realimentado se obtiene pi con un error del 2,29 por ciento, mientras que en el del láser realimentado obtiene pi con un error del 0,69 por ciento. Por último, *Ent* también se encarga de obtener el coeficiente de correlación, de modo que el coeficiente ideal si la secuencia fuese aleatoria sería cero. La secuencia del láser realimentado arroja un coeficiente de -0,002105, mientras que la del láser no realimentado es de 0,009623.

11.1.1 Resumen de los resultados proporcionados por Ent

En la Tabla 2 se puede comprobar como los resultados proporcionados por *Ent* para las secuencias introducidos resultan por lo general bastante buenos. La existencia de ruido en las muestras tanto del láser sin realimentar como realimentado provocan que las secuencias se puedan aproximar a ser uniformidades, encontrándonos como único punto negro el test χ^2 para los datos del láser sin realimentar, algo que a todas luces refuerza la necesidad de emplear el circuito de realimentación para conseguir el generador de números aleatorios.

Herramienta de análisis	Láser sin realimentar	Láser realimentado
Entropía (8 bits por byte completamente uniforme)	7,989890	7,998945
χ^2 (entre el 10% y 90% para ser completamente uniforme)	0,01	75
Media aritmética (127,5 para ser completamente uniforme)	126,4988	127,5089
Test de Montecarlo para obtener pi (3,141592653 para ser completamente uniforme)	3,213530338 (error 2.29%)	3,163129078 (error 0.69 %)
Coeficiente de correlación (0 para ser completamente uniforme	0,009623	-0,002105

Tabla 2. Resumen con los resultados obtenidos en las herramientas de análisis de aleatoriedad del Ent.

11.2 Dieharder

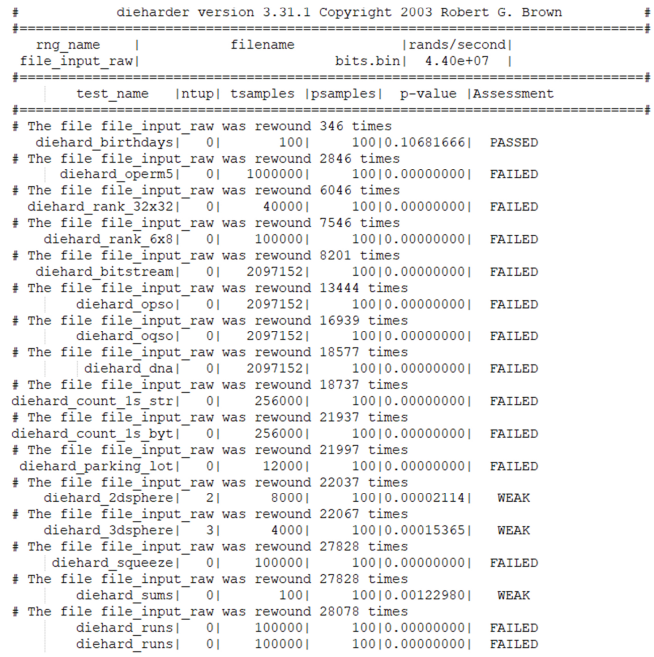
Dieharder es otra herramienta que consiste en una batería de tests que son capaces de determinar si un generador de números aleatorios es lo suficientemente robusto. Para ello sus tests cuentan con tres estados de diagnóstico como son *failed, weak* o *passed*, indicándonos si los resultados resultan o no favorables. Un detalle a tener en cuenta es que muchos de los tests investigan secuencias de bits solapadas, ya que estas secuencias no son independientes y es necesario tener en cuenta la covariancia entre las muestras [26]. Debido a esta característica entre muchas otras, es necesario precisar que a mayores cantidades de datos se obtienen resultados mucho más fiables, simplemente por el hecho de que los tests puedan realizar todas las comprobaciones necesarias. Por ejemplo, hay que tener en cuenta que con 4 GBytes de datos a la entrada, los tests tienen que volver a tomar desde el principio los datos

57 veces, para que puedan llevarse a cabo por completo con las repeticiones con los tamaños por defecto necesarios [26].

A lo largo de los test se llevan a cabo técnicas de lo más variadas para comprobar la aleatoriedad de los números. Algunos ejemplos pueden ser tomar muestras de forma aleatoria en un intervalo muy largo y que su espaciado se corresponda con una distribución exponencial, tomar cinco números consecutivos y las 120 ordenaciones posibles que se pueden lograr con ellos tienen que ser equiprobables, tratar las secuencias de bits como palabras y contar las palabras superpuestas en las secuencias o generar secuencias de números que al sumarlos a la secuencia original de cómo resultado una distribución normal [27].

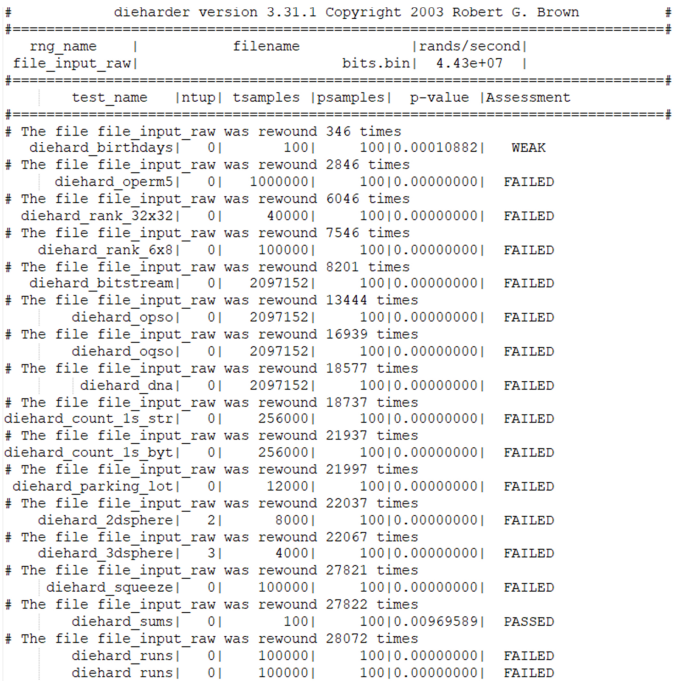
Si sometemos los mismos ficheros ficheros del láser realimentado como el no realimentado que hemos analizado con el *Ent*, nos encontramos como los resultados son bastante malos en ambos casos. Estos resultados realmente no son nada esperanzadores, pero tenemos que tener en cuenta en la información proporcionada en la salida que los datos se vuelven a leer multitud de veces, por lo que más que un problema de la aleatoriedad de los bits reside en que el tamaño de los ficheros conseguidos es bastante escaso. Esto será un problema que se comentará en los capítulos finales, ya que la cantidad de bits en la entrada resulta insuficiente.

Si se genera una secuencia del mismo tamaño con el generador de números aleatorios Quantis [28], que se empleará y explicará más detalladamente en el *Capítulo 12*, los tests resultan incluso aún más desfavorables ya que la cantidad de bits se queda escasa. Esto indica que a pesar de la calidad de los números que se presupone que es capaz de lograr Quantis, para estas cantidades de datos *Dieharder* no los va a poder validar.



```
#            dieharder version 3.31.1 Copyright 2003 Robert G. Brown          #
#===========================================================================#
   rng_name    |           filename             |rands/second|
 file_input_raw|                         bits.bin|  4.40e+07  |
#===========================================================================#
      test_name   |ntup| tsamples |psamples|  p-value |Assessment
#===========================================================================#
# The file file_input_raw was rewound 346 times
   diehard_birthdays|   0|       100|     100|0.10681666|  PASSED
# The file file_input_raw was rewound 2846 times
      diehard_operm5|   0|   1000000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 6046 times
  diehard_rank_32x32|   0|     40000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 7546 times
    diehard_rank_6x8|   0|    100000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 8201 times
   diehard_bitstream|   0|   2097152|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 13444 times
        diehard_opso|   0|   2097152|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 16939 times
        diehard_oqso|   0|   2097152|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 18577 times
         diehard_dna|   0|   2097152|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 18737 times
diehard_count_1s_str|   0|    256000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 21937 times
diehard_count_1s_byt|   0|    256000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 21997 times
 diehard_parking_lot|   0|     12000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 22037 times
    diehard_2dsphere|   2|      8000|     100|0.00002114|   WEAK
# The file file_input_raw was rewound 22067 times
    diehard_3dsphere|   3|      4000|     100|0.00015365|   WEAK
# The file file_input_raw was rewound 27828 times
     diehard_squeeze|   0|    100000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 27828 times
        diehard_sums|   0|       100|     100|0.00122980|   WEAK
# The file file_input_raw was rewound 28078 times
        diehard_runs|   0|    100000|     100|0.00000000|  FAILED
        diehard_runs|   0|    100000|     100|0.00000000|  FAILED
```

Figura 26. Resultados del test Dieharder para los bits generados por el láser sin realimentar.



```
#            dieharder version 3.31.1 Copyright 2003 Robert G. Brown          #
#===========================================================================#
   rng_name    |           filename             |rands/second|
 file_input_raw|                         bits.bin|  4.43e+07  |
#===========================================================================#
      test_name   |ntup| tsamples |psamples|  p-value |Assessment
#===========================================================================#
# The file file_input_raw was rewound 346 times
   diehard_birthdays|   0|       100|     100|0.00010882|   WEAK
# The file file_input_raw was rewound 2846 times
      diehard_operm5|   0|   1000000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 6046 times
  diehard_rank_32x32|   0|     40000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 7546 times
    diehard_rank_6x8|   0|    100000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 8201 times
   diehard_bitstream|   0|   2097152|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 13444 times
        diehard_opso|   0|   2097152|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 16939 times
        diehard_oqso|   0|   2097152|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 18577 times
         diehard_dna|   0|   2097152|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 18737 times
diehard_count_1s_str|   0|    256000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 21937 times
diehard_count_1s_byt|   0|    256000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 21997 times
 diehard_parking_lot|   0|     12000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 22037 times
    diehard_2dsphere|   2|      8000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 22067 times
    diehard_3dsphere|   3|      4000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 27821 times
     diehard_squeeze|   0|    100000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 27822 times
        diehard_sums|   0|       100|     100|0.00969589|  PASSED
# The file file_input_raw was rewound 28072 times
        diehard_runs|   0|    100000|     100|0.00000000|  FAILED
        diehard_runs|   0|    100000|     100|0.00000000|  FAILED
```

Figura 27. Resultados del test Dieharder para los bits generados por el láser realimentado.

11.3 Rgntests

La última herramienta para analizar la aleatoriedad de los números contiene tests con un gran potencial, cuyo funcionamiento a priori parece más sencillo de entender que los que presenta el *Dieharder* en ciertos casos. Algunos de estos tests pueden ser el test de la

frecuencia, basado en contar la cantidad de unos presentes en una secuencia, de modo que si es cercana a la mitad de la secuencia, el número de ceros deberá ser también cercano a la mitad, confirmando de este modo la aleatoriedad. Relacionado con este anterior, también existe otro relacionado con buscar ceros y unos dentro de bloques de M-bits, de modo que en cada bloque se espera obtener una frecuencia de ambos de la mitad. También existen algunos otros de mayor complejidad matemática como puede ser el test de la transformada discreta de Fourier, detectando patrones periódicos en el espectro frecuencial. En total son 15 tests que arrojarán más información sobre la aleatoriedad de las secuencias [29].

En el caso del análisis de los ficheros correspondientes, para el del láser realimentado se han obtenido 63 aciertos según el test de la FIPS 140-2 (estándares federales de procesamiento de la información, publicación 140-2) y ningún fallo, mientras que para el láser sin realimentar 57 aciertos y 6 fallos según el mismo test. También nos indica como el fichero del láser sin realimentar presenta 3 indicaciones en el *Runs* test, indicando como en tres subsecuencias hay un número mayor de unos consecutivos de lo que cabría esperar. De esta forma se puede comprobar como el fichero del láser realimentado presenta resultados más favorables de aleatoriedad que el del láser sin realimentar. También hay que destacar que para ficheros mayores, si la secuencia realmente se acerca a la uniformidad, se obtendrían también algún fallo según el test de la FIPS 140-2 debido al propio comportamiento de la secuencia.

```
Copyright (c) 2004 by Henrique de Moraes Holschuh
This is free software; see the source for copying conditions.  There is NO warranty;
not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
rngtest: starting FIPS tests...
rngtest: entropy source drained
rngtest: bits received from input: 1280000
rngtest: FIPS 140-2 successes: 57
rngtest: FIPS 140-2 failures: 6
rngtest: FIPS 140-2(2001-10-10) Monobit: 0
rngtest: FIPS 140-2(2001-10-10) Poker: 4
rngtest: FIPS 140-2(2001-10-10) Runs: 3
rngtest: FIPS 140-2(2001-10-10) Long run: 0
rngtest: FIPS 140-2(2001-10-10) Continuous run: 0
rngtest: input channel speed: (min=1.164; avg=5.239; max=18.626)Gibits/s
rngtest: FIPS tests speed: (min=14.181; avg=40.468; max=61.527)Mibits/s
rngtest: Program run time: 30426 microseconds
```

Figura 28. Resultados del rgntest para los bits generados por el láser sin realimentar.

```
Copyright (c) 2004 by Henrique de Moraes Holschuh
This is free software; see the source for copying conditions.  There is NO warranty;
not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
rngtest: starting FIPS tests...
rngtest: entropy source drained
rngtest: bits received from input: 1280000
rngtest: FIPS 140-2 successes: 63
rngtest: FIPS 140-2 failures: 0
rngtest: FIPS 140-2(2001-10-10) Monobit: 0
rngtest: FIPS 140-2(2001-10-10) Poker: 0
rngtest: FIPS 140-2(2001-10-10) Runs: 0
rngtest: FIPS 140-2(2001-10-10) Long run: 0
rngtest: FIPS 140-2(2001-10-10) Continuous run: 0
rngtest: input channel speed: (min=1.096; avg=5.262; max=18.626)Gibits/s
rngtest: FIPS tests speed: (min=13.922; avg=39.075; max=61.330)Mibits/s
rngtest: Program run time: 31535 microseconds
```

Figura 29. Resultados del rgntest para los bits generados por el láser realimentado.

12. Mejorando la calidad del generador: extractor

Como se ha visto, en el anterior capítulo, los bits del fichero generados a partir de las muestras del láser realimentado poseen una uniformidad. Esta uniformidad aún se puede mejorar mediante un extractor capaz de implementar un mecanismo con el que se obtiene a partir de la secuencia original de bits de entrada una salida con mayor uniformidad. Uno de los primeros en realizar este planteamiento fue Von Neumann [30], logrando obtener bits aleatorios no sesgados e independientes de una fuente de bits independientes pero sesgados, con un sesgo desconocido.

El planteamiento realizado fue bastante simple, ya que si la fuente produce unos con una probabilidad p, entonces se pueden tomar dos bits y asignar un 1 en las salida si los bits tomados son 10. Lo mismo ocurriría si se generasen ceros con una probabilidad 1-p, tomando dos bits consecutivos y asignando a la salida un 0 si los bits tomados son 01. Como se puede comprobar, para obtener un bit en la salida, es preciso tomar un número mayor de bits de la entrada, debido a que la fuente de bits de la entrada es muy débil. Del mismo modo, el extractor de Von Neuman es un ejemplo de los llamados extractores de aleatoriedad determinística, ya que para una entrada determinada, este extractor siempre proporcionará la misma salida. Por lo tanto es válido solo para eliminar sesgos, pero no imperfecciones relacionadas con la correlación de muestras [30].

En este caso se empleará un extractor mucho más sofisticado como es el proporcionado para el generador de números aleatorios Quantis [28]. Este generador de números aleatorios se aprovecha de la detección de fotones, siendo estable frente a cambios en su entorno además de permitir comprobar el estado de generación de los números aleatorios en tiempo real. Este generador posee una tasa de generación de 4 Mbits/s, indicando que las secuencias de bits producidas pasan los tests de NIST y Diehard. Su funcionamiento se basa en la detección de fotones que tienen la misma probabilidad de llegar a dos detectores diferentes, asignando 0 ó 1 según al que lleguen, siendo esta operación continuamente motorizada. El generador se encuentra acompañado por un software muy potente que es el que permitirá emplear el extractor que lleva incorporado.

La idea de este extractor, cuyo principio de funcionamiento son las funciones Hash 2-Universales [31], es obtener k bits y_i que posean una aleatoriedad mayor que los n>k bits x_i en la entrada. Si se supone que cada uno de los bits de la entrada tiene una entropía s, donde el valor 1 es el mayor posible, la probabilidad de que la secuencia de bits a la salida del extractor se desvíe de un secuencia perfectamente aleatoria será $\varepsilon_{hash}=2^{-(sn-k)/2}$. A pesar de que ε_{hash} se

puede hacer arbitrariamente pequeño, el propósito es mantener este valor entre 2^{-100} y 2^{-30}, ya que lograr un valor de cero es prácticamente imposible. Al mismo tiempo, es preciso fijar unos valores de n y k que permitan una buena reducción de ε_{hash} y que al mismo tiempo puedan garantizar que la pérdida de bits no sea muy alta. Si se tienen en cuenta las limitaciones del software de Quantis, para n=1024 y k=768 se pierden el 25% de los bits, mientras que si n=2048 y k=1792 se pierden el 12,5% [31].

La configuración del extractor se basa en una matriz aleatoria m, también llamada semilla, de modo que los bits de salida y_i se obtienen de la forma $yi = \sum_{j=1}^{n} mij * xj$. Las propiedades de esta matriz requieren que su número de columnas sea igual al número de bits del vector de entrada. El número de bits del fichero de salida será igual al número de filas de dicha matriz. Las matrices incluidas en el software pueden ser de 1024 líneas x 768 columnas, por lo que la secuencia de entrada deberá ser recortada en subsecuencias de tamaño 768 [32].

Para comprobar el funcionamiento de este extractor, se empleará la matriz *default_idq_matrix* incluida en el software. Esta matriz tiene unas dimensiones de 1024 líneas y 768 columnas. Pues bien, el extractor la aplicará a un fichero de bits de tamaño 612352 Bytes, logrado a partir del algoritmo del *Capítulo 10* actuando sobre un fichero del láser realimentado cuyas muestras ocupan un total de 313,5 MBytes. A la salida del extractor se obtendrá uno de tamaño 459264 Bytes, es decir, 25% veces menor. Como se puede comprobar en la Tabla 3, los datos del *Ent* presentan una ligera mejora, calculando ε_{hash} y obteniendo un valor de 2^{-128}, siendo este muy reducido.

	Fichero de bits en crudo	Fichero de bits obtenidos del extractor
Entropía (8 bits por byte totalmente aleatorio)	7,999684	7,999594
Chi cuadrado (entre el 10% y 90% para ser completamente aleatorio)	50,00	50,00
Media aritmética (127,5 para ser completamente aleatorio)	127,1817	127,5955
Test de Montecarlo para obtener pi (3,141592653 para ser aleatorio)	3,158478512	3,137176003
Coeficiente de correlación (0 para ser completamente aleatorio)	-0.002413	0,000117

Tabla 3. Comparativa de los resultados aportados por ent respecto a los bits en crudo y los bits obtenidos del extractor, todos ellos generados por el láser realimentado.

Ahora bien, con ficheros de bits que ya poseen una gran entropía, los resultados del uso del extractor no resultan tan llamativos, cobrando más significado cuando se emplea con ficheros de entrada cuyos bits tienen valores de entropía más bajos. Para comprobar los buenos resultados que proporciona el extractor de Quantis, se va a llevar a cabo un proceso con el que se aumentará el tamaño del fichero de entrada a costa de reducir la entropía de sus bits. Este proceso partirá del algoritmo original explicado en el *Capítulo 10*. Al igual que en el mencionado algoritmo, se realizará la diferencia entre dos muestras consecutivas, solo que en esta ocasión, la diferencia en formato *float* será convertida a caracteres tomando el número de bytes menos significativos que se desee. En el caso de la *Figura 30* se tomarían los 2 bytes menos significativos.

```
from __future__ import print_function
import sys
import struct

def float_to_bin(num):
    return format(struct.unpack('!I', struct.pack('!f', num))[0], '032b')

def bin_to_float(binary):
    return struct.unpack('!f',struct.pack('!I', int(binary, 2)))[0]

par=0
vant=0
for line in sys.stdin:
    if par==1:
       bit=(float(line)-vant)
       binaria=float_to_bin(float(bit))
       chars=[ chr(int(binaria[0+8*i:8*(i+1)],2)) for i in range(0,4)]
       print(chars[2], sep='', end='')
       print(chars[3], sep='', end='')
    par=1-par
    vant=float(line)
```

Figura 30. Algoritmo para la generación de bits pudiendo tomar el número de bytes deseado de la diferencia entre dos muestras consecutivas.

A continuación, en la *Tabla 4*, se expone una gráfica elaborada a partir de tomar uno, dos, tres o cuatro bits menos significativos de la diferencia entre muestras de un fichero de muestras del láser realimentado de tamaño total 80,2 MBytes. Si se selecciona el byte menos significativo, el fichero de bits a la entrada del extractor tendrá tamaño 1,3 MBytes, si se seleccionan los dos bytes menos significativos tendrá tamaño 2,6 MBytes, si se hace lo propio con los tres bytes menos significativo tendrá tamaño 3,9 MBytes y el correspondiente a tomar los cuatro tendrá tamaño 5,3 MBytes. Cabe destacar, que el aumento de tamaño lógicamente conlleva un aumento de la redundancia introducida en su contenido. Partiendo de valores de entropía bastante lejanos a lo ideal que sería 8 bits por byte, se puede comprobar el fichero de salida se acerca mucho a ella, garantizando del mismo modo que ε_{hash} sea en todos los casos muy reducido.

Número de bits menos significativos seleccionados	Entropía de los bits del fichero a la entrada del extractor	Entropía de los bits del fichero a la salida del extractor	ε_{hash}
1	6,414144	7,999897	2^-26
2	7,038064	7,999901	2^-66
3	7,435437	7,999941	2^-91
4	6,956613	7,999958	2^-61

Tabla 4. Entropía del fichero a la entrada del extractor, entropía del fichero a la salida del extractor y valor de ε_{hash}.

13. Conclusiones y líneas futuras de trabajo

A medida que se han ido realizando las diferentes comprobaciones relativas al diseño y funcionamiento del generador de números aleatorios, se han ido encontrado ciertas dificultades relacionadas con el número de recursos limitados existente en el laboratorio. A pesar de ello, ha sido posible sacar adelante un sistema de generación de números aleatorios completo y con grado total de automatización, ya que se pueden controlar todos los procesos con un simple fichero ejecutable en Python previamente configurado. Desde la potencia emitida y modos de funcionamiento del propio láser, hasta todo el proceso relacionado con adquisición, registro y procesamiento de datos para obtener un fichero de bits aleatorios.

Más características destacadas del sistema llegan con la posibilidad de poder operar con diferentes niveles de potencia emitidos por el láser, moviéndose en un rango que se puede extender desde aproximadamente los 16 mW hasta la potencia máxima alcanzada por el láser, teniendo siempre en cuenta las limitaciones de potencia del resto de los componentes. Del mismo modo, la versatilidad también se extiende al apartado relacionado con la adquisición de datos, comprobando como desde un período de muestreo de 5 µs (el mínimo posible limitado por el conversor analógico digital del osciloscopio disponible) hasta 50 ms, la generación de números aleatorios es correcta y no se ve influenciada por este parámetro. Por último, como característica destacada, hay que remarcar que la tasa de generación de bits se aproxima a los 200 kBps, influyendo principalmente en esta tasa la velocidad de concatenación de los ficheros intermedios del equipo empleado.

En el apartado correspondiente a puntos más débiles que presenta el generador hay que mencionar la necesidad de establecer un control de temperatura estricto, siendo preciso que el láser disponga de un buen sistema de regulación, ya que las medidas se ven afectadas considerablemente, del orden de varias decenas de mV con tan solo un incremento o decremento de 3°C. Tampoco se puede pasar por alto el hecho de lograr una disposición adecuada de los componentes de trabajo, ya que al trabajar con fibra óptica es preciso que las conexiones se realicen sobre una superficie lo más estable posible. Este simple hecho puede provocar que los niveles de voltaje obtenidos en el osciloscopio decaigan drásticamente.

En cuanto a las líneas futuras con las que este Trabajo podría continuar desarrollándose, cabe mencionar algunas bastante destacadas relacionadas con alcanzar una mayor velocidad en la generación de bits, mayor capacidad de actuación automática frente a la detección de errores e integración en un solo dispositivo. Entre ellas se pueden destacar:

- Eliminación de ficheros intermediarios en la captura de datos, evitando de este modo un mayor consumo de espacio en el disco duro. Para ello cada vez que se genere un número de muestras del tamaño del buffer del osciloscopio, se añada a un fichero que acumule todas las muestras en vez de una concatenación final de todos los ficheros.

- Emplear una FPGA con un conversor analógico digital de mayor velocidad para poder explorar el período mínimo de muestreo al que es posible tomar muestras.

- Implementación del sistema como un conjunto más compacto, diseñando para ello una estructura que permita disponer todos los elementos en su interior y así poder ser portable sin necesidad de separar sus componentes. Un paso a mayores podría ser trasladar este generador al campo de la micro óptica.

- Estudiar cómo afectarían diferentes criterios de decisión a la hora de generar bits a partir de las muestras obtenidas. En el algoritmo descrito en este Trabajo el umbral de decisión es el propio signo de la diferencia entre dos muestras consecutivas, pudiéndose aumentar el número de umbrales de decisión. Para ello, se podría partir de investigaciones ya realizadas como las de Werner Schindler y Wolfgang Killman [33].

- Implementar en el proceso de adquisición de datos un detector de errores en el descenso significativo de la media de las muestras adquiridas. Esto podría indicar fenómenos indeseados como el agotamiento de la pila del detector durante la captura, algo que ha sido bastante frecuente durante la realización de este Trabajo.

- Manteniendo el mismo mecanismo de realimentación del láser, lograr una realimentación mayor mediante el uso de componentes que puedan presentar un menor número de pérdidas en la transmisión de potencia. Así se podría comprobar los efectos de un porcentaje de potencia de realimentación mayor al logrado en este sistema.

Bibliografía

[1] Reidler, I., Aviad, Y., Rosenbluh, M., & Kanter, I. (2009). Ultrahigh-speed random number generation based on a chaotic semiconductor laser. *Physical review letters*, *103*(2), 024102.

[2] Lorenz, E. N. (1963). Deterministic nonperiodic flow. *Journal of the atmospheric sciences*, *20*(2), 130-141.

[3] Hořák J., Krlín L., Aleš, R (2007). Deterministic chaos. Praha Academia. 1ª Ed.

[4] Uchida, A. (2012). *Optical communication with chaotic lasers: applications of nonlinear dynamics and synchronization*. John Wiley & Sons. 1ª Ed.

[5] Manual y hoja de especificaciones del PiscosCope 9200 A. https://www.picotech.com/download/datasheets/picoscope-9201a-9211a-9221a-9231a-data-sheet-es.pdf Consultado por última vez en julio del 2019.

[6] Hoja de especificaciones del DET025AFC/M. https://www.thorlabs.com/newgrouppage9.cfm?objectgroup_id=1297&pn=DET025AFC/M. Consultado por última vez en julio del 2019.

[7] Manual y hoja de especificaciones del Cobolt 06-01 Series. https://www.coboltlasers.com/wp-content/uploads/2018/12/D0136-K_Manual-Cobolt-06-01-Series_December_2018.pdf. Consultado por última vez en julio del 2019.

[8] Hoja de especificaciones del P5-630A-PCAPC-1. https://www.thorlabs.com/newgrouppage9.cfm?objectgroup_id=1634&pn=P5-630A-PCAPC-1. Consultado por última vez en julio del 2019.

[9] Hoja de especificaciones del FC632-50B-FC - 2x2. https://www.thorlabs.com/thorproduct.cfm?partnumber=FC632-50B-FC Consultado por última vez en julio del 2019.

[10] Hoja de especificaciones del VOA630-FC. https://www.thorlabs.com/newgrouppage9.cfm?objectgroup_id=6161 Consultado por última vez en julio del 2019.

[11] Hoja de especificaciones del P1-630A-FC-2. https://www.thorlabs.com/newgrouppage9.cfm?objectgroup_id=1362&pn=P1-630A-FC-2 Consultado por última vez en julio del 2019.

[12] Hoja de especificaciones de los ADAFC1 y ADAFC3. https://www.thorlabs.com/newgrouppage9.cfm?objectgroup_id=314&pn=ADAFC3 Consultado por última vez en julio del 2019.

[13] Ohtsubo, J. (2002). Chaos synchronization and chaotic signal masking in semiconductor lasers with optical feedback. *IEEE Journal of Quantum Electronics*, *38*(9), 1141-1154.

[14] Oliver, N., Soriano, M. C., Sukow, D. W., & Fischer, I. (2011). Dynamics of a semiconductor laser with polarization-rotated feedback and its utilization for random bit generation. *Optics letters*, *36*(23), 4632-4634.

[15] Rice, F. (2016). A frequency-domain derivation of shot-noise. *American Journal of Physics*, *84*(1), 44-51.

[16] Biblioteca Nolds de Python. https://pypi.org/project/Nolds/ Consultado por última vez en julio del 2019.

[17] Kantz, H., & Schreiber, T. (2004). *Nonlinear time series analysis* (Vol. 7). Cambridge university press.

[18] Rosenstein, M. T., Collins, J. J., & De Luca, C. J. (1993). A practical method for calculating largest Lyapunov exponents from small data sets. *Physica D: Nonlinear Phenomena*, *65*(1-2), 117-134.

[19] Grassberger, P., & Procaccia, I. (1983). Characterization of strange attractors. *Physical review letters*, *50*(5), 346.

[20] Pchelintsev, A. N. (2014). Numerical and physical modeling of the dynamics of the Lorenz system. *Numerical analysis and Applications*, *7*(2), 159-167.

[21] Argyris, J., Andreadis, I., Pavlos, G., & Athanasiou, M. (1998). The influence of noise on the correlation dimension of chaotic attractors. *Chaos, Solitons & Fractals*, *9*(3), 343-361.

[22] Steeb, W. H., & Andrieu, E. C. (2005). Lyapunov exponents, hyperchaos and Hurst exponent. *Zeitschrift für Naturforschung A*, *60*(4), 252-254.

[23] Lorentz Lo Sauer, Ent. Repositorio GitHub. https://github.com/lsauer/entropy Consultado por última vez en julio del 2019.

[24] Shannon, C. E. (1948). A mathematical theory of communication. *Bell system technical journal*, *27*(3), 379-423.

[25] Walker, J. (2008). ENT: a pseudorandom number sequence test program. http://www.fourmilab.ch/random/ Última vez consultado en julio del 2019.

[26] Brown, R. G., Eddelbuettel, D., & Bauer, D. (2013). Dieharder: A random number test suite. *Open Source software library, under development.* http://www.phy.duke.edu/~rgb/General/dieharder.php Última vez consultado en julio del 2019.

[27] Anderson, W. (2015). A Study of Entropy. https://sites.google.com/site/astudyofentropy/background-information/the-tests Última vez consultado en julio del 2019.

[28] Quantis Random Number Generator (2019). True random number generator exploring the randomness of quantum physics. https://www.idquantique.com/random-number-generation/products/quantis-random-number-generator/ Última vez consultado en julio del 2019.

[29] National Institute of Standards and Technology (2016). Random Bit Generation. https://csrc.nist.gov/projects/random-bit-generation/ Última vez consultado en julio del 2019.

[30] Von Neumann, J. (1963). Various techniques used in connection with random digits. *John von Neumann, Collected Works*, *5*, 768-770.

[31] ID Quantique Technical Paper on Randomness Extractor (2019), Version 1.0.

[32] ID Quantique White Paper. Randomness Extractor for the Quantis True Random Number Generation (2019). Version 1.0.

[33] Schindler, W., & Killmann, W. (2002, August). Evaluation criteria for true (physical) random number generators used in cryptographic applications. In *International Workshop on Cryptographic Hardware and Embedded Systems* (pp. 431-449). Springer, Berlin, Heidelberg.

Anexos

Anexo 1: fichero de configuración del osciloscopio (osciloscopio.py)

```
#!/usr/bin/python
# -*- coding: utf-8 -*-
#
# Copyright © 2018 Pico Technology Ltd. See LICENSE file for terms.
#
"""
This is a Python script for controlling a PicoScope 9200 Series Sampling Oscilloscope using the PicoScope 9000 COM object.
This will also work using the demo device in PicoSample3.
"""
#importación de bibliotecas
import win32com.client
import numpy as np
import matplotlib.pyplot as plt
import time
from timeit import timeit
import string
import sys
import shutil

#mensaje informativo para desconectar las entradas y realizar la autocalibración
print("Desconecte las entradas del panel local para iniciar la calibración. Introduzca 1 cuando lo haya hecho" +"\n")
ok = input()
if not ok=="1":
 sys.exit(0)
COMRCW = win32com.client.Dispatch("PicoScope9000.COMRC") #se crea un objeto COM
COMRCW.ExecCommand("Gui:RemoteLocal")
COMRCW.ExecCommand("Ch1:Acquire 1")  #sólo se adquirirán datos a  través del canal 1
COMRCW.ExecCommand("Ch2:Acquire 0")
COMRCW.ExecCommand("Gui:RemoteLocal")
COMRCW.ExecCommand("Flash:Calibr:AutocalTB") #se calibra tanto la escala de tiempos como las escala vertical
COMRCW.ExecCommand("Flash:Calibr:AutocalCh")
resultado=COMRCW.ExecCommand("Flash:Calibr:AutocalResult?") #se analiza si la autocalibración ha sido exitosa
resultado=resultado[28]
time.sleep(5)
if not resultado == "0":
  sys.exit(0)
#se solicita al usuario volver a conectar la entrada para que se inicie la adquisición de datos
if resultado == "0":
 print("Por favor, conecta la entrada. Introduzca 1 cuando lo haya hecho" +"\n")
 ok = input()
 if not ok=="1":
  sys.exit(0)
print("Iniciando captura de datos" +"\n")
tiempo_ini=time.time()#se inicia un contado para poder saber el tiempo que durarán todos los procesos
print("Tiempo inicial: 0"+"\n")
COMRCW.ExecCommand("TB:Mode A") #escala temporal modo A (sin retardos)
COMRCW.ExecCommand("TB:ScaleA 10e-4") #ajuste de la escala temporal en segundos por división
COMRCW.ExecCommand("Ch1:Display 1") #se muestra el canal 1 por pantalla
COMRCW.ExecCommand("Ch2:Display 0") #se desactiva el canal 2 por pantalla
COMRCW.ExecCommand("Acq:FitTo Single") #se ajusta la adquisición para el modo single
COMRCW.ExecCommand("Acq:Ch1:Mode Sample") #se adquieren las muestras en modo single
COMRCW.ExecCommand("Display:Ch1:Style Vectors") #se muestran las muestras por pantalla unidas por vectores
s1 = r'D:datos2\muestras\SenalGrabada.txty' #ruta de los ficheros intermedios
s2 = r'D:datos2\datos.txt' #ruta del fichero general con todas las muestras
f2 = open(s2,'w')
i = 1
cont = 1
num=5000
while i <=num: #se repite la iteracción el valor de num. Inicio de la adquisición y grabado de muestras
 COMRCW.ExecCommand("Acq:Ch1:RecLen 512") #longitud del buffer
 COMRCW.ExecCommand("Save:Memo:Source Ch1")
 COMRCW.ExecCommand("Save:Disk:NameMode Manual")
 COMRCW.ExecCommand("Save:Disk:FileFormat YOnly") #formato de salida exponencial
 COMRCW.ExecCommand("Save:Disk:FileName D:datos2\muestras\S"+str(i)+".txty") #generación de ficheros intermedios
 COMRCW.ExecCommand("Save:Disk:Save") #almacenamiento de muestras
 i += 1
tiempo_fini=time.time()-tiempo_ini
print("Tiempo final de adquisición: " + str(tiempo_fini) + "\n") #tiempo final de adquisición
```

```
i=1
while i <=num: #inicio de la concatenación de los ficheros intermedios en uno solo
 try:
  s1 = r"D:datos2\muestras\S"+str(i)+".txty"
  f1= open(s1,'r')
 except OSError: #control de errores en los ficheros intermedios
  i += 1
 else:
  f2 = open (s2,'a')
  contenido=f1.read()
  f2.write(contenido)
  i += 1
f2.close()
f1.close()
tiempo_fini2=time.time()-tiempo_ini
print("Tiempo final de concatenación: " + str(tiempo_fini2) + "\n") #tiempo final de concatenación
```

Anexo 2: creación de los histogramas (histogramas.m)

```
x=dlmread('datos.txt'); %carga las muestras
C=unique(x); %indica el número de muestras diferentes
histfit(x,75) %creación del histograma
pd = fitdist(x,'Normal') %parámetros del histograma de la gaussiana
```

Anexo 3: fichero de análisis de indicios de comportamiento caótico (medidasCaos.py)

```
# -*- coding: utf-8 -*-
"""
Created on Fri Jun 21 09:31:52 2019

@author: USUARIO_PORTATIL
"""
#importación de bibliotecas
from IPython.core.getipython import get_ipython
import bdb
from distutils.version import LooseVersion
import io
import os
import os.path as osp
import pdb
import shlex
import sys
import time
import warnings
import nolds
import numpy as np
import nolds
import numpy as np
#ruta con el fichero de las muestras
s1=r'C:\Users\USUARIO_PORTATIL\Desktop\datos\datos.txt'
listaDatos = []
f1= open(s1,'r')
i=1
cont=0
limite=8000
while True:
    line = f1.readline() #lectura muestra a muestra
    if not line: break
    cont=cont + 1
    if cont==limite: #se almacena una de cada limite muestras y se almacenan en una lista
      if line[-1] == '\n': #se elimina el caracter de salto de línea
       line = line[0:30]
       value2=float(line)
       listaDatos.append(value2)
       cont=0

LyapvExpE=nolds.lyap_e(listaDatos, 30,2) #cálculo de dos exponentes de Lyapunov (algoritmo de Rosenstein)
#con una dimensión embebida de 30
LyapvExpR=nolds.lyap_r(listaDatos,30) #cálculo de dos exponentes de Lyapunov (algoritmo de Eckmann)
#con una dimensión embebida de 30
hurst=nolds.hurst_rs(listaDatos) #cálculo del exponente de Hurst
dimension=nolds.corr_dim(listaDatos, 3, debug_plot=True,debug_data=True) #cálculo de la dimensión de
#correlación para una dimensión embebida de 3
```

Anexo 4: salida de Ent, Rgntest y Dieharder para un fichero de 2560000 muestras del láser sin realimentar

```
noerrivas@Michelangelo ~/Escritorio/pruebas $ ent bits.binEntropy = 7.989890 bits per byte.
Optimum compression would reduce the size
of this 160000 byte file by 0 percent.

Chi square distribution for 160000 samples is 2288.84, and randomly
would exceed this value 0.01 percent of the times.

Arithmetic mean value of data bytes is 126.4988 (127.5 = random).
Monte Carlo value for Pi is 3.213530338 (error 2.29 percent).
Serial correlation coefficient is 0.009623 (totally uncorrelated = 0.0).

noerrivas@Michelangelo ~/Escritorio/pruebas $ cat bits.bin|rngtestrngtest 4
Copyright (c) 2004 by Henrique de Moraes Holschuh
This is free software; see the source for copying conditions.  There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

rngtest: starting FIPS tests...
rngtest: entropy source drained
rngtest: bits received from input: 1280000
rngtest: FIPS 140-2 successes: 57
rngtest: FIPS 140-2 failures: 6
rngtest: FIPS 140-2(2001-10-10) Monobit: 0
rngtest: FIPS 140-2(2001-10-10) Poker: 4
rngtest: FIPS 140-2(2001-10-10) Runs: 3
rngtest: FIPS 140-2(2001-10-10) Long run: 0
rngtest: FIPS 140-2(2001-10-10) Continuous run: 0
rngtest: input channel speed: (min=1.164; avg=5.239; max=18.626)Gibits/s
rngtest: FIPS tests speed: (min=14.181; avg=40.468; max=61.527)Mibits/s
rngtest: Program run time: 30426 microseconds

noerrivas@Michelangelo ~/Escritorio/pruebas $ dieharder -a -g 201 -f bits.bin#=============================================================================#
#            dieharder version 3.31.1 Copyright 2003 Robert G. Brown          #
#=============================================================================#
   rng_name    |           filename             |rands/second|
 file_input_raw|                        bits.bin|  4.40e+07  |
#=============================================================================#
        test_name   |ntup| tsamples |psamples|  p-value |Assessment
#=============================================================================#
# The file file_input_raw was rewound 346 times
   diehard_birthdays|   0|       100|     100|0.10681666|  PASSED
# The file file_input_raw was rewound 2846 times
      diehard_operm5|   0|   1000000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 6046 times
  diehard_rank_32x32|   0|     40000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 7546 times
    diehard_rank_6x8|   0|    100000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 8201 times
   diehard_bitstream|   0|   2097152|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 13444 times
        diehard_opso|   0|   2097152|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 16939 times
        diehard_oqso|   0|   2097152|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 18577 times
         diehard_dna|   0|   2097152|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 18737 times
diehard_count_1s_str|   0|    256000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 21937 times
diehard_count_1s_byt|   0|    256000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 21997 times
 diehard_parking_lot|   0|     12000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 22037 times
    diehard_2dsphere|   2|      8000|     100|0.00002114|   WEAK
# The file file_input_raw was rewound 22067 times
    diehard_3dsphere|   3|      4000|     100|0.00015365|   WEAK
# The file file_input_raw was rewound 27828 times
     diehard_squeeze|   0|    100000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 27828 times
        diehard_sums|   0|       100|     100|0.00122980|   WEAK
# The file file_input_raw was rewound 28078 times
        diehard_runs|   0|    100000|     100|0.00000000|  FAILED
        diehard_runs|   0|    100000|     100|0.00000000|  FAILED
```

Anexo 5: salida de Ent, Rgntest y Dieharder para un fichero de 2560000 muestras del láser realimentado

```
noerrivas@Michelangelo ~/Escritorio/pruebas $ ent bits.binEntropy = 7.998945 bits per byte.

Optimum compression would reduce the size
of this 160000 byte file by 0 percent.

Chi square distribution for 160000 samples is 233.67, and randomly
would exceed this value 75.00 percent of the times.

Arithmetic mean value of data bytes is 127.5089 (127.5 = random).
Monte Carlo value for Pi is 3.163129078 (error 0.69 percent).
Serial correlation coefficient is -0.002105 (totally uncorrelated = 0.0).

noerrivas@Michelangelo ~/Escritorio/pruebas $ cat bits.bin|rngtestrngtest 4
Copyright (c) 2004 by Henrique de Moraes Holschuh
This is free software; see the source for copying conditions.  There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

rngtest: starting FIPS tests...
rngtest: entropy source drained
rngtest: bits received from input: 1280000
rngtest: FIPS 140-2 successes: 63
rngtest: FIPS 140-2 failures: 0
rngtest: FIPS 140-2(2001-10-10) Monobit: 0
rngtest: FIPS 140-2(2001-10-10) Poker: 0
rngtest: FIPS 140-2(2001-10-10) Runs: 0
rngtest: FIPS 140-2(2001-10-10) Long run: 0
rngtest: FIPS 140-2(2001-10-10) Continuous run: 0
rngtest: input channel speed: (min=1.096; avg=5.561; max=18.626)Gibits/s
rngtest: FIPS tests speed: (min=14.363; avg=41.621; max=61.330)Mibits/s
rngtest: Program run time: 29590 microseconds
noerrivas@Michelangelo ~/Escritorio/pruebas $

noerrivas@Michelangelo ~/Escritorio/pruebas $ dieharder -a -g 201 -f bits.bin#=============================================================================#
#            dieharder version 3.31.1 Copyright 2003 Robert G. Brown          #
#=============================================================================#
   rng_name    |           filename             |rands/second|
 file_input_raw|                       bits.bin|  4.43e+07  |
#=============================================================================#
        test_name   |ntup| tsamples |psamples|  p-value |Assessment
#=============================================================================#
# The file file_input_raw was rewound 346 times
   diehard_birthdays|   0|       100|     100|0.00010882|   WEAK
# The file file_input_raw was rewound 2846 times
      diehard_operm5|   0|   1000000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 6046 times
  diehard_rank_32x32|   0|     40000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 7546 times
    diehard_rank_6x8|   0|    100000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 8201 times
   diehard_bitstream|   0|   2097152|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 13444 times
         diehard_opso|   0|   2097152|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 16939 times
         diehard_oqso|   0|   2097152|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 18577 times
          diehard_dna|   0|   2097152|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 10737 times
diehard_count_1s_str|   0|    256000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 21937 times
diehard_count_1s_byt|   0|    256000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 21997 times
 diehard_parking_lot|   0|     12000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 22037 times
    diehard_2dsphere|   2|      8000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 22067 times
    diehard_3dsphere|   3|      4000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 27821 times
     diehard_squeeze|   0|    100000|     100|0.00000000|  FAILED
# The file file_input_raw was rewound 27822 times
         diehard_sums|   0|       100|     100|0.00969589|  PASSED
# The file file_input_raw was rewound 28072 times
         diehard_runs|   0|    100000|     100|0.00000000|  FAILED
         diehard_runs|   0|    100000|     100|0.00000000|  FAILED
```

Anexo 6: fichero de conversión de muestras a bits (DifToBin.py)

```python
import sys

bitn=0
byte=0
par=0
for line in sys.stdin: #lectura de la entrada
    if par==1:
       bit=(float(line)-vant)>0 #se escribe un uno si el signo de la diferencia es positivo
       byte=byte+int(bit)*2**bitn
       bitn=bitn+1
       if bitn==8:
         sys.stdout.write(chr(byte)) #escritura de un byte
         bitn=0
         byte=0
    par=1-par
    vant=float(line) #se almacena la muestra de la iteracción anterior
```

Anexo 7: fichero de conversión de muestras a bits tomando cierto número de ellos (DifToBinNbits.py)

```
from __future__ import print_function
import sys
import struct

def float_to_bin(num):
    return format(struct.unpack('!I', struct.pack('!f', num))[0], '032b')

def bin_to_float(binary):
    return struct.unpack('!f',struct.pack('!I', int(binary, 2)))[0]

bitn=0
byte=0
par=0
for line in sys.stdin:
    if par==1:
       bit=(float(line)-vant) #diferencia de dos muestras consecutivas
       binaria=float_to_bin(float(bit)) #conversión de float a binario
       chars=[ chr(int(binaria[0+8*i:8*(i+1)],2)) for i in range(0,4)]
       print(chars[0], sep='', end='') #cuarto bit menos significativo
       print(chars[1], sep='', end='') #tercer bit menos significativo
       print(chars[2], sep='', end='') #segundo bit menos significativo
       print(chars[3], sep='', end='') #primer bit menos significativo
    par=1-par
    vant=float(line) #se almacena la muestra de la iteracción saliente
```