

TESIS DOCTORAL

Doctorado en Ingeniería
Procesamiento de Señales e Imágenes

Título:

“Procesamiento en tiempo real de series temporales de láseres de estado sólido y su aplicación a la optimización de las señales de salida del láser”

Autora: Ing. Myriam Eugenia Nonaka
Director de tesis: Dr. Marcelo Gregorio Kovalsky

Buenos Aires - 2021

AGRADECIMIENTOS

En primer lugar, quisiera agradecer a mi director de tesis Dr. Marcelo G. Kovalsky por guiarme, por su dedicación, por sus aportes, por sus consejos, por su calidez y por su bondad durante el transcurso de esta tesis.

A CITEDEF, por formarme en la parte científica durante mis estudios de grado con la beca de capacitación profesional PIDDEF, permitiendo luego mi incorporación al grupo de investigación del DEILAP como así también a realizar mi trabajo de tesis doctoral. A mis colegas, quienes siempre han estado presentes y con los cuales he formado un grupo de trabajo muy unido.

A la UTN, por brindarme una formación de grado como de posgrado. A todo su personal docente y no docente, por el compromiso con el alumno y su amabilidad de siempre.

A toda mi familia y amigos, quienes me han acompañado en esta etapa. En especial, a mi esposo, por estar presente en todos los aspectos, siendo incondicionalmente mi gran apoyo emocional. Finalmente, a mis hijos, por brindarme tantas alegrías y por motivarme a concluir mi tesis doctoral.

ÍNDICE GENERAL

ÍNDICE GENERAL	5
ÍNDICE DE FIGURAS	9
ÍNDICE DE TABLAS	13
LISTA DE ACRÓNIMOS	15
RESUMEN	17
ABSTRACT	19
INTRODUCCIÓN	23
Problema de investigación.....	24
Hipótesis.....	24
Objetivo principal.....	26
Objetivos secundarios.....	26
Desarrollo de la tesis por capítulos.....	26
CAPÍTULO 1: Señales de salida de láseres de estado sólido	27
1.1 Introducción.....	27
1.2 Señales de salida.....	29
1.3 Detección de señales.....	30
1.3.1 <i>Time stamping</i>	30
1.3.2 Series temporales.....	31
1.4 Tipos de señales.....	36
1.4.1 Señales periódicas.....	37
1.4.2 Señales caóticas.....	37
1.4.3 Señales aleatorias (<i>random</i>).....	39
CAPÍTULO 2: Fundamentos del procesamiento de series temporales	43
2.1 Introducción.....	43
2.2 Reconstrucción a partir de una serie temporal de un atractor.....	44
2.3 Enfoques de aleatoriedad.....	46
2.3.1 Tests del NIST.....	46
2.3.1.1 Desarrollo de códigos de los tests del NIST en MATLAB.....	54
2.3.1.2 Consideraciones de los tests del NIST.....	55
2.3.2 Complejidad de Lempel-Ziv como aproximación de la complejidad de Kolmogorov.....	55

2.3.2.1 Códigos de la complejidad de Lempel-Ziv en MATLAB.....	57
CAPÍTULO 3: Arreglos experimentales.....	59
3.1 Introducción.....	59
3.2 Experimento desigualdad de Bell.....	59
3.3 Experimento láser sólido con modulación de pérdidas.....	65
CAPÍTULO 4: Automatización, control y procesamiento.....	71
4.1 Introducción.....	71
4.2 Automatización y control para señales cuánticas.....	72
4.2.1 Comunicación wi fi entre las tres estaciones.....	72
4.2.2 Actividades realizadas por los hijos en cada estación A y B.....	76
4.2.3 Actividades realizadas por la madre en la estación C.....	83
4.2.4 Programación de dispositivos.....	86
4.2.4.1 Control de rotadores.....	86
4.2.4.2 Control del generador SDG.....	88
4.2.4.3 Control de ID900.....	92
4.2.4.3.1 Comunicación de los ID900.....	93
4.2.4.3.2 Sincronización de los relojes internos de ambos ID900.....	93
4.2.4.3.3 Salvaguarda de archivos.....	94
4.2.4.3.4 Software desarrollado para el área de señales cuánticas.....	95
4.3 Automatización, control y procesamiento para señales clásicas.....	98
4.3.1 Fotodetector.....	100
4.3.2 Control del PicoScope.....	100
4.3.3 Control del generador de funciones.....	103
4.3.4 Algoritmo de procesamiento.....	103
4.3.5 Software de <i>streaming</i> para láser con modulación de pérdidas.....	110
RESULTADOS.....	111
CONCLUSIONES.....	129
Publicaciones originadas por esta tesis.....	135
BIBLIOGRAFÍA.....	137
Apéndice A: Códigos de los tests del NIST desarrollados en MATLAB.....	141
Apéndice B: Códigos desarrollados en MATLAB script node de LabVIEW para los PRM1Z8	
.....	177

Apéndice C: Capturas de pantalla de software desarrollado en LabVIEW	181
Apéndice D: Códigos de algoritmos de procesamiento desarrollados/utilizados en MATLAB	185
Apéndice E: Comandos SPCI utilizados en el ID900 para el desarrollo del software propio	191
Apéndice F: Implementación en LabVIEW para el cálculo de la dimensión de <i>embedding</i> y de los coeficientes de Lyapunov con paquetes de TISEAN	193
Apéndice G: Ecuaciones y referencias de nomenclaturas para el cálculo del parámetro de Bell S_{CHSH}	201

ÍNDICE DE FIGURAS

Figura 1. (a) Gráfica de parte de la señal de salida del láser con modulación de pérdidas y la detección de sus picos de amplitud. (b) Gráfica de la señal amplificada de salida del láser con modulación de pérdidas y la detección de sus picos de amplitud para un pulso de señal.	32
Figura 2. Gráfica de una parte de la serie temporal de amplitud de la señal de la figura 1.	33
Figura 3. Gráfica de una parte de la serie temporal de diferencia de tiempos entre picos sucesivos de amplitud de la señal de la figura 1.	33
Figura 4. Señal experimental muestreada a 2 MS/s. Los puntos azules son las diferentes muestras. En color cian, se realiza el trazado de las muestras.	35
Figura 5. Señal experimental muestreada a 10 MS/s. Los puntos azules son las diferentes muestras. En color cian, se realiza el trazado de las muestras.	35
Figura 6. Gráfica de una parte de una señal experimental de la salida del láser con modulación de pérdidas. Esta señal se caracteriza por ser periódica.	37
Figura 7. Gráfica de una parte de una señal experimental de salida del láser con modulación de pérdidas. Esta señal se caracteriza por ser caótica.	39
Figura 8. Gráfica de una parte de la señal experimental de la salida del láser con modulación de pérdidas. Esta señal se caracteriza por ser aleatoria.	41
Figura 9. Esquema del arreglo experimental para experimento de desigualdades de Bell con estaciones A y B separadas por 60 m. 1) y 8): Comunicación Ethernet de las computadoras hijo A y B con sus TDC. 2) y 3): Comunicación USB de la computadora madre con los rotadores motorizados PRM1Z8 en donde se montan las láminas de media onda. 4) Comunicación USB de computadora madre con generador de señales. 5) Señal del generador para modulador del láser. 6) y 7), Señal de salida del fotodetector atenuado al canal 3 de los TDCs (<i>Time to Digital Converters</i>), en los canales 1 y 2 recibe las señales de los SPCM (módulo para conteo de fotones). OE: Óptica de enfoque. CB: Compensador de birrefringencia. FO: Fibra óptica de 60 m (total). PF: Polarizador de fibra. BBO: Cristal generador de pares de fotones entrelazados.	61
Figura 10. Fotografía del arreglo experimental de la estación A: D.1) y D.2): Señal de los SPCM hacia el TDC de estación A, ch1 y ch2. D.3) y D.4): Entradas a los SPCM provenientes del polarizador a fibra. D.PF: Polarizador de fibra. D.CB: Compensador de birrefringencia. E) PC hijo A. F) TDC estación A. G.5) Rotador motorizado PRM1Z8 estación A.	62
Figura 11. Fotografía del arreglo experimental de la estación B: D.1) y D.2): Señal de los SPCM hacia el TDC de estación B, ch1 y ch2. D.3) y D.4): Entradas a los SPCM provenientes del polarizador a fibra. D.CB: Compensador de birrefringencia. D.PF: Polarizador de fibra. E.6) PC hijo B. E.5) TDC estación B F.7) Rotador motorizado PRM1Z8 estación B.	63
Figura 12. Fotografía del arreglo experimental de la estación C. En recuadros de líneas verde y magenta, se encuentran indicadas partes de la estación A y B respectivamente, correspondiente a los rotadores PRM1Z8 . EPS: Espejo de primera superficie. E.5) Salida de fotodetector al ch3 de los TDC de las estaciones A y B. D.1) y D.2): Drivers y conexión USB de rotadores a PC madre. D.3) Pantalla. F) Diodo láser con su óptica de enfoque, permite modulación con generador de funciones en D.4). G) PC madre.	64
Figura 13. Fotografía del arreglo experimental del láser con modulación de pérdidas. 1) Conexión al amplificador-generador de funciones. 2) Conexión al PicoScope. 3) Conexión a la fuente del diodo.	65
Figura 14. Fotografía de parte del sistema de bombeo + medio activo. En el recuadro se muestra una vista superior en detalle del medio activo y la lente GRIN.	66
Figura 15. 1) Fotografía del amplificador Falcon Systems 50x. 2) Fotografía de la fuente del diodo Spectra Diode modelo SDL 805. 3) Imagen del generador de funciones SDG1025. 4) Imagen PicoScope modelo 6403B.	67

Figura 16. Esquema de la configuración del láser con cavidad en V y los dispositivos involucrados para la automatización y control de la señal de salida del láser.	68
Figura 17. Esquema de comunicación entre las estaciones A, B y C.	72
Figura 18. Imagen de router wi fi alta potencia	73
Figura 19. Diagrama de flujo simplificado de comunicación tanto de hijo A como B.....	75
Figura 20. Diagrama de flujo simplificado de la comunicación madre	76
Figura 21. Captura archivo de texto con ángulos para los cuales se rotará al polarizador de media onda.	77
Figura 22. Parte del diagrama en bloques en LabVIEW para la ejecución del ejecutable para la transferencia de datos provisto por el fabricante.....	78
Figura 23. Parte del diagrama en bloques en LabVIEW de la comunicación para la transferencia de datos desde el ID900 con la computadora portátil por medio de la dirección "tcp://localhost:6060"	78
Figura 24. Parte del diagrama en bloques en LabVIEW para la configuración de puerto del canal 1 y 2. Las flechas en violeta indican el bloque que permite realizar el envío de dicho mensaje al localhost:6060.	79
Figura 25. Parte del diagrama en bloques en LabVIEW de la configuración del ID900 mediante tcp://dirección IP del dispositivo:5555.....	80
Figura 26. Esquema de bloques en LabVIEW para la conexión, programa escucha el puerto 8006 por donde la madre envía el carácter.....	80
Figura 27. Esquema de bloques en LabVIEW para la espera de carácter B y desconexión del puerto 8006.....	81
Figura 28. Parte de esquema de bloques en LabVIEW para la conexión con la IP MADRE y envío de caracteres por puerto 8006.....	81
Figura 29. Parte de esquema de bloques en LabVIEW para el envío de carácter por puerto 8006 y desconexión con la madre.	82
Figura 30. Diagrama de flujo extendido de la comunicación genérica de los hijos.....	83
Figura 31. Señal de estado alto para modular el láser	84
Figura 32. Listado de ángulos. Ejemplificación arbitraria.	85
Figura 33. Lectura de archivo de ángulos y salvaguarda en un vector.	85
Figura 34. Diagrama de flujo extendido de la comunicación madre	86
Figura 35. Diagrama de flujo simplificado del control de plataformas motorizadas	88
Figura 36. Diagrama de flujo del control con generador de señales	90
Figura 37. Diagrama en bloques de LabVIEW para la activación del generador de funciones ..	91
Figura 38. Diagrama en bloques de LabVIEW para la configuración del estado ON/OFF del generador de funciones.....	91
Figura 39. Diagrama en bloques de LabVIEW para la desactivación del generador de funciones	92
Figura 40. ID900 correspondiente a la estación B. Actúa como maestro:(1) Parte trasera (2) Parte delantera	94
Figura 41. ID900 correspondiente a la estación A. Actúa como esclavo:(1) Parte trasera (2) Parte delantera.	94
Figura 42. Diagrama de flujo de la lógica de decisión para el análisis del tipo de señal en una serie temporal.....	99
Figura 43. Diagrama de flujo del software de streamming para láser con modulación de pérdidas.	101
Figura 44. (1) Diagrama en bloques en LabVIEW de apertura de la comunicación de la PC con el PicoScope del kit SDK. (2) Diagrama en bloques en LabVIEW de cierre de la comunicación de la PC con el PicoScope del kit SDK.	102

Figura 45. Diagrama en bloques en LabVIEW para la configuración del PicoScope del kit SDK.	102
Figura 46. Diagrama en bloques en LabVIEW para la adquisición del bloque de datos del kit SDK.	102
Figura 47. Señal de salida del láser experimental y ampliación de señal.	104
Figura 48. Puntos en rojo, son los puntos que serán de interés para la detección de picos	105
Figura 49. Picos de amplitud de señal en color verde.....	106
Figura 50. Fotografía de plataformas motorizadas de la estación A y B con láminas de media onda montadas	113
Figura 51. Fotografía de la computadora con el TDC ID900 de la estación B con el programa de <i>time stamping</i>	113
Figura 52. Diagrama para la adquisición de archivos <i>time stamping</i>	115
Figura 53. Curvas de contraste obtenidas con registro <i>time stamping</i>	116
Figura 54. Captura del osciloscopio de la señal de la modulación del láser en color amarillo y salida del láser en color celeste de la medición nro 1.	118
Figura 55. Fotografía del arreglo experimental del láser con modulación de pérdidas, adquiriendo con el PicoScope y visualizando la señal con la computadora en tiempo real.....	119
Figura 56. Captura de pantalla de la gráfica correspondiente al software desarrollado de adquisición en tiempo real de la señal de salida del láser con modulación de pérdidas de la medición nro 1.....	120
Figura 57. Captura de pantalla de la gráfica que arroja el programa en LabVIEW a través de Matlab correspondiente a picos detectados de la medición nro 1.	121
Figura 58. Captura de pantalla de la gráfica que arroja el programa en LabVIEW a través de MATLAB correspondiente a picos detectados de la medición nro 1 (la de la derecha es la misma señal, pero observada más en detalle).	121
Figura B.1 Código correspondiente al bloque 1 del diagrama de flujos de la Figura 37.....	177
Figura B.2. Código correspondiente al bloque 2 de la Figura 38	178
Figura B.3. Código correspondiente al bloque 3 de la Figura 38	179
Figura C.1 Captura de pantalla del software madre	181
Figura C.2 Captura de pantalla del software <i>time stamping</i> Alice	181
Figura C.3 Captura de pantalla del Software de coincidencias en LabVIEW	182
Figura C.4 Captura de pantalla del software de <i>streaming</i> para láser con modulación de pérdidas.	182
Figura C.5 Captura de pantalla de GUI o <i>app</i> tests del NIST en MATLAB.....	185
Figura F.1. Input del comando para ejecución del <i>false_nearest</i> en MATLAB script Node de LabVIEW.	194
Figura F.2. Captura de pantalla de apertura de archivo resultado.txt de los <i>outputs</i> (salidas) del programa <i>false_nearest</i> de TISEAN en LabVIEW y salvaguarda en arreglos.	195
Figura F.3. Captura de pantalla de salvaguarda de primeros coeficientes de cada dimensión del archivo resultado.txt en LabVIEW.	195

Figura F.4. Captura de pantalla de comparación (decreciente) de últimos valores de tres elementos en los arreglos de la figura F.2 en LabVIEW.	196
Figura F.5. Captura de pantalla de comparación de último elemento decreciente del arreglo respecto del valor umbral configurado en LabVIEW	196
Figura F.6. Captura de pantalla si se ha detectado dimensión de <i>embedding</i> en LabVIEW.	198
Figura F.7. Captura de pantalla de apertura de archivo resultado_lyap.txt de los <i>outputs</i> (salidas) del programa lyap_spec de TISEAN en LabVIEW.....	198
Figura F.8. Captura de pantalla de apertura de archivo resultado_lyap.txt de los <i>outputs</i> (salidas) del programa lyap_spec de TISEAN en LabVIEW y salvaguarda en arreglos.	199
Figura F.9. Captura de pantalla de comparación de al menos un coeficiente positivo y sumatoria de coeficientes total negativa en LabVIEW.....	199

ÍNDICE DE TABLAS

Tabla 1. Listado de 10 valores de <i>time stamping</i> de la estación A.....	31
Tabla 2. Especificaciones técnicas del router wi fi de alta potencia	73
Tabla 3. Direcciones IP para la comunicación TCP/IP	74
Tabla 4. Primeros 12 elementos de la serie temporal de picos de señal. En la columna 1, se encuentra el tiempo del pico de amplitud y en la columna 2, la intensidad.....	107
Tabla 5. Primeros 12 elementos de la serie temporal de diferencias de tiempo entre picos de amplitud. En la columna 1, se encuentra el número de pico consecutivo y en la columna 2, el valor entre diferencia de tiempo entre picos consecutivos.....	107
Tabla 6. Tabla de valores de configuración de señal de modulación del láser en estado ON y OFF, para el experimento del arreglo experimental de señales cuánticas.....	111
Tabla 7. Tabla de valores de posiciones angulares correspondientes a los rotadores de las estaciones A y B.....	112
Tabla 8. Tabla de valores de pasos a realizar el inicio del proceso de medición para el experimento de señales cuánticas.....	114
Tabla 9. Tabla de valores de configuración de señal de modulación del láser y corriente de la fuente del diodo, para el experimento correspondiente al arreglo experimental de señales clásicas de la medición nro 1.....	117
Tabla 10. Tabla de los pasos a realizar para la adquisición y procesamiento de señales clásicas de un láser con modulación de pérdidas.....	118
Tabla 11. 1) Listado de 13 primeros picos de amplitud de la serie temporal en unidades de V de la medición nro 1. 2) Listado de 13 primeros picos de amplitud de la serie temporal binarizada de la medición nro 1.....	122
Tabla 12. Tabla de resultados de la medición nro 1 al aplicar el software para láser con modulación de pérdidas.....	123
Tabla 13. Tabla de resultados de la medición nro 2 al aplicar el software para láser con modulación de pérdidas.....	125
Tabla 14. Tabla de resultados de la medición nro 3 al aplicar el software para láser con modulación de pérdidas.....	126
Tabla 15. Tabla de resultados de la medición nro 4 al aplicar el software para láser con modulación de pérdidas.....	127
Tabla F.1. Pasos en LabVIEW para el cálculo de la dimensión de <i>embedding</i> con programa false_nearest de TISEAN.....	194
Tabla F.2. Pasos en LabVIEW para el cálculo de los exponentes de Lyapunov con programa lyap_spec de TISEAN.....	197
Tabla G.1. Coincidencias correspondientes al parámetro S_0	202
Tabla G.2. Coincidencias correspondientes al parámetro S_1	202
Tabla G.3. Coincidencias correspondientes al parámetro S_2	202
Tabla G.4. Coincidencias correspondientes al parámetro S_3	203

LISTA DE ACRÓNIMOS

BBO	<i>Beta-BaB2O4</i>
CITEDEF	Instituto de Investigaciones Científicas y Técnicas para la Defensa
DEILAP	Departamento de Investigaciones en Láseres y sus Aplicaciones
FFT	<i>Fast Fourier Transform</i>
FWHM	<i>Full Width at Half Maximum</i>
GPIB	<i>General Purpose Interface Bus</i>
GS	<i>Gigasamples</i>
KCM	<i>Kolmogorov complexity spectrum highest value</i>
LASER	<i>Light Amplification by Stimulated Emission of Radiation</i>
LFSR	<i>Linear Feedback Shift Register</i>
LIDAR	<i>Laser Imaging Detection and Ranging</i>
LXI	<i>LAN based eXtensions for Instrumentation</i>
MASER	<i>Microwave Amplification by Stimulated Emission of Radiation</i>
MS	<i>Megasamples</i>
NIST	<i>National Institute of Standards and Technology</i>
NI-VISA	<i>National Instruments -Virtual Instrument Software Architecture</i>
PIDDEF	Programa de Investigación y Desarrollo para la Defensa
PXI	<i>PC eXtensions for Instrumentation</i>
QKD	<i>Quantum Key Distribution</i>
SC	Sistema de Colección
SPDC	<i>Spontaneous Parametric Down-Conversion</i>
SPCI	<i>Standard commands for Programable Instruments</i>
SPCM	<i>Single Photon Counters Modules</i>
TISEAN	<i>Time Series Analysis</i>
TCP/IP	<i>Transmission Control Protocol/Internet Protocol</i>
TDC	<i>Time to Digital Converter</i>
TTL	<i>Transistor-Transistor Logic</i>
UPS	<i>Uninterruptible Power Supply</i>
USB	<i>Universal Serial Bus</i>
VXI	<i>VME eXtensions for Instrumentation</i>

RESUMEN

En esta tesis, se estudiarán sistemas basados en láseres sólidos. Se buscará develar las propiedades de dos sistemas físicos relevantes: uno para señales cuánticas y otro para señales clásicas.

En cuanto al conocimiento básico, aportará una exhaustiva y fundamental comprensión de la dinámica de un sistema basado en láseres sólidos bombeados por diodos. Respecto del punto de vista tecnológico, se encararán los primeros desarrollos experimentales de procesamiento de la señal del láser para láseres con modulación de pérdidas que permitirán, a partir de la medición de alguna variable del sistema, la determinación del carácter de la señal. Siendo de gran interés para el estudio de señales caóticas y aleatorias de un sistema no lineal.

Se presentará la implementación de la automatización y el control del experimento de las desigualdades de Bell con las estaciones de Alice y Bob a distancia (60 m aprox.), siendo este arreglo experimental único en su tipo en nuestro país. Se desarrollarán modelos matemáticos y algoritmos que permitirán, a partir de una serie temporal de datos, mensurar la aleatoriedad de una secuencia. Esto tiene implicancias cruciales en áreas como los fundamentos de la mecánica cuántica, así como las aplicaciones prácticas como la generación de números al azar.

Palabras claves: láseres sólidos, dinámica no lineal, caos, procesamiento de señales.

ABSTRACT

In this thesis, systems based on solid lasers will be studied. It will seek to reveal the properties of two relevant physical systems: one for the quantum signals and other for the classical signals.

In terms of basic knowledge, it will provide an exhaustive and fundamental comprehension of the dynamics of a system based on diode pumped solid state lasers. Regarding technological point of view, it will be addressed the first experimental developments of laser signal processing for lasers with loss modulation that will allow, from the measurement of some variable of the system, the determination of the character of the signal. Being of great interest for the study of chaotic and random signals of nonlinear system.

The implementation of automation and control of the experiment of Bell's inequalities with the Alice and Bob stations at a distance (approx. 60m) will be presented, this set up is being the only one of its kind in our country. Mathematical models and algorithms will be developed that will allow, from a time series of data, to measure the randomness of a sequence. This has crucial implicatons in areas such as the fundamentals of quantum mechanics, as well as the practical application such as the generation of random numbers.

Key words: solid lasers, nonlinear dynamics, chaos, signal processing.

Dedicado a mi familia.

INTRODUCCIÓN

Los láseres sólidos son los más ampliamente empleados en aplicaciones en la investigación, en el área médica, en la industria, en aplicaciones civiles y militares. En cuanto al estudio de sus propiedades físicas, no solo se presenta como un interesante desafío académico sino que tiene implicancias prácticas muy importantes: en el área de pares de fotones entrelazados, para señales cuánticas, determinar la aleatoriedad de una secuencia temporal o serie temporal, tiene un alto impacto en áreas de la mecánica cuántica, así como las aplicaciones prácticas como la generación de números al azar; en cuanto a las señales clásicas, la comprensión de sus propiedades, conducirán a láseres más compactos y eficientes.

Con la aparición de diodos láser de alta potencia, se pudo reemplazar el tradicional bombeo por lámpara flash de los viejos láseres sólidos, incrementando así su robustez y disminuyendo su tamaño y requerimientos de consumo. Sin embargo, estos tipos de láseres presentan inestabilidades propias de los sistemas no lineales, que hacen que estas técnicas, atractivas por lo sencillas y económicas, sean inaplicables en algunos casos. La intensidad de los pulsos a menudo no es uniforme y tampoco la separación temporal entre pulsos sucesivos. El estudio de esta dinámica no lineal no solo se presenta como un interesante desafío académico, sino que tiene implicancias prácticas muy importantes, toda vez que la comprensión de esta inestabilidad intrínseca del sistema conducirá a láseres más compactos, eficientes y con señales de salida estables. En el trabajo [1], se demuestra tanto teórica como experimentalmente, por primera vez a nivel mundial, que las inestabilidades en la frecuencia de repetición de esta clase de láseres (*jitter*), responde al caos determinístico en baja dimensión. Este hallazgo, más allá de su interés básico, para el área de sistemas dinámicos no lineales, tiene repercusiones prácticas ya que permitiría aprovechar la impredecibilidad intrínseca del sistema para transmitir la información codificada según el esquema de criptografía láser caótica.

Las no linealidades en láseres son conocidas casi en forma contemporánea con los láseres mismos. Sin embargo, su estudio detallado solo cobró fuerza, sobre todo en el campo experimental, en los años recientes, con la aparición de medios digitales rápidos, capaces de almacenar gran cantidad de información. Nuestro grupo de trabajo, el Laboratorio de Láseres Sólidos del DEILAP –CITEDEF es pionero, a nivel internacional, en el estudio experimental de la dinámica en láseres sólidos y tiene numerosas publicaciones al respecto. Con la realización de esta tesis, se pretende definir e implementar técnicas de procesamiento de señales que permitan, a partir de la medición de alguna variable del sistema físico, la determinación del carácter de la señal, es decir, si presenta características clásicas o cuánticas y poder discernir entre señales caóticas y aleatorias.

Problema de investigación

La adquisición de señales cuánticas se realiza en el laboratorio a oscuras, debido a la sensibilidad de los dispositivos de detección. Cometer errores en la manipulación de dispositivos durante la medición es muy alta. En el país, no se dispone de un sistema que permita el control de los diferentes módulos en forma sistematizada para arreglos experimentales de las desigualdades de Bell con las estaciones Alice y Bob. La utilización de técnicas como el método de *time stamping* crea listados de tiempos de llegada de fotones para cada una de las estaciones Alice y Bob. Para el estudio de procesos dinámicos en el área de señales cuánticas, puede generar resultados desacertados si la creación de dichos listados no se encuentra en sincronismo. Salvaguardar en forma correcta los datos, es esencial para el procesamiento de la señal a posteriori. Poder efectuar ensayos de esta índole, permite la realización de experimentos fundamentales de mecánica cuántica, aplicaciones en comunicaciones, QKD (*Quantum Key Distribution*), generación de números al azar, etc.

Para el estudio de señales clásicas se requiere de la construcción de un prototipo de láser sólido bombeado por diodos, cuyo comportamiento presenta inestabilidades propias de los sistemas no lineales. Se requerirá del desarrollo de algoritmos para la adquisición y procesamiento para la descripción y análisis de dicho comportamiento. Mediante el conocimiento de la dinámica no lineal del sistema, desde el punto de vista tecnológico, se encararán los primeros desarrollos experimentales de técnicas de procesamiento de señales del láser de automatización de dispositivos para la adquisición de señal en tiempo real y el control del instrumental para el ajuste de la señal mediante la variación de una variable del sistema físico. Esto constituye una gran contribución en el área de criptografía, al poder emplear estos láseres compactos y robustos para la transmisión de mensajes cifrados, con aplicaciones inmediatas en las áreas de Defensa y aquellas aplicaciones que requieran la transmisión de información segura (sector gubernamental, bancario, comercial). Por otra parte, la optimización de un sistema láser implica desarrollar conocimientos y técnicas novedosas en el campo del procesamiento de señales. Se deberá afrontar la dificultad de extraer información de un sistema físico complejo, que puede evolucionar hacia un estado de caos determinístico de baja dimensión.

Hipótesis

La hipótesis central, es que mediante dos sistemas físicos diferentes basados en láseres sólidos: uno para señales cuánticas y otro para señales clásicas, se develarán propiedades físicas relevantes

que permitirán determinar el carácter de la señal, sus características y el discernimiento entre señales caóticas y aleatorias.

En cuanto a la suposición que se quiere llegar a demostrar, las metodologías que se desarrollarán para verificar esta hipótesis serán una combinación de técnicas experimentales y teóricas.

Respecto de la verificación de los hechos a través de métodos de observación: para las señales clásicas, se estudiará la dinámica no lineal del sistema láser sólido, tanto desde el punto de vista teórico como experimental. Para ello, se prevé la construcción de prototipos láseres. Posterior a su construcción, se caracterizará mediante el empleo de medidores de potencia y energía, fotodiodos rápidos, osciloscopios y cámaras infrarrojas para determinar la forma espacial del haz. Además, se empleará el estudio computacional mediante simulaciones numéricas; para las señales cuánticas, se determinará, a partir de mediciones experimentales de coincidencias para distintos ángulos de los polarizadores, las desigualdades de Bell con las estaciones Alice y Bob separadas una distancia de 60m aproximadamente. Se desarrollará un sistema para la automatización y control de dispositivos involucrados, esto permitirá adquirir mediante la utilización de técnicas como el método de *time stamping* (listados de tiempos de llegada de fotones para cada una de las estaciones). El proceso de diseño de algoritmos será realizado en la plataforma MATLAB y mediante el empleo de software propio utilizando LabVIEW utilizando el PicoScope (osciloscopio de PC), para la recolección de datos en el caso de las señales clásicas y el TDC (*time to digital converter*), para las señales cuánticas.

En lo referente a la evaluación de señales aperiódicas o caóticas a partir de observaciones, merece recordarse aquí que no tiene sentido comparar la evolución "instante a instante" de una serie de tiempos, sino los parámetros que caracterizan a su atractor.

Para el análisis de las señales se empleará todo el arsenal estándar de análisis de señales y herramientas de la dinámica no lineal, como se ha mencionado en el párrafo anterior. Para evaluar estas funciones se desarrollará software propio para tratar las series temporales.

En cuanto a la experimentación con lo observado y comprobación de la suposición por demostrar, se desarrollarán algoritmos que implementen los modelos numéricos que resulten del estudio teórico-experimental del láser. Con estas herramientas, se podrá a partir de la medición de alguna variable del sistema, la determinación del carácter de la señal.

Objetivo principal

El objetivo principal de esta tesis es la descripción de las propiedades físicas relevantes de sistemas basados en láseres de estado sólido y la implementación experimental de técnicas de procesamiento de señales para la determinación del carácter de la señal, es decir, si presenta características clásicas o cuánticas a partir de la medición de alguna variable del sistema y el discernimiento entre señales caóticas y aleatorias.

Objetivos secundarios

Los objetivos secundarios son los siguientes: construcción de láser de estado sólido bombeado por diodos láser para la obtención de señales clásicas experimentales; diseño y desarrollo de programas para el control y automatización para la adquisición tanto de señales cuánticas como clásicas; implementación de algoritmos para el procesamiento de señales.

Desarrollo de la tesis por capítulos

En esta tesis, en el capítulo 1, se comenzará por introducir al láser de estado sólido como elemento común de los sistemas físicos que se estudiarán para la generación de señales experimentales. Además, se detallarán las técnicas que se utilizaron para la detección como así también sobre la clasificación de dichas señales de salida del láser. En el capítulo 2, se referirá a los fundamentos teóricos que serán la base para el procesamiento de las señales bajo estudio. Se mencionarán los enfoques de aleatoriedad y aproximación aplicable. En el capítulo 3, se presentarán los arreglos experimentales de los sistemas físicos relevantes. Luego, en el capítulo 4, se realizará una descripción detallada de la automatización, control y procesamiento de señales. Finalmente, se presentarán resultados, conclusiones y consideraciones para el futuro.

CAPÍTULO 1:

SEÑALES DE SALIDA DE LÁSERES DE ESTADO SÓLIDO

1.1 Introducción

Un láser está compuesto por tres componentes básicos: un mecanismo de bombeo, un medio amplificador y un medio de realimentación. El mecanismo de bombeo es la forma que tiene el sistema para recibir la energía necesaria para sostener la emisión láser, puede ser muy variada. El segundo componente primordial de un láser es el medio activo o medio amplificador, que es el que recibe la energía del bombeo y la transfiere al haz láser que genera. Este medio activo, puede ser un medio sólido (cristalino o amorfo), un líquido o un plasma. El último elemento con que cuentan la mayoría de los láseres es un mecanismo de realimentación de la radiación (luz) emitida por el amplificador, permitiendo de esta manera, que el láser adquiera sus características distintivas. En el caso de luz, la realimentación, se logra mediante espejos que reinyectan una parte de la señal nuevamente en el amplificador de luz, dicho juego de espejos (dos o más) que se encuentran alineados de manera tal que la luz emitida por el amplificador se refleja sobre sí misma recorriendo el mismo camino óptico varias veces, se denomina cavidad resonante. Se coloca uno de los espejos de forma tal que tengan una reflectividad menor que el 100%, de tal manera que parte de la luz que está oscilando salga de la cavidad produciendo un haz colimado que es lo que se mide como la salida del láser [2].

Los diferentes tipos de medios activos permiten categorizar los láseres según el material con el que funcionan y, en consecuencia, sus características generales resultantes. Los láseres semiconductores poseen gran eficiencia y permiten generar haces de gran potencia, pero sus perfiles espaciales son de calidad pobre. Los láseres gaseosos permiten obtener emisiones de una gran variedad de longitudes de onda, haces de buena calidad óptica y de alta potencia. Los láseres de estado sólido emplean cristales o vidrios dopados con iones metálicos como medio activo y siendo que presentan líneas espectrales de fluorescencia definidas, fuertes bandas de absorción y alta eficiencia cuántica resultan particularmente útiles para generar haces de alta calidad, estables o con alta pureza espectral [3]. Las señales experimentales que se estudiarán en esta tesis poseen como elemento común al láser de estado sólido.

Históricamente, la búsqueda de los láseres comenzó como una extensión en el empleo de técnicas en la región de microondas del espectro electromagnético. Por ese entonces, ya existía un dispositivo de base, el cual, realizaba la amplificación en el área de las microondas por la emisión estimulada de la radiación, el conocido "maser" (*microwave amplification by stimulated emission of radiation*), construido en 1954 por C. Townes. Pocos años después, en la década del 60,

T. Maiman logró el desarrollo del primer láser, que en honor a su predecesor maser, lo denominó "laser" por amplificador de luz por emisión estimulada de la radiación, cuyas siglas en inglés corresponden a *light amplification by stimulated emission of radiation* [4]. Este primer láser utilizó un cristal de rubí como medio amplificador y una lámpara flash como fuente de energía. Al año siguiente, A. Javan, W. Bennett y D. Harriot de los Laboratorios Bell, crearon el láser de estado gaseoso utilizando una mezcla de gases de Helio y Neón. En el mismo laboratorio, L. F. Johnson y K. Nassau demostraron el primer láser de Neodimio, que desde ese entonces se convirtió en el más confiable de los láseres. Luego, en 1962, llegó el láser semiconductor, por R. Hall de los Laboratorios de Investigación de General Electric. Un año después, C.K.N. Patel, nuevamente de los Laboratorios Bell, descubrió los láseres infrarrojos de dióxido de Carbono, uno de los más potentes láseres disponibles en la actualidad. Años subsiguientes, surgieron los láseres de iones, en vapor de mercurio; los láseres líquidos, utilizando un tinte orgánico disuelto en solvente. En la década del 70, se desarrollaron los láseres de vacío en ultravioleta; se demostró el primer amplificador láser libre de electrones que operaba en la longitud de onda del infrarrojo en láseres de CO₂. En 1979, se obtuvo un láser de salida ampliamente sintonizable a partir de láseres de estado sólido cuyo material se denominó alejandrita. Hacia 1985, surgieron los primeros láseres de rayos-X suaves. En los años posteriores, le siguieron los láseres de titanio zafiro; el diodo láser azul-verde y los láseres de cascada cuántica. En 1996, se desarrolló el diodo láser azul en los materiales basados en Nitruro de Galio [5].

En su etapa inicial, los láseres eran una herramienta de difícil acceso, por ejemplo, los primeros láseres de rubí, solo se utilizaban para la investigación, siendo casi de aplicación nula en las industrias. Esto se debía por la elevada inversión monetaria que implicaba el uso de dicha tecnología, que se sumaba al desconocimiento de esta novedosa fuente de radiación. Tampoco resultaba atractiva por la baja frecuencia de repetición que ofrecían los láseres de aquella época. Por su puesto, con el avance de la tecnología, se incorporaron los láseres sólidos bombeados por diodos reemplazando al viejo bombeo por lámpara flash, que permitieron mejorar el rendimiento, ofreciendo además una reducción significativa en el costo del equipamiento, descentralizando así su uso casi exclusivo para algunos sectores, abriéndose al mercado en forma masiva. A lo largo del tiempo, se fueron explorando nuevos materiales para nuevas aplicaciones en láseres, como el Nd: YVO₄ (ortovanadato de Itrio dopado con Neodimio).

En la actualidad, los láseres sólidos se utilizan no solo en la investigación, como ya hemos mencionado con anterioridad (estudio de la dinámica no lineal, mecánica cuántica, etc.), sino también en áreas médicas: en cirugía en general o para especialidades como cortar, coagular o remover tejido. En la medicina moderna, las aplicaciones de terapias láser son atractivas debido a su mínimo efecto invasivo. Se utiliza para el tratamiento y diagnóstico como diferentes tipos de cáncer, litotricia, oftalmología, etc. [6]; en área de las comunicaciones, para comunicaciones por medio de fibras ópticas para cubrir largas distancias con baja pérdida, para comunicaciones en las redes bajo el agua, en comunicaciones espaciales, radares y satélites; en el área industrial, para cortes de vidrio, en la industria automotriz, para el tratamiento de calor, para establecimientos para la lectura de productos; en el área militar, para telemetría, para criptografía (también para áreas bancarias), LIDAR (*Laser Imaging Detection and Ranging*), etc.

1.2 Señales de salida

A partir del análisis de señales de salida experimentales, se descubrirá el comportamiento de dos sistemas físicos complejos. En esta sección, se centrará en dos tipos de señales, cuánticas y clásicas, en base a dos arreglos experimentales diferentes.

El primer arreglo experimental, permite la realización de experimentos en el área cuántica de pares de fotones entrelazados, que es el elemento básico para experimentos fundamentales de mecánica cuántica, aplicaciones en comunicaciones, QKD (*Quantum Key Distribution*), generación de números al azar, etc. El fenómeno de entrelazamiento es un fenómeno que solamente se aprecia en las señales cuánticas y se puede identificar a partir del modo de adquisición *time stamping* mediante fotodiodos de avalancha en modo geiger (SPCM, *single photon counting module*).

El segundo arreglo experimental, es un sistema complejo que es representativo de muchos problemas de la naturaleza. Los sistemas caóticos se encuentran en la naturaleza y en particular en muchos láseres de uso masivo. Se presentará el estudio de señales clásicas en el esquema de automatización aplicado al láser sólido con modulación de pérdidas a partir de series temporales mediante fotodiodos pin, pudiendo determinar características fundamentales del sistema.

Los motivos de la elección de estos tipos de señales para nuestro trabajo se fundamentan básicamente, por un lado, porque son dos sistemas importantes en diversos campos tanto de investigación como en aplicaciones. Por otro lado, las señales que se generan son aleatorias o caóticas, señales más difíciles de caracterizar en contraposición con las señales periódicas que se caracterizan por su simpleza y predictibilidad, por lo tanto, más fáciles de describir.

1.3 Detección de señales

En esta sección, debido a las desigualdades con respecto a las características de las señales que se adquieren en forma experimental, se deben elegir dos formas distintas para la adquisición de valores experimentales: el *time stamping*, que permite generar listas con los tiempos en que se han detectado ciertos sucesos y las series temporales, que habilitan a realizar una sucesión de valores ordenados cronológicamente.

Para el caso de las señales experimentales cuánticas, se tiene que los fotones aparecen en el sistema de detección de a uno a la vez. Al encontrarse en un nivel cuántico de análisis, es preciso efectuar el método de *time stamping*, siendo los sucesos la detección de fotones. En cambio, para el estudio de señales experimentales clásicas, se puede omitir el estudio de los fotones, y extraer las propiedades globales de la señal, como lo es la intensidad de los pulsos o su separación temporal. Por lo cual, dependiendo del tipo de señal experimental, será más conveniente uno u otro método de adquisición.

1.3.1 *Time stamping*

Las desigualdades de Bell imponen un límite a la correlación entre observaciones de fotones realizadas en estaciones remotas, y esta correlación involucra necesariamente medir el número de coincidencias obtenidas. Dos detecciones suman como una coincidencia sólo si su separación temporal es menor que cierta ventana de coincidencias T_w . Consecuentemente, el número de coincidencias (y por lo tanto la correlación) depende inevitablemente del valor de T_w . En la mayoría de los experimentos realizados el valor de T_w es fijo, dado por el diseño de la electrónica de conteo de coincidencias, página 20 de [7].

Para el estudio de procesos dinámicos o correlaciones (entre eventos detectados) con resolución temporal es necesario conocer en que instante de tiempo los fotones fueron detectados. Este modo de adquisición conocido como *time tagging* o *time stamping* permite generar listas con los tiempos de llegada de los fotones que se han detectado.

Estos datos, se pueden procesar a posteriori, con la libertad de variar a voluntad la ventana de coincidencias. En la tabla 1, se puede visualizar el registro de una lista con valores experimentales de *time stamping* con los tiempos de llegada de los fotones detectados con un TDC (*Time to Digital Converter*). Estos valores, corresponden a la estación A, la unidad de tiempos es 1ps y el tiempo inicial es arbitrario. En este ejemplo, la diferencia entre el fotón 2 y el fotón 1 es de 20.284 μ s.

Listado de <i>time stamping</i>	
1	4.4461063526823000e+15
2	4.4461063729663000e+15
3	4.4461077329890000e+15
4	4.4461080876813000e+15
5	4.4461088129474000e+15
6	4.4461088277625000e+15
7	4.4461089378307000e+15
8	4.4461094080174000e+15
9	4.4461100648299000e+15
10	4.4461110328960000e+15

Tabla 1. Listado de 10 valores de *time stamping* de la estación A.

1.3.2 Series temporales

Para el estudio de la dinámica no lineal, la forma de estudiar los sistemas es a través de la extracción de la información a partir del análisis de las series temporales del sistema. Se podría decir que las series temporales son una sucesión de valores ordenados cronológicamente.

Por ejemplo, se pueden generar distintos tipos de series temporales a partir de los valores de una señal de salida del láser. En la figura 1. a, se puede observar una gráfica de una señal experimental: en color magenta, se indican los picos de amplitud de dicha señal; en color azul, las distintas muestras de amplitud de señal y en color cian, una curva trazada con dichas muestras. En la figura 1. b, se puede visualizar una porción de señal, pero amplificada.

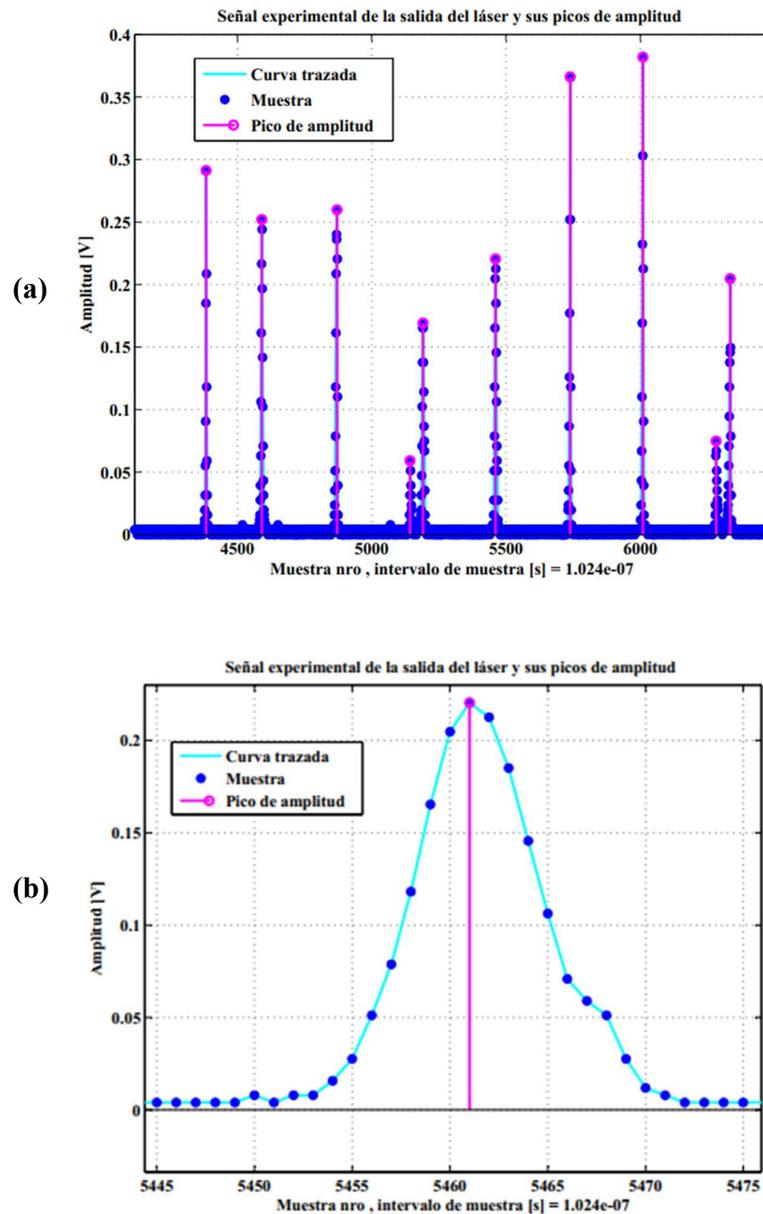


Figura 1. (a) Gráfica de parte de la señal de salida del láser con modulación de pérdidas y la detección de sus picos de amplitud. (b) Gráfica de la señal amplificada de salida del láser con modulación de pérdidas y la detección de sus picos de amplitud amplificada para un pulso de señal.

Las series temporales se pueden generar a partir de una señal como la mencionada anteriormente. Se pueden obtener: series temporales de amplitud de picos y series temporales de diferencia de tiempos entre picos.

En las series temporales de amplitud, los picos de amplitud de la señal de salida del láser corresponden a la variable intensidad del pulso láser. En la figura 2, se muestra la gráfica de una parte de la serie temporal de amplitud de la señal de la figura 1.

En las series temporales de la diferencia de tiempos, la variable es la diferencia de tiempo entre dos picos sucesivos de amplitud. A partir del análisis de estas series temporales, se pueden develar propiedades físicas de los sistemas. En la figura 3, se muestra la gráfica de una parte de la serie temporal de diferencias de tiempo de la señal de la figura 1.

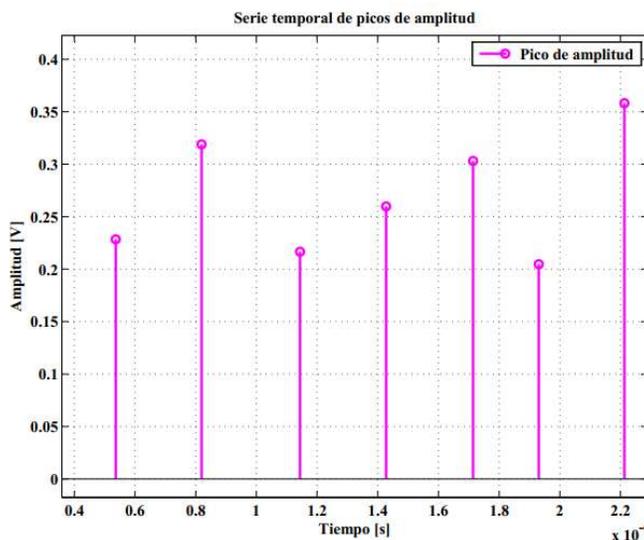


Figura 2. Gráfica de una parte de la serie temporal de amplitud de la señal de la figura 1.

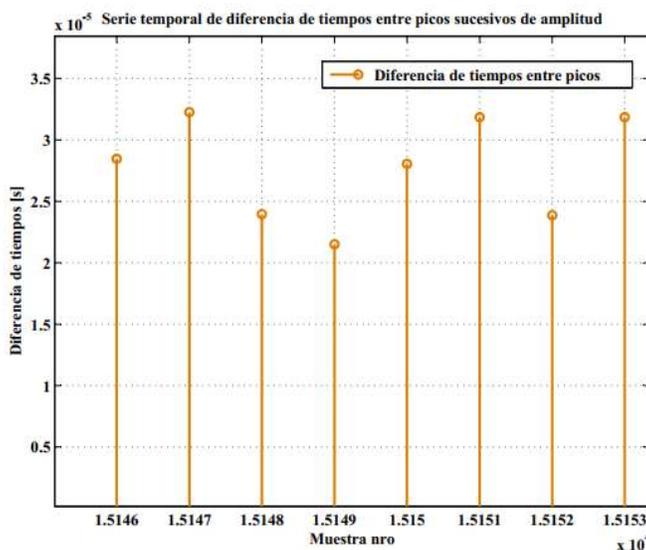


Figura 3. Gráfica de una parte de la serie temporal de diferencia de tiempos entre picos sucesivos de amplitud de la señal de la figura 1.

El tiempo de muestreo de la señal juega un rol importante en cuanto a la obtención de la forma de la señal y, por lo tanto, a la detección de la amplitud máxima del pulso de la señal de la salida del láser. Poder obtener una cantidad razonable de muestras en el pulso de la señal experimental, es esencial para el análisis de esta. Es por ello, que se debe elegir o definir dicha cantidad óptima de muestras por segundos necesaria en la adquisición de la señal. Luego de diversas mediciones y análisis de señales experimentales, se descubrió que 8 muestras por pulso eran suficientes para la reconstrucción de la forma del pulso de la señal de salida del láser. En cuanto a términos de precisión, las series temporales tanto de amplitud de picos de señal como de diferencias de tiempo entre dichos picos dependen del muestreo de la señal. A mayor muestreo, mayor precisión se tendrá, ya que el pico de amplitud de señal se podrá determinar con mayor exactitud. Esto implica que la diferencia de tiempos entre dichos picos también será más precisa. Sin embargo, el aumento en la cantidad de muestreo involucra el requerimiento de hardware más avanzado (osciloscopios con mejor frecuencia de muestreo). Consecuentemente, las series se convierten en series "pesadas", con mayor volumen de datos. Por lo cual, por practicidad, se intenta buscar un equilibrio entre simplicidad (cuanto menor sea el número de muestreo, menor volumen de datos ocupará la serie y más rápido será el procesamiento de la serie) y precisión (sobre muestrear la señal, permite una buena reproducción de la señal. Sin embargo, genera un alto volumen de datos en la serie y un desperdicio en los recursos). Existen métodos como el recurso del polinomio u otros métodos de interpolación [3], es decir, a partir de las muestras, se reconstruye con un polinomio la forma del pulso y se corrige el máximo del pico del pulso de señal. Esto permite mejorar la precisión sin el aumento del muestreo, sin embargo, el precio es el tiempo de cálculo. Para aplicaciones en tiempo real, en donde se adquieren los datos constantemente, no resulta eficiente esta interpolación de datos para su implementación debido a que esto implicaría un alto costo de recursos en tiempo de procesamiento.

En este trabajo, será indistinto utilizar series temporales de picos de amplitud o diferencia de tiempos entre picos. Tanto la figura 4 como en la figura 5, pertenecen a una misma señal experimental, la diferencia radica en la tasa de muestreo. En la figura 4, la tasa de muestreo es de 2 MS/s, para este caso en particular, se puede observar en la gráfica, que existe solo una muestra por pulso, por lo cual, la reconstrucción de la forma de la señal es inadecuada y, por lo tanto, generar una serie temporal de picos de amplitud o diferencia de tiempos entre dichos picos con baja cantidad de muestras por pulso, sería muy imprecisa. En cambio, si se aumenta el muestreo a 10 MS/s, como puede ver en la figura 5, se observa una cantidad de muestras suficientes para la reconstrucción de la señal y, por lo tanto, la obtención de una serie temporal con mayor precisión.

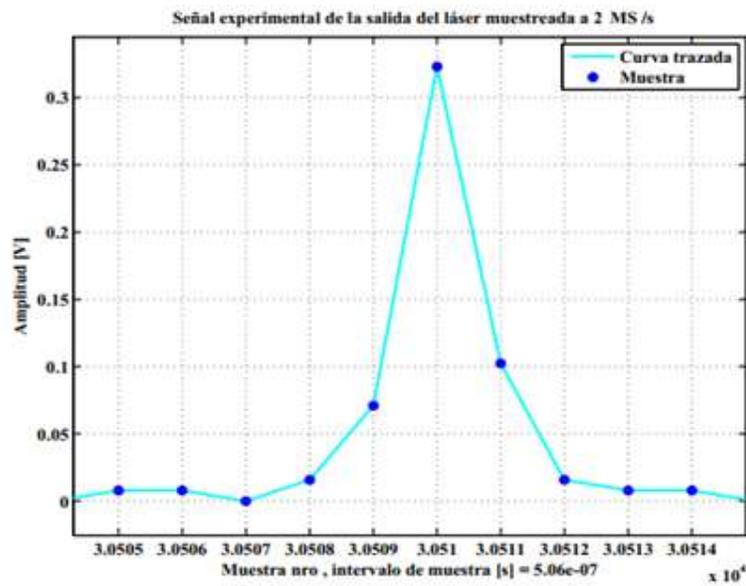


Figura 4. Señal experimental muestreada a 2 MS/s. Los puntos azules son las diferentes muestras. En color cian, se realiza el trazado de las muestras.

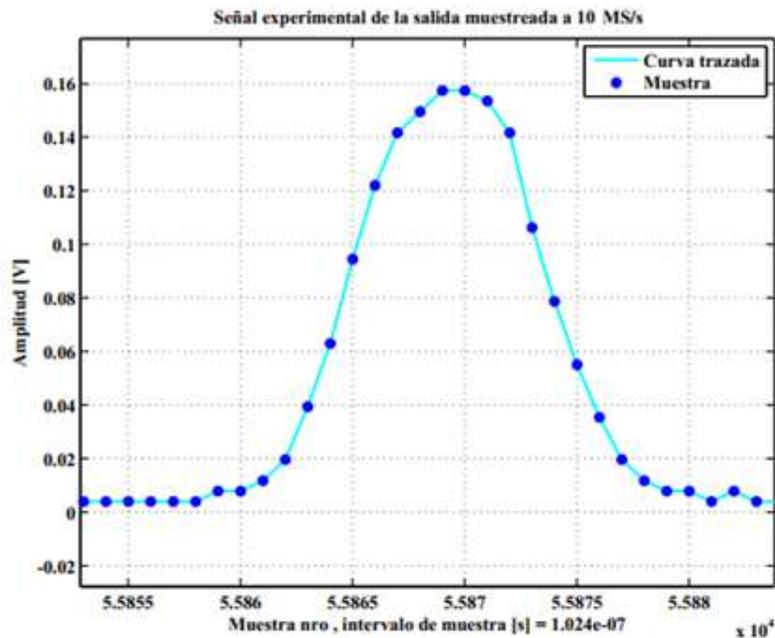


Figura 5. Señal experimental muestreada a 10 MS/s. Los puntos azules son las diferentes muestras. En color cian, se realiza el trazado de las muestras.

A partir de la señal muestreada, que se obtiene luego de haber elegido una tasa de muestreo adecuada (como se ha mencionado con anterioridad, con 8 muestras por pulso, resulta una buena reconstrucción de la señal experimental), se puede observar, como también sucede en la figura 5, que no todas las muestras de la señal aportan información relevante. Por ejemplo, en este caso en particular, las muestras por debajo del umbral 0.03 V de la gráfica de la figura 5, no son de importancia para nuestro estudio, ya que solo nos interesa aquellas muestras que constituyen los pulsos de señal para la detección de los picos de amplitud de cada pulso. Por lo cual, dichos datos pueden filtrarse, omitiendo su salvaguarda y realizando un procesamiento más eficiente al manejar un menor volumen de datos. Todas las consideraciones anteriores (tasa de muestreo y filtrado de umbral), se deben tener en cuenta para la conformación de series temporales experimentales.

1.4 Tipos de señales

A partir de los modelos teóricos sobre los sistemas físicos estudiados podemos saber, a priori, qué tipo de señal podemos encontrarnos.

En el caso de un láser caótico, el modelo [8] predice comportamientos que, dependientes del parámetro de control, pueden ser periódicos o caóticos.

En el área cuántica, determinar si la secuencia de *outcomes* (las salidas de las mediciones) es aleatoria o no lo es, posee fuertísimas implicancias fundamentales. La verificación de la violación de la desigualdad de Bell implica abandonar la Localidad, es decir, los eventos que ocurren fuera del cono de luz de una medición no afectan su resultado, o Realismo, en donde las propiedades del mundo físico están bien definidas, aunque no se las observe. Una corriente, asume no Localidad, entonces para poder conservar el principio de *no signaling*, o sea, no se puede transmitir información a velocidad mayor a la de la luz, los *outcomes* no deben llevar información, es decir, deben ser *random* (aleatorios). Los fundamentos de la criptografía cuántica se encuentran en esta suposición. Por otro lado, la "ortodoxia" cuántica, la interpretación de Copenhagen, descarta el Realismo para conservar la Localidad. Otra alternativa, es que exista una dinámica no ergódica, preservando así Realismo y Localidad. La ergodicidad es una condición necesaria (a veces olvidada) para deducir las desigualdades de Bell. Es entonces de crucial importancia, tanto de nivel teórico como en las aplicaciones tecnológicas, determinar si las series de *outcomes* son aleatorias o no.

Existen herramientas muy poderosas para descubrir señales periódicas, como la transformada de Fourier, en donde los picos de la transformada corresponden a la frecuencia de la señal periódica.

Sin embargo, una problemática de análisis de señales es la discriminación y descripción en forma correcta de las señales que no presentan periodicidad y que "a la vista" se presentan como "aperiódicas", "bruscas", "irregulares" o lo que vulgarmente se denomina "caóticas". En las secciones siguientes, se centrará en los métodos para poder discernir e identificar señales caóticas y señales aleatorias.

1.4.1. Señales periódicas

En una señal periódica se presentan patrones repetitivos durante lapsos de tiempo que se conocen como el periodo de la señal. Una característica interesante de este tipo de señal periódica es que, al calcular el espectro de la transformada de Fourier, presentará picos bien definidos. En cambio, no ocurre así con las señales que pudieran resultar caóticas, cuyo espectro tiene aspecto ancho, pero que deberán completarse con otro tipo de análisis para confirmar dicha caoticidad. En la figura 6, se muestra la gráfica de una señal periódica experimental obtenida del láser de modulación de pérdidas.

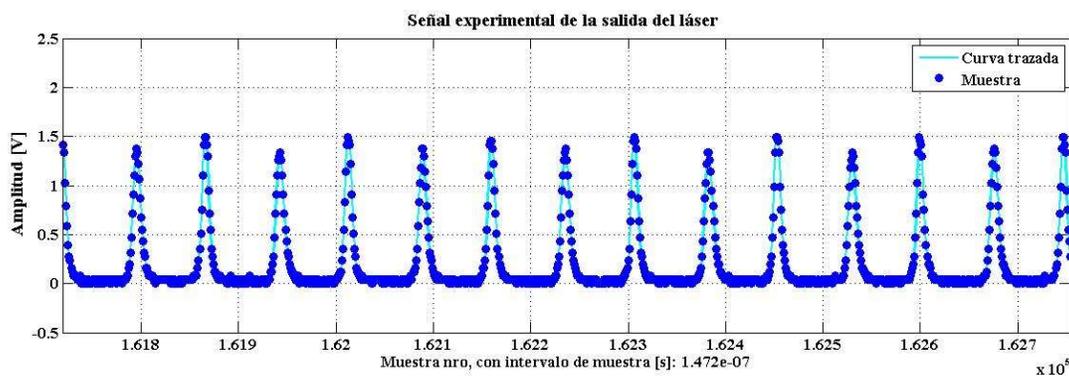


Figura 6. Gráfica de una parte de una señal experimental de la salida del láser con modulación de pérdidas. Esta señal se caracteriza por ser periódica.

1.4.2. Señales caóticas

El estudio de la dinámica caótica se remonta a inicios del siglo XX con el trabajo del matemático Henri Poincaré. La motivación de Poincaré fue en parte proporcionada por el problema de las órbitas de tres cuerpos celestes que experimentan atracción gravitatoria mutua, por ejemplo, una estrella y dos planetas. Al tener en cuenta el comportamiento de las órbitas derivados de conjuntos de puntos iniciales (en lugar de centrarse en órbitas individuales), Poincaré fue capaz de demostrar que órbitas muy complejas, ahora llamadas caóticas, eran posibles. Se destacan, en una época posterior, los trabajos en este campo de los matemáticos G. Birkhoff, en la década de 1920, M.L.

Cartwright y J.E. Littlewood en la década de 1940, S. Smale en la década de 1960, de los matemáticos soviéticos, en particular A. N. Kolmogorov y unos años después, E. Lorenz quien no estaba familiarizado con el trabajo de Poincaré en sus inicios en la meteorología, estudió un sistema muy sencillo para modelar el clima, empleando su computadora para sus simulaciones y comenzó con lo que actualmente se conoce como caos, fue distinguido por ser el autor del “Efecto Mariposa” [9].

A pesar del conocimiento matemático que existía sobre el tema caos, en la parte de los sistemas físicos reales, no fue aprovechado hasta hace relativamente poco. Las razones, se fundamentan por un lado porque muchas veces los artículos matemáticos resultan de difícil lectura para científicos de otras áreas y, por otra parte, los teoremas que se habían probado a menudo no eran lo suficientemente consolidados como para convencer a los investigadores en otros campos, que este tipo de comportamiento fuese de importancia en sus sistemas. En la actualidad, esta situación se ha revertido drásticamente y gran parte del crédito por esto puede atribuirse a la amplia gama de soluciones numéricas de sistemas dinámicos que se obtuvieron con el empleo de las computadoras. Usando este tipo de soluciones, el carácter caótico de las evoluciones temporales en situaciones de importancia práctica se ha vuelto dramáticamente claro.

Al visualizar señales del tipo caóticas, adjetivos como salvaje, turbulento, y al azar vienen a la mente. A pesar de su complejidad, los movimientos caóticos pueden provenir de sistemas físicos muy simples. Según los modelos teóricos que describen el láser, la transformada de Fourier se verá con un espectro ancho, sin picos definidos. Otra característica de las señales caóticas es la extrema sensibilidad a las condiciones iniciales. Podemos ubicar las señales caóticas entre las totalmente predecibles, regulares, periódicas o cuasi periódicas y las totalmente irregulares, estocásticas, conocidas como ruido y que son completamente impredecibles. El caos es irregular en el tiempo, en cierto modo predecible y tiene una estructura en el espacio de fase.

En la figura 7, se muestra la gráfica de una señal caótica experimental obtenida del láser de modulación de pérdidas.

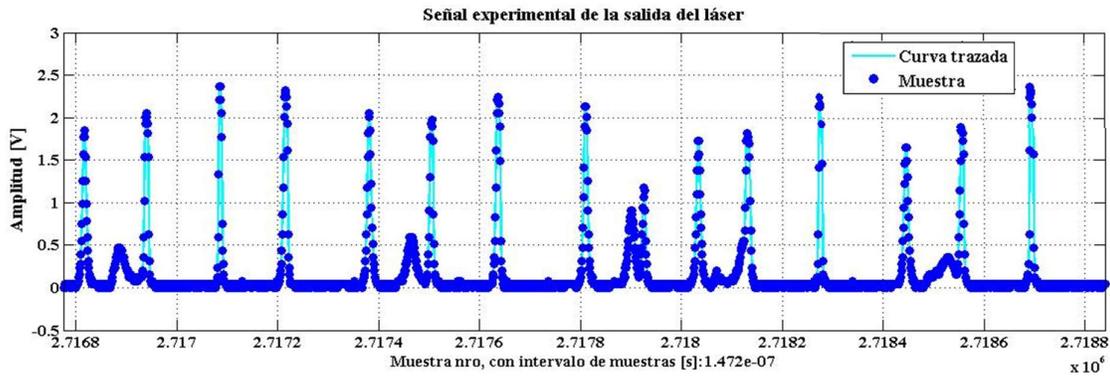


Figura 7. Gráfica de una parte de una señal experimental de salida del láser con modulación de pérdidas. Esta señal se caracteriza por ser caótica.

1.4.3 Señales aleatorias (*random*)

Que un evento sea aleatorio o *random* implica que no es predecible. En un mundo determinista, es decir, regido por una teoría determinista, la aleatoriedad, en principio, proviene de la falta de información. Si conociera con toda precisión las condiciones iniciales de un sistema, podría predecir con exactitud su evolución a cualquier tiempo posterior. Esta pretenciosa afirmación del determinismo falla ostensiblemente ya en el problema de tres cuerpos, que no tiene solución analítica. A partir de ecuaciones deterministas, entonces, podemos tener caos, conocido como caos determinista y solo es parcialmente predecible. Sin embargo, cualquiera que sea la razón para la aleatoriedad, siempre se puede realizar un estudio de probabilidades, ya que es una alternativa para abordar el problema y realizar una búsqueda de información de ese tipo de ocurrencias. Por ejemplo, lanzar al aire una moneda y observar, cuantas veces, es cara o seca, resultará que, a lo largo de muchas repeticiones experimentales, la mitad de las veces corresponderá a una de las caras de la moneda.

Un experimento, se denomina aleatorio cuando en una medición, no se puede predeterminar con cierta precisión y completitud el resultado de un ensayo. En el área de las probabilidades, para que un experimento se considere aleatorio depende de la precisión por la que se desea distinguir posibles resultados. Todo este estudio estadístico forma parte de la aleatoriedad estadística [10].

A su vez, también se tiene el estudio de la aleatoriedad algorítmica, que surge del estudio de la aleatoriedad de secuencias binarias. Por ejemplo, si se suponen dos secuencias como las que siguen:

101010101010101010101010101010101010... (I)

101101011101010111100001010100010111... (II)

Se podría intuir, que la secuencia I es más aleatoria que la secuencia II. El origen del estudio de la aleatoriedad algorítmica se remonta a principios del siglo XX, por Richard von Mises quien argumentó que una secuencia se contabiliza como aleatoria si todas las posibles infinitas subsecuencias satisfacen las leyes de los números largos (por ejemplo, si tienen en la misma proporción, la misma cantidad de 0s y 1s en el límite). Este comportamiento, es el que se espera encontrar intuitivamente en una secuencia aleatoria. Sin embargo, una secuencia como la (I), si bien satisface la ley de los números largos, no debería contar como aleatoria, ya que contienen subsecuencias fáciles de describir [11]. A partir de la aplicación de reglas para la detección de secuencias aleatorias, surge todo el estudio de la aleatoriedad algorítmica. Se puede mencionar la complejidad de Kolmogorov, en donde una secuencia binaria es aleatoria si es la más corta posible que se puede obtener respecto a otras equivalentes producidas por programas de computadora. Existen métodos que permiten mensurar dicha complejidad algorítmica, como el trabajo de F. Kaspar y H.G. [12], en donde se implementa un algoritmo para el cálculo de la complejidad de una cadena binaria finita. La idea de aleatoriedad radica en el sentido de incompresibilidad en donde es imposible comprimir la secuencia a otra que no sea de la longitud de esta. Además, existe un conjunto de pruebas estadísticas del NIST (*National Institute of Standards and Technology*), en total 15, que permiten evaluar la aleatoriedad de generadores de secuencia de números aleatorios y pseudoaleatorios [13]. En la figura 8, se muestra una señal experimental de la salida del láser con modulación de pérdidas aleatoria.

En esta tesis, a partir de señales de salida del láser experimentales, se obtendrán series temporales. A las cuales, se las convertirá en secuencias binarias a partir de un valor de umbral para el cual se decide que, para valores superiores o iguales a este, se le asignará un uno y para valores inferiores, un cero. El valor del umbral que se elegirá será la mediana de los datos. Esto permitirá evaluar tanto la complejidad de Kolmogorov como así también los tests del NIST, el primero permitirá caracterizar la aleatoriedad desde el punto de vista algorítmico y el segundo, respecto del punto de vista estadístico.

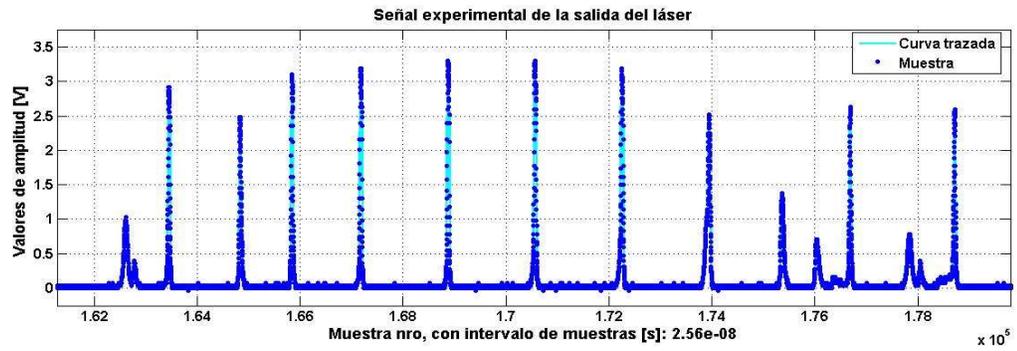


Figura 8. Gráfica de una parte de la señal experimental de la salida del láser con modulación de pérdidas. Esta señal se caracteriza por ser aleatoria.

CAPÍTULO 2:

FUNDAMENTOS DEL PROCESAMIENTO DE SERIES TEMPORALES

2.1 Introducción

En este capítulo, nos centraremos en los fundamentos teóricos que nos permitirán realizar el procesamiento de nuestras series temporales experimentales, tanto para la detección de señales caóticas como aleatorias. Por un lado, para la detección de señales caóticas, se efectuará el estudio de la reconstrucción a partir de series temporales de un atractor. Por otro lado, para la detección de señales aleatorias, se centrará en dos enfoques: el estudio estadístico de la aleatoriedad, en donde se detallarán los tests del NIST y el estudio algorítmico de la aleatoriedad, en donde se describirán los métodos de aproximación para la medición de la complejidad de Kolmogorov.

La complejidad de Kolmogorov, en la teoría de la computación, es una medida de la cantidad de recursos computacionales necesarios para describir una cierta cantidad de información, debe su nombre a Andrei Kolmogorov. Esta complejidad, también se conoce como complejidad de Kolmogorov-Chaitin, complejidad descriptiva, complejidad estocástica o entropía algorítmica.

Andrei Kolmogorov (1903-1987), fue un matemático ruso que realizó importantes aportes en los campos de la teoría de la probabilidad y la topología. Fue el fundador de la teoría de la complejidad algorítmica. La teoría de la complejidad de Kolmogorov contiene una matemática muy avanzada y difícil de comprender para aquellos que no se encuentran en el área de las matemáticas [14].

Si se posee dos secuencias binarias y si se desea resolver cuál de las dos es más sencilla que otra, se deberá comparar si aquellas pueden reducirse a otras más simples. Las secuencias más complejas son aquellas para las que no existe una descripción (compresión) más pequeña. Realizar una comparación que no permita ambigüedades del lenguaje natural, implica fijar un lenguaje de programación, es decir, un lenguaje que utilizan las computadoras. Este tipo de análisis fue propuesto por Andrei Kolmogorov y el argentino-estadounidense Gregorio Chaitin, en donde se busca la longitud del programa más corto que produce la secuencia, medido en bits, cuando se lo ejecuta en una computadora (máquina universal de Turing). Esta medida, es una medida universal de complejidad porque utiliza el modelo universal de Turing y su generalidad, permite obtener innumerables propiedades intuitivas que se las suele relacionar con aleatoriedad (por ejemplo, la impredecibilidad y la irregularidad, características que nos interesan descubrir de nuestras series temporales experimentales). Sin embargo, esta generalidad trae como consecuencia que no exista un algoritmo que, dada una secuencia, prediga el tamaño más corto que genere la secuencia (complejidad algorítmica). Esto ocurre con la misma analogía por la que existen números no

calculables: porque las máquinas de Turing no pueden resolver el problema de la detención (es decir, no se sabe con certeza si el programa se detendrá o no), lo que implica que no se puede medir con absoluta exactitud la complejidad algorítmica de una secuencia. Es por ello, que la complejidad de Kolmogorov, como se define en su teoría, es incomputable a secas. Sin embargo, existen métodos de compresión que pueden medir la complejidad de Kolmogorov en forma estimada, identificando subsecuencias repetitivas dentro de la secuencia [15].

Abraham Lempel y Jacob Ziv, dos científicos israelíes en el área de la computación desarrollaron en 1976 la complejidad de Lempel-Ziv. Esta complejidad, se encuentra relacionada con la complejidad de Kolmogorov y se puede utilizar para medir la repetición de secuencias binarias. La complejidad aumentará a medida que crece la secuencia en longitud e irregularidad.

Todos estos fundamentos teóricos, serán la base para el tratamiento y creación de algoritmos, en MATLAB y LabVIEW, de las señales experimentales de nuestra tesis.

2.2 Reconstrucción a partir de una serie temporal de un atractor

Como se ya ha comentado con anterioridad, las señales caóticas se las ubica entre las totalmente predecibles, regulares, periódicas o cuasi periódicas y las totalmente irregulares, estocásticas, conocidas como ruido y que son completamente impredecibles. Sin embargo, a pesar de su comportamiento irregular, la dinámica caótica presenta una estructura definida en el espacio de fases. Un espacio de fases es una forma para representar un sistema dinámico. Posee tantas dimensiones como variables necesarias para especificar el estado del sistema original. Con cierta estructura vectorial, representa el conjunto de estados posibles en los que se puede encontrar el sistema modelado a lo largo del tiempo. La órbita de un estado particular se representa por una trayectoria en el espacio. Dicha representación permite una descripción cualitativa de la evolución temporal del modelo bajo estudio [16]. A partir de este espacio de fases, se pueden extraer innumerables propiedades del sistema, que permiten la modelización del sistema y la realización de predicciones. Estas estructuras características se denominan atractores. Para poder determinar si la serie temporal es caótica, se debe reconstruir el atractor, es decir, el conjunto de puntos, en el espacio de fases, por el cual evoluciona el sistema a partir de un observable escalar, la serie temporal [17]. El teorema de *embedding*, que surge a partir de estudios de F. Takens [18] y R. Mañé [19], nos proporciona el marco teórico y nos garantiza que el atractor reconstruido de esta manera es topológicamente equivalente al del sistema físico real. Así, los vectores en el espacio de fase d -dimensional reconstruido tendrán la forma como la ecuación 1:

$$x(t_j) = (x_i(t_j), x_i(t_j + \tau), \dots, x_i(t_j + (d - 1)\tau)) \quad (1)$$

donde $x_i(t_0), x_i(t_1), \dots, x_i(t_j)$ es la serie temporal, τ es lo que se conoce como tiempo retardo y d es la dimensión de *embedding* del atractor. Es decir, d es el número de elementos que debe tener el vector para poder desplegar el atractor en el espacio de fase en forma completa.

Si bien no existe un criterio unívoco, en esta tesis se elegirá a τ arbitrariamente como el primer mínimo de la información mutua de la serie temporal o 1 si la función es monótona, esto se hace por simplicidad de cálculo. Para determinar la dimensión de *embedding* se implementa el método de los falsos vecinos.

Supongamos un vector en el espacio de fase reconstruido como la ecuación 2:

$$y(k) = (s(k), s(k + T), \dots, s(k + (d - 1)T)) \quad (2)$$

donde $s(k)$ es un escalar observable y T es el tiempo sugerido por el criterio de la información mutua.

Si se examinan los vecinos más cercanos (*nearest neighbors*) en el espacio de fase del vector $y(k)$ como la ecuación 3:

$$y^{NN}(k) = (s^{NN}(k), s^{NN}(k + T), \dots, s^{NN}(k + (d - 1)T)) \quad (3)$$

Si este vector es un vecino verdadero de $y(k)$, provendrá de una vecindad de $y(k)$ en un sentido dinámico. Será el más cercano anterior o posterior en la órbita. Dado que el atractor de un sistema físico real es compacto en el espacio de fase, cada punto en el espacio de fase tendrá numerosos vecinos en la medida que se tenga datos suficientes. Si, por otra parte, el vector $y^{NN}(k)$ es un vecino falso de $y(k)$, llegará a su vecindad como consecuencia de la proyección desde una dimensión mayor, debido a que el atractor no está completamente desplegado en la dimensión d . Se puede aumentar la dimensión a $d+1$, y ver si el punto desaparece de la vecindad de $y(k)$, lo cual, revelará que se trataba de un falso vecino. Analizando cada punto $y(k)$ y preguntando qué dimensión elimina por completo los falsos vecinos, se remueven secuencialmente las intersecciones de órbitas de dimensiones más bajas, hasta quitar la última, en ese momento, se habrá identificado la dimensión más baja que permite desplegar el atractor, es decir, la dimensión de *embedding*. Habiendo descartado las señales periódicas con la FFT (*Fast Fourier Transform*), la existencia de una dimensión de *embedding* finita es un indicio de caos, ya que una señal aleatoria tiende a llenar todo el espacio de fase y, por lo tanto, al intentar emplear el método de falsos vecinos, se verá que los falsos vecinos nunca caen a cero.

Finalmente, una vez determinada la dimensión de *embedding*, se pueden determinar los exponentes de Lyapunov. En un sistema cuyo movimiento se vuelve caótico, dos puntos cercanos en el espacio de fase se apartan exponencialmente en el tiempo. Esta rápida separación de las órbitas en el tiempo, que se conoce como sensibilidad a las condiciones iniciales, es la manifestación precisa de las inestabilidades en el espacio de fase que conducen a los movimientos no periódicos que denominamos caos. Los exponentes de Lyapunov dan cuenta de la divergencia de las trayectorias en el espacio de fase. El número de estos exponentes es igual a la dimensión del espacio de fase: $\lambda_1, \lambda_2, \dots, \lambda_d$. El significado de los exponentes puede explicarse en forma intuitiva con argumentos geométricos. Si las trayectorias en el espacio de fase tienen la forma $e^{\lambda t}$ siendo t un índice temporal, $\lambda < 0$ indica trayectorias convergentes, mientras que trayectorias inicialmente muy cercanas serán divergentes si $\lambda > 0$. De este criterio se establece que para que un sistema sea globalmente estable, la sumatoria de todos los exponentes deben ser negativos. Si las orbitas en el espacio de fases o de *embedding* no divergen y el máximo exponente de Lyapunov es positivo, tenemos una clara indicación de caos. También se ve que la suma de todos los exponentes debe ser negativa para que el sistema sea disipativo.

En resumen, para que un sistema sea caótico debe cumplir con las siguientes condiciones:

- ✓ que haya una dimensión de *embedding* finita,
- ✓ que la sumatoria de los coeficientes de Lyapunov sea negativa,
- ✓ y que al menos un coeficiente de Lyapunov sea positivo.

2.3 Enfoques de Aleatoriedad

Anteriormente, se mencionó sobre la caracterización de la aleatoriedad a partir de dos enfoques: el estadístico y el algorítmico. En esta sección, se ahondará en detalle a cerca de los distintos tests del NIST, que permitirán evaluar la aleatoriedad desde el enfoque estadístico y, a partir de la definición de la complejidad de Kolmogorov, el enfoque algorítmico.

2.3.1 Tests del NIST

En la actualidad, para aplicaciones como, por ejemplo, la criptografía, requieren de la utilización de números aleatorios para la creación de las claves criptográficas. Habitualmente, estas claves son secuencias de números mediante el cual, en criptografía, se cifra la información. Muchos protocolos en el área de criptografía solicitan entradas aleatorias, por ejemplo, para la generación de firmas digitales o protocolos de autenticación.

Existen varios tests que, evaluando propiedades estadísticas de la secuencia, permiten evaluar su aleatoriedad. En nuestro trabajo, las secuencias bajo estudio serán las series temporales de amplitud de picos de señal experimental de la salida del láser. Debido a que para estos tests se requieren secuencias binarias, se debe binarizar la secuencia. Este se basa en el código kolbin.m, que se encuentra en el apéndice D.

Los tests del NIST, consisten en 15 tests que evalúan la aleatoriedad de secuencias binarias lo suficientemente largas:

1. *The Frequency (Monobit) Test:*

Este test se centra en la tasa de unos y ceros que posee la secuencia de n bits de longitud bajo análisis. En una secuencia aleatoria, se esperaría que la cantidad de unos como de ceros sean aproximadamente de la misma magnitud. Este test, evalúa cuan próxima se encuentra la fracción de unos cercana a la mitad. Es el test más importante, ya que todos los tests subsecuentes dependen de pasar este test.

Se realiza el cálculo de probabilidad, páginas 2-2 a 2-3 de [13], mediante la ecuación 4:

$$Pvalue = \operatorname{erfc}\left(\frac{s_{obs}}{\sqrt{2}}\right) \quad (4)$$

donde erfc es la función de error complementaria, s_{obs} es un estadístico del test. Si Pvalue es menor a 0.01, se considera que la secuencia es no aleatoria. En el caso contrario, se concluye que la secuencia es aleatoria.

Para este test se recomienda que cada secuencia a ser testada consista en un mínimo de 100 bits, es decir, $n=100$.

2. *Frequency Test within a Block:*

El objetivo de este test es partir de la secuencia de n bits de longitud, particionarla en M bloques y evaluar que la tasa de unos en cada M bloque en particular sea $M/2$. En el caso particular que M sea igual a 1, es análogo al test 1.

Se realiza el cálculo de probabilidad, páginas 2-4 a 2-5 [13], mediante la ecuación 5:

$$Pvalue = \Gamma\left(N/2, \chi^2(obs)/2\right) \quad (5)$$

donde Γ es la función gamma incompleta, N es el valor entero de $\frac{n}{M}$, χ^2 es un estadístico del test. Si Pvalue es menor a 0.01, se considera que la secuencia es no aleatoria. En el caso contrario, se concluye que la secuencia es aleatoria.

Para este test se recomienda que cada secuencia a ser testada consista en un mínimo de 100 bits, es decir, $n=100$. Se debe seleccionar un tamaño de bloque M tal que cumpla con las siguientes condiciones: $M \geq 20$, $M > 0.01n$ y $N < 100$.

3. *The Runs Test:*

El objetivo de este test es evaluar la cantidad total de secuencias ininterrumpidas de bits idénticos (*runs*). Estas subsecuencias idénticas, son de longitud k , y se encuentran acotados por bits opuestos a ambos extremos. El propósito de este test radica en determinar si el número de subsecuencias de bits idénticos tanto de unos o como de ceros, como de diferentes longitudes, se corresponden a una secuencia aleatoria. En esencia, este test determina si las oscilaciones entre ceros y unos son rápidas o lentas.

Se realiza el cálculo de probabilidad, páginas 2-5 a 2-7 [13], mediante la ecuación 6:

$$Pvalue = \operatorname{erfc} \left(\frac{V_{\pi} - 2n\pi(1-\pi)}{2\sqrt{2n\pi(1-\pi)}} \right) \quad (6)$$

donde V_{π} es un estadístico del test, π es una proporción de unos de la secuencia de entrada y n la cantidad de bits de la secuencia. Si Pvalue es menor a 0.01, se considera que la secuencia es no aleatoria. Caso contrario, se concluye que la secuencia es aleatoria.

Para este test se recomienda que cada secuencia a ser testada consista en un mínimo de 100 bits, es decir, $n=100$.

4. *Tests for the Longest-Run-of-Ones in a Block:*

Este test se enfoca en el número más largo de subsecuencias de unos dentro de los M bloques. El propósito es la determinación si dicha longitud es consistente con la que se esperaría en una secuencia aleatoria. Se debe hacer hincapié en que una irregularidad en la longitud del número más largo de subsecuencias de unos implica que también los hay en la de ceros, por lo cual, solo es necesario el testeo para el caso de unos.

Se realiza el cálculo de probabilidad, páginas 2-8 a 2-9 [13], mediante la ecuación 7:

$$Pvalue = \Gamma \left(\frac{K}{2}, \frac{\chi^2(obs)}{2} \right) \quad (7)$$

donde K se determina por los valores tabulados de M. χ^2 es un estadístico del *test*. Si Pvalue es menor a 0.01, se considera que la secuencia es no aleatoria. Caso contrario, se concluye que la secuencia es aleatoria.

Para este test se recomienda que cada secuencia a ser testeada consista en un mínimo de bits tabulado según valores de M.

5. *The Binary Matrix Rank:*

Este test se enfoca en el rango de submatrices disjuntas de la secuencia en su totalidad. El propósito es la verificación de dependencias lineales entre subsecuencias de longitudes prefijadas de la secuencia original.

Se realiza el cálculo de probabilidad, páginas 2-10 a 2-12 [13], mediante la ecuación 8:

$$Pvalue = e^{-\chi^2(obs)/2} \quad (8)$$

Si Pvalue es menor a 0.01, se considera que la secuencia es no aleatoria. Caso contrario, se concluye que la secuencia es aleatoria.

Para este test se recomienda que cada secuencia a ser testeada consista en un mínimo de bits de tal forma que n, la cantidad de bits de la secuencia sea $n \geq 38MQ$, donde M son las filas y Q es el número de columnas de la matriz seleccionada por el test (por lo general se elige un valor de 32, para cada uno).

6. *The Discrete Fourier Transform (Spectral) Test:*

Este test se enfoca en la altura de los picos altura de la transformada discreta de Fourier de la secuencia. El propósito de este test es detectar características periódicas, por ejemplo, la repetición de patrones cercanos unos a otros en la secuencia bajo análisis indicarían una desviación respecto a la suposición de aleatoriedad. La intención de este test es detectar el número de picos que excede al 95 por ciento del umbral es significativamente diferente al 5 por ciento.

Se realiza el cálculo de probabilidad, páginas 2-12 a 2-14 [13], mediante la ecuación 9:

$$Pvalue = \operatorname{erfc}\left(\frac{|d|}{\sqrt{2}}\right) \quad (9)$$

donde d es un coeficiente que depende de n, de N_o , el número de picos teóricos esperado al 95 por ciento menores a un umbral T y de N_1 , es el número de picos observados realmente, que son

menores a T . Si $Pvalue$ es menor a 0.01, se considera que la secuencia es no aleatoria. Caso contrario, se concluye que la secuencia es aleatoria.

Para este test se recomienda que cada secuencia a ser testada consista en un mínimo de bits de tal forma que n , la cantidad de bits de la secuencia sea $n \geq 1000$.

7. *The Non-overlapping Template Matching Test:*

Este test se enfoca en el número de ocurrencias de una subsecuencia con un patrón no periódico. El propósito es la detección de aquellas secuencias que poseen muchas ocurrencias de este tipo de subsecuencia con un patrón en particular. Se utiliza una ventana de m -bit para buscar un patrón específico de m -bit. Si no se encuentra el patrón, la ventana se desplaza un bit de posición. En el caso que se encuentre el patrón, la ventana se reinicia al bit siguiente del patrón hallado y se reanuda la búsqueda.

Se realiza el cálculo de probabilidad, páginas 2-14 a 2-17 [13], mediante la ecuación 10:

$$Pvalue = \Gamma\left(\frac{N}{2}, \frac{\chi^2(obs)}{2}\right) \quad (10)$$

para cada caso particular de *template*, donde N es el número de bloques independientes, para cada caso en particular. Si $Pvalue$ es menor a 0.01, se considera que la secuencia es no aleatoria. Caso contrario, se concluye que la secuencia es aleatoria.

Para este test se recomienda que cada secuencia a ser testada consista en un mínimo de bits de tal forma que la cantidad de subsecuencia en bits a testear sea $M > 0.01Nn$ y $N = \lfloor n/M \rfloor$.

8. *The Overlapping Template Matching Test:*

Este test es parecido al test 7 pero con la salvedad que, al encontrar el patrón, la ventana se desliza una posición antes de reanudar la búsqueda.

Se realiza el cálculo de probabilidad, páginas 2-17 a 2-19 [13], mediante la ecuación 11:

$$Pvalue = \Gamma\left(\frac{5}{2}, \frac{\chi^2(obs)}{2}\right) \quad (11)$$

Si $Pvalue$ es menor a 0.01, se considera que la secuencia es no aleatoria. Caso contrario, se concluye que la secuencia es aleatoria.

Para este test se recomienda que cada secuencia a ser testada consista en un mínimo de bits de tal forma que n , la cantidad de bits de la secuencia sea $n \geq 1000000$.

9. Maurer's "Universal Statistical" Test:

Este test se enfoca en la cantidad de bits entre patrones iguales (una medición de este tipo se relaciona con la longitud de una secuencia compresible). El propósito de este test es detectar por sí o por no si la secuencia se puede comprimir significativamente sin la pérdida de la información. Una secuencia significativamente compresible se considera no aleatoria.

Se realiza el cálculo de probabilidad, páginas 2-20 a 2-23 de [13], mediante la ecuación 12:

$$Pvalue = \operatorname{erfc}\left(\frac{f_n - \operatorname{expectedValue}(L)}{\sqrt{2}\sigma}\right) \quad (12)$$

donde f_n , es un estadístico del test, los valores de $\operatorname{expectedValue}(L)$ y σ , son valores pre computados tomados de tablas. Si Pvalue es menor a 0.01, se considera que la secuencia es no aleatoria. Caso contrario, se concluye que la secuencia es aleatoria.

Para este test se recomienda que cada secuencia a ser testeada consista en un mínimo de bits de tal forma que n , la cantidad de bits de la secuencia, se corresponda a valores tabulados según L y Q , donde L es la longitud de cada bloque y Q el número de bloques en la secuencia de inicialización.

10. The Linear Complexity Test:

Este test se enfoca en la longitud del registro de corrimiento de realimentación lineal LFSR (*Linear Feedback Shift Register*). El propósito de este test es determinar si una secuencia es lo suficientemente compleja para considerarla aleatoria. Las secuencias aleatorias se caracterizan por más largas LFSRs. Un LFSR corto, implica no aleatoriedad.

Se realiza el cálculo de probabilidad, páginas 2-24 a 2-26 [13], mediante la ecuación 13:

$$Pvalue = \Gamma\left(\frac{K}{2}, \frac{\chi^2(\operatorname{obs})}{2}\right) \quad (13)$$

donde K es el grado de libertad. Si Pvalue es menor a 0.01, se considera que la secuencia es no aleatoria. En el caso contrario, se concluye que la secuencia es aleatoria.

Para este test se recomienda que cada secuencia a ser testeada consista en un mínimo de bits de tal forma que n , la cantidad de bits de la secuencia sea $n \geq 1000000$.

11. The Serial Test:

Este test se enfoca en la tasa de todos los posibles patrones de m -bit que se superponen en la secuencia. El propósito de este test es determinar si el número de ocurrencias de las 2^m patrones

superpuestos de m-bit son aproximadamente de igual manera que se esperarían para una secuencia aleatoria. Las secuencias aleatorias tienen uniformidad. Esto es, para cada m-bit patrón posee la misma oportunidad de aparecer al igual que cada otro m-bit patrón. Notar que para m igual a 1, este test es equivalente al *Frequency test*.

Se debe realizar dos cálculos de probabilidad, páginas 2-26 a 2-28 [13], mediante las ecuaciones 14 y 15:

$$Pvalue1 = \Gamma(2^{m-2}, \nabla\Psi_m^2) \quad (14)$$

$$Pvalue2 = \Gamma(2^{m-3}, \nabla^2\Psi_m^2) \quad (15)$$

donde $\nabla\Psi_m^2$ y $\nabla^2\Psi_m^2$, son estadísticos del test que miden cuán buenas son las tasas observadas de los m-bit patrones que se corresponden con las tasas de los m-bit patrones esperados. Para ambos Pvalue se debe analizar si son menores a 0.01, se considera que la secuencia es no aleatoria. En el caso contrario, se concluye que la secuencia es aleatoria.

Para este test se recomienda que cada secuencia a ser testeada consista en un número de bits n de tal forma que $m < \lfloor \log_2 n \rfloor - 2$, donde m es la longitud de bits de cada bloque.

12. The Approximate Entropy Test:

Tal como el *Serial test*, este test se enfoca en la tasa de todos los posibles m-bit patrones que se superponen en la secuencia. El propósito de este test es comparar la tasa de los bloques de superposición de dos longitudes adyacentes o consecutivas (m y m+1) respecto el resultado esperado para una secuencia aleatoria.

Se debe realizar el cálculo, páginas 2-29 a 2-31 [13], mediante la ecuación 16:

$$Pvalue = \Gamma\left(2^{m-1}, \frac{\chi^2}{2}\right) \quad (16)$$

donde χ^2 es el estadístico del test. Si Pvalue es menor a 0.01, se considera que la secuencia es no aleatoria. Caso contrario, se concluye que la secuencia es aleatoria.

Para este test se recomienda que cada secuencia a ser testeada consista en un número de n bits de tal forma que $m < \lfloor \log_2 n \rfloor - 5$, donde m es la longitud de bits de cada bloque.

13. *The Cumulative Sums (Cusums) Test:*

Este test se enfoca en la máxima excursión (a partir de cero) de una corrida aleatoria definida por la suma acumulativa de los dígitos ajustados (-1, +1) de una secuencia. El propósito de este test es determinar si la suma acumulativa de las secuencias parciales que ocurren dentro de las secuencias testeadas es muy larga o relativamente pequeña para el comportamiento que se espera para una secuencia aleatoria de una suma acumulativa. Esta suma acumulativa se considera como una corrida aleatoria. Para secuencias aleatorias, las excursiones de una corrida aleatoria deberían ser cercanas a cero. Para ciertos tipos de secuencias no aleatorias, las excursiones de las corridas aleatorias a partir de cero serán largas.

Se debe realizar el cálculo, páginas 2-31 a 2-33 de [13], mediante la ecuación 17:

$$Pvalue = 1 - \sum_{k=\frac{(-n+1)}{z}}^{\frac{(n-1)}{z}} \left[\phi \left(\frac{(4k+1)z}{\sqrt{n}} \right) - \phi \left(\frac{(4k-1)z}{\sqrt{n}} \right) \right] + \sum_{k=\frac{(-n+1)}{z}}^{\frac{(n-1)}{z}} \left[\phi \left(\frac{(4k+3)z}{\sqrt{n}} \right) - \phi \left(\frac{(4k+1)z}{\sqrt{n}} \right) \right] \quad (17)$$

donde ϕ es la función de distribución de probabilidad acumulativa normalizada, z es el estadístico del test. Si $Pvalue$ es menor a 0.01, se considera que la secuencia es no aleatoria. En el caso contrario, se concluye que la secuencia es aleatoria.

Para este test se recomienda que cada secuencia a ser testada consista en un mínimo de bits de tal forma que n , la cantidad de bits de la secuencia sea $n \geq 100$.

14. *The Random Excursions Test:*

Este test se enfoca en el número de ciclos que posee exactamente K veces una suma acumulativa en una corrida aleatoria. La suma acumulativa de una corrida aleatoria se deriva de las sumas parciales después que la secuencia binaria (0,1) se convierta apropiadamente a una secuencia (-1, +1). Un ciclo de corrida aleatoria consiste en los pasos de secuencia de longitud unitaria tomados aleatoriamente que comienzan desde y regresan al origen. El propósito de este test es determinar el número de veces de un estado particular dentro de un ciclo que deriva de uno que se esperaría en una secuencia aleatoria. Este test es realmente una serie de ocho tests (y conclusiones), un test y conclusión para cada estado: -4,-3,-2,-1 y +1, +2, +3, +4.

Se debe realizar el cálculo de la probabilidad, páginas 2-33 a 2-37 de [13], mediante la ecuación 18:

$$Pvalue = \Gamma\left(\frac{5}{2}, \frac{\chi^2(obs)}{2}\right) \quad (18)$$

para los 8 casos. Si Pvalue es menor a 0.01, se considera que la secuencia es no aleatoria. En el caso contrario, se concluye que la secuencia es aleatoria.

Para este test se recomienda que cada secuencia a ser testeada consista en un mínimo de bits de tal forma que n , la cantidad de bits de la secuencia sea $n \geq 1000000$.

15. *The Random Excursions Variant Test:*

Este test se enfoca en el número total de veces que se visita un estado en particular (por ejemplo, ocurrencia) en una suma acumulativa de una corrida aleatoria. El propósito de este test es la detección de desviaciones desde el número esperado de visitas de varios estados en una corrida aleatoria. El test en realidad es una serie de 18 tests (y conclusiones), un test y conclusión por cada estado: -9,-8, ..., -1 y +1, +2..., +9

Para cada $\zeta(x)$, número total de veces que el estado x ocurre en los J ciclos, se debe realizar el cálculo de probabilidad, páginas 2-38 a 2-40 de [13], mediante la ecuación 19:

$$Pvalue = erfc\left(\frac{|\zeta(x)-J|}{\sqrt{2J(4|x|-2)}}\right) \quad (19)$$

Se calcularán 18 valores de Pvalue. Si Pvalue es menor a 0.01, se considera que la secuencia es no aleatoria. Caso contrario, se concluye que la secuencia es aleatoria.

Para este test se recomienda que cada secuencia a ser testeada consista en un mínimo de bits de tal forma que n , la cantidad de bits de la secuencia sea $n \geq 1000000$.

2.3.1.1. **Desarrollo de códigos de los Tests del NIST en MATLAB**

Si bien el NIST proporciona códigos de los 15 tests en lenguaje de programación C, para nuestra tesis, era necesario poseer los códigos de los 15 tests en MATLAB. La necesidad de poder manipular el control del análisis de series temporales experimentales, para nuestro trabajo, implicaba el desafío de poder procesar dichas series en MATLAB. Sin embargo, en la actualidad, no existen implementaciones verificadas en MATLAB, algunas de las fuentes no certificadas poseían errores groseros. Es por lo que, para nuestra tesis, se tuvieron que implementar los 15 tests del NIST en MATLAB. Cada test desarrollado en MATLAB se corroboró con los resultados empíricos provistos por el NIST páginas del Apéndice B de [13], arrojando idénticos resultados y verificando de esta forma, que los algoritmos implementados en MATLAB eran confiables para

utilizarlos en nuestro trabajo. Los códigos en MATLAB desarrollados se adjuntan en el Apéndice A de nuestra tesis.

2.3.1.2 Consideraciones de los tests del NIST

Los tests del NIST forman parte de una herramienta poderosa para la evaluación de la aleatoriedad de una secuencia de datos. Sin embargo, algunos tests poseen la desventaja que, para realizar un cálculo confiable, requieren de millones de puntos. Y, esta limitación, para la implementación en tiempo real, en donde se adquieren bloques de datos de cierta cantidad de muestras, o en series temporales cortas, no resulta la mejor opción. Por otro lado, la aleatoriedad algorítmica resulta una buena alternativa para series temporales cortas. Sin embargo, al resultar la definición de la complejidad de Kolmogorov, como se ha visto con anterioridad, una opción muy intuitiva, resulta incomputable, es decir, no existe un algoritmo que la pueda calcular. Es por ello, que se requieren de aproximaciones, como los algoritmos de Lempel-Ziv.

2.3.2 Complejidad de Lempel-Ziv como aproximación de la complejidad de Kolmogorov

Se ha comentado con anterioridad, que la complejidad de Kolmogorov, resulta incomputable. Sin embargo, existen métodos que permiten el cálculo en forma aproximada, como lo es la complejidad de Lempel-Ziv.

Si se posee secuencias compuestas solo por ceros y unos, es decir, binarias, según Kolmogorov, la complejidad de una secuencia binaria se obtiene a partir del número de bits del programa más corto que puede generar dicha secuencia. Un algoritmo que pueda realizar dicha evaluación, como ya se ha mencionado, no es posible. Lempel y Ziv eligieron de todos los posibles programas una clase que permite dos tipos de operaciones: copiar e insertar, midiendo un número $c(n)$, que es una medida apropiada de la complejidad de Kolmogorov, sin necesidad de calcular la longitud del programa generador de la secuencia cuyo tamaño de bits es n . El cálculo de este número $c(n)$ procede de la siguiente forma:

Suponer una secuencia binaria como la expresión: $s_1 s_2 \dots s_n$, que fue reconstruida por un programa hasta antes de cierta posición r , es decir, hasta s_r . Esta secuencia hasta la recientemente insertada s_r , se denotará como $S = s_1 s_2 \dots s_r$. donde el punto indica que s_r se ha insertado recientemente. Para saber si el resto de la secuencia de S , es decir, $s_{r+1} \dots s_n$, se puede reconstruir con el simple hecho de copiar (o si uno tiene que insertar nuevos dígitos), Lempel y Ziv, consideran los siguientes pasos: toma el valor de $Q \equiv s_{r+1}$ y le consulta si, este término se encuentra en el vocabulario, es decir, si pertenece o si corresponde a un conjunto de

valores que se encuentra en la secuencia S, de tal forma que Q, puede obtenerse copiándose de un conjunto de caracteres de S. Equivale a la pregunta si Q se encuentra contenida en el vocabulario de $SQ\pi$, $v(SQ\pi)$, donde $SQ\pi$ denota el conjunto de caracteres que componen S y Q, y π significa que el último dígito debe eliminarse, es decir, la expresión $SQ\pi = S$. Esta situación puede extenderse a si Q posee dos o más elementos. Por ejemplo, $Q = s_{r+1}s_{r+2}$. Si s_{r+1} se encuentra contenida en el vocabulario de $SQ\pi$, es decir $v(Ss_{r+1})$, y luego, sucede lo mismo con el siguiente, y así sucesivamente hasta que Q se convierte en tan largo que ya no se puede obtener copiando de una palabra del vocabulario $v(SQ\pi)$, y esto implique que se deba insertar un dígito nuevo al vocabulario. El número c de pasos producidos para insertar nuevos dígitos (más uno si corresponde a una última copia que no poseerá más dígitos subsiguientes), se utiliza para medir la complejidad de una cadena de caracteres o secuencia, páginas 482-483 de [12]. A continuación, se ejemplificará lo desarrollado en el párrafo anterior:

Ejemplo 1: Suponer una secuencia $s_1s_2 \dots s_n$ como 1111.....1, supongamos una secuencia $S = s_1 = s_r = 1$, y nuestro $Q = s_2 = s_{r+1} = 1$, entonces nuestro vocabulario es $v(SQ\pi)$ donde si es igual a S. Como en el vocabulario $v(SQ\pi)$, se elimina el último dígito, es equivalente a $v(S)$, en este caso, a 1. Si seguimos analizando los valores de la secuencia y comparando con el conjunto, para el último valor, como ya no habrá un nuevo dígito a insertar, se le sumará a la complejidad un uno más, por lo cual, la complejidad c de Lempel-Ziv para una secuencia de unos es $c = 2$.

Ejemplo 2: Suponer una secuencia $s_1s_2 \dots s_n$ como 10 10 1010, y una secuencia $S = s_1 = s_r = 1$ y un $Q = s_2 = s_{r+1} = 0$, $SQ = 10$, sacando el último dígito, el vocabulario queda conformado con $v(SQ\pi) = 1$, pero Q es cero, por lo cual, no pertenece al vocabulario y se agrega un dígito más al vocabulario, es decir, ahora estará compuesto por 1 y 0. Si tomamos $Q = 10$, para los siguientes dígitos, hasta el último par, se tendrá que estos estarán dentro del vocabulario. Para el último par, como ya no habrá nuevo carácter a insertar, se le sumará 1, por lo cual, la complejidad c de Lempel-Ziv para una secuencia de unos es $c = 3$.

Ejemplo 3: Suponer una secuencia $s_1s_2 \dots s_n$ como 0010, y se elige $S = s_1 = s_r = 0$, y $Q = s_2 = s_{r+1} = 0$, $SQ = 00$, $v(SQ\pi) = 0$. Para el segundo dígito en cuestión, pertenece al vocabulario. Luego, $S = s_1 = s_r = 0$, y $Q = s_{r+1}s_{r+2} = 01$, $SQ = 001$, $v(SQ\pi) = 00$, Q no pertenece al vocabulario. Entonces, se agrega un dígito nuevo al vocabulario. Nuevamente, $S = 001$ y $Q = 0$, $SQ = 0010$, $SQ\pi = 001$, Q pertenece al vocabulario. Entonces, al ser Q el último, se suma 1 a la complejidad c. Por lo cual, $c = 3$.

Si bien cada secuencia n , tendrá su complejidad $c(n)$, ya que dependerá de la longitud de la secuencia a evaluar, a modo comparativo, se normaliza dicha complejidad con una cota $b(n)$. Esta cota, surge de un análisis [20], en donde se obtiene que el valor de $b(n)$, que es el valor que tiende la complejidad de la secuencia cuando n tiende al infinito. Este, está dado por $b(n) \equiv n/\log_2 n$, y la complejidad normalizada, se calcula como el cociente de la complejidad de la secuencia $c(n)$ respecto de la cota $b(n)$, es decir, la complejidad normalizada $C(n)$ se calcula como $C(n) = \frac{c(n)}{b(n)}$.

Cuando la complejidad normalizada tiende a 0, significa que la secuencia binaria corresponde a una serie temporal periódica. En cambio, cuando esta complejidad tiende a 1, se corresponde con series temporales aleatorias. Si bien las series temporales no lineales poseen una variación de $C(n)$ entre valores de 0 y 1, se ha demostrado que la complejidad $C(n)$ puede resultar mayor a 1, página 2975 de [20].

También existe una herramienta como lo es el espectro de la complejidad de Kolmogorov en donde dadas las secuencias de series temporales binarizadas respecto la mediana de las muestras x_i , con N muestras, y las secuencias c_i , con N secuencias, a las cuales se le calcula la complejidad de Lempel-Ziv N veces sobre las secuencias, se obtiene la complejidad de Kolmogorov aproximada de las N secuencias. Este espectro, permite explorar el rango de amplitudes en una serie temporal representando un proceso. El valor máximo de estos se denomina KCM (*Kolmogorov complexity spectrum highest value*), páginas 2976-2977 de [20].

2.3.2.1. Códigos de la complejidad de Lempel-Ziv en MATLAB

En la actualidad, preexisten implementaciones de algoritmos que pueden calcular la complejidad de Lempel-Ziv [20] basados en el diagrama de flujos del algoritmo que calculan la complejidad de una secuencia $c(n)$ de longitud n , figura 1 de la página 483 de [12].

En nuestra tesis, como ya se ha comentado, a partir de las series temporales experimentales, se obtienen secuencias binarizadas respecto de la mediana de la muestra: los valores por debajo de la mediana serán considerados 0, y los valores por arriba, serán considerados 1. Luego, se le calculará la complejidad de Kolmogorov en forma aproximada a partir del algoritmo de complejidad de Lempel-Ziv mencionado en el párrafo anterior.

CAPÍTULO 3:

ARREGLOS EXPERIMENTALES

3.1 Introducción

Se describirán los arreglos experimentales con los que se ha trabajado en esta tesis.

En primer lugar, se presentará el arreglo experimental correspondiente al estudio de señales cuánticas en el área de pares de fotones entrelazados para la implementación de experimentos de las desigualdades de Bell. En 1935 Einstein, Podolsky y Rosen presentaron un experimento ideado para mostrar que la descripción cuántica de un sistema físico es incompleta. El Teorema de Bell o desigualdades de Bell se aplica en mecánica cuántica para cuantificar matemáticamente las implicaciones planteadas teóricamente en la paradoja EPR (Einstein-Podolsky-Rosen) y permitir así su demostración experimental. En nuestro trabajo utilizaremos un arreglo experimental con el diseño e implementación de software necesario para la realización del experimento de las desigualdades de Bell con las estaciones a distancia (60 m aprox.). Se destaca que este arreglo experimental es único en su tipo en nuestro país. Se logró el diseño y construcción de la red wi fi que comunica a la computadora central con las estaciones remotas, la determinación automática de las posiciones de las láminas de media onda que determinan la polarización de los fotones, la operación a distancia del encendido y apagado del láser, en forma sincrónica con el comienzo de la adquisición de datos por el método de *time stamping* de cada uno de los TDCs (*Time to Digital Converters*), así como el análisis de los datos obtenidos.

En segundo lugar, se presentará el arreglo experimental correspondiente al estudio de señales clásicas en el esquema de automatización aplicado al láser sólido con modulación de pérdidas construido en el laboratorio, permitiendo analizar la señal del láser en tiempo real.

3.2. Experimento desigualdad de Bell

En este experimento, los pares de fotones entrelazados en polarización se generaron por fluorescencia paramétrica, SPDC (*Spontaneous Parametric Down-Conversion*), bombeando un par de cristales no lineales BBO (*beta-BaB2O4*) tipo I con un láser de diodo que emite en la longitud de onda de 405 nm. El láser puede operar en modo continuo o pulsado y tiene una longitud de coherencia mayor a 4 cm. El láser se modula por medio de una señal provista por un generador de señales, comandado por la computadora madre de la estación C. En la figura 9, se puede observar el esquema del arreglo experimental y en las figuras 10, 11 y 12, se muestran fotografías de dicho arreglo experimental montado en el laboratorio.

La detección de los fotones se realiza en las estaciones A y B. Cada rollo de fibra óptica de cada estación es de aprox. 30 m cada una, por lo cual, ambas estaciones, se encuentran separadas por aprox. 60 m de distancia física real desde la salida de la óptica de enfoque de las láminas de media onda hasta el polarizador de fibra. Los fotones atraviesan los polarizadores, láminas de media onda HWP (*Half Wave Plate*) para 810nm, y óptica de enfoque: filtros interferenciales (centrados en 810nm) y objetivos de microscopio. Las láminas HWP se montan en los rotadores motorizados (figuras 10.G y 11.F), PRM1Z8 (*Motorized Rotation Stage de Thorlabs*), que son plataformas motorizadas de precisión. La rotación se realiza a través de un servo motor de continua. El rango de rotación es de 360 grados. Su velocidad máxima de rotación es de 25 grados por segundo. Su resolución calculada es de 0.0005 grados. Su voltaje nominal es de 6 V. Las plataformas motorizadas, se encuentran comandadas por la computadora madre de la estación central C a través de comunicación USB (figura 9.2 y 9.3). Luego, la radiación se acopla a fibras ópticas multimodo conectadas a los detectores de fotones individuales SPCM-AQR-13-FC (*Single Photon Counters Modules*) de Perkin-Elmer.

Cada estación posee dos SPCM, cada uno de los cuales corresponde a una salida de los polarizadores de fibra para la compensación de la birrefringencia (figura 10.D.CB y 11.D.CB). Estos módulos SPCM, detectan fotones simples de una longitud de onda desde 400 nm a 1060 nm, utilizando un fotodiodo de avalancha de silicio. Poseen una precisión de 350 ps FWHM (*Full Width at Half Maximun*). Requieren de una fuente de +5 V. Al detectar un fotón, estos módulos emiten un pulso TTL (*Transistor-Transistor Logic*) de 2.5 V (mínimo) en una carga de 50 ohm, con una duración temporal de 35ns y con un tiempo muerto de 50 ns entre pulso y pulso. Estos pulsos TTL se conectan directamente a dos entradas del conversor TDC ID900 de ID Quantique que se muestra en la figura 10.F y 11.E.5. Posee 4 canales de entrada y 4 canales de salida, resolución de 10 ps y versión de firmware 0.11. De sus cuatro canales de entrada disponibles, en el experimento en el área de fotones entrelazados, se utilizan tres canales: dos canales para el registro de los pulsos provenientes de los SPCM y uno para la salida del fotodetector atenuado por un filtro (figura 12.E.5), que detecta una pérdida de la señal original.

El control de los datos de adquisición de los TDC de cada estación A y B se realizan mediante las computadoras hijo A e hijo B. Las tres computadoras de las tres estaciones A, B y C se encuentran comunicadas mediante wi fi por comunicación TCP/IP a través de una red local del laboratorio de láseres sólidos del DEILAP.

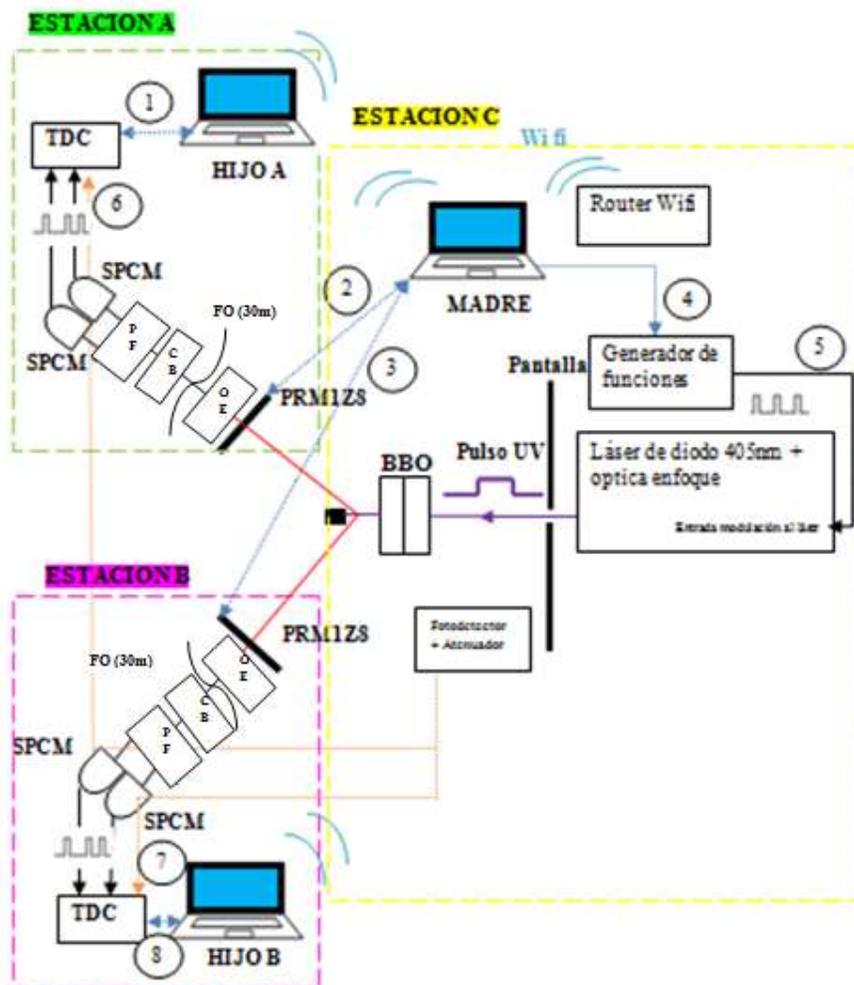


Figura 9. Esquema del arreglo experimental para experimento de desigualdades de Bell con estaciones A y B separadas por 60m. **1) y 8):** Comunicación Ethernet de las computadoras hijo A y B con sus TDC. **2) y 3):** Comunicación USB de la computadora madre con los rotadores motorizados PRM1Z8 en donde se montan las láminas de media onda. **4)** Comunicación USB de computadora madre con generador de señales. **5)** Señal del generador para modulador del láser. **6) y 7):** Señal de salida del fotodetector atenuado al canal 3 de los TDCs (Time to Digital Converters), en los canales 1 y 2 recibe las señales de los SPCM (módulo para conteo de fotones). **OE:** Óptica de Enfoque. **CB:** Compensador de birrefringencia. **FO:** Fibra óptica de 60m (total). **PF:** Polarizador de Fibra. **BBO:** Cristal generador de pares de fotones entrelazados.

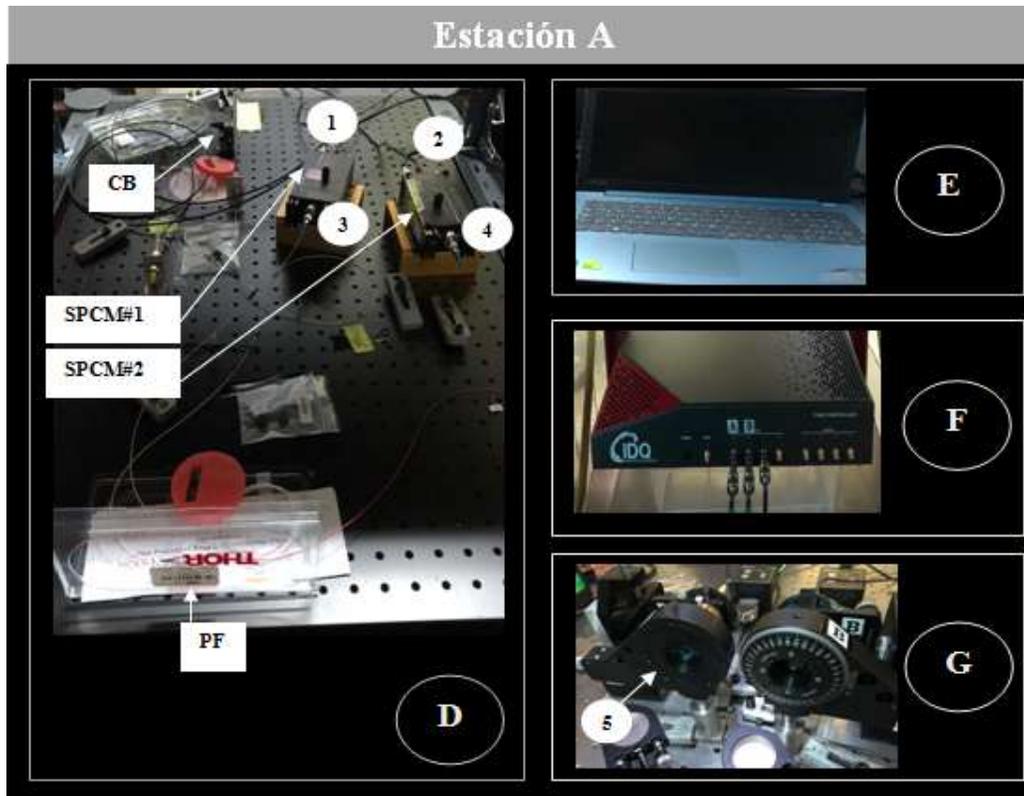


Figura 10. Fotografía del arreglo experimental de la Estación A: **D.1)** y **D.2)**; Señal de los SPCM hacia el TDC de estación A, ch1 y ch2. **D.3)** y **D.4)**; Entradas a los SPCM provenientes del polarizador a fibra. **D.PF**: Polarizador de fibra. **D.CB**: Compensador de birrefringencia. **E)** PC hijo A. **F)** TDC estación A. **G.5)** Rotador motorizado PRM1Z8 estación A.

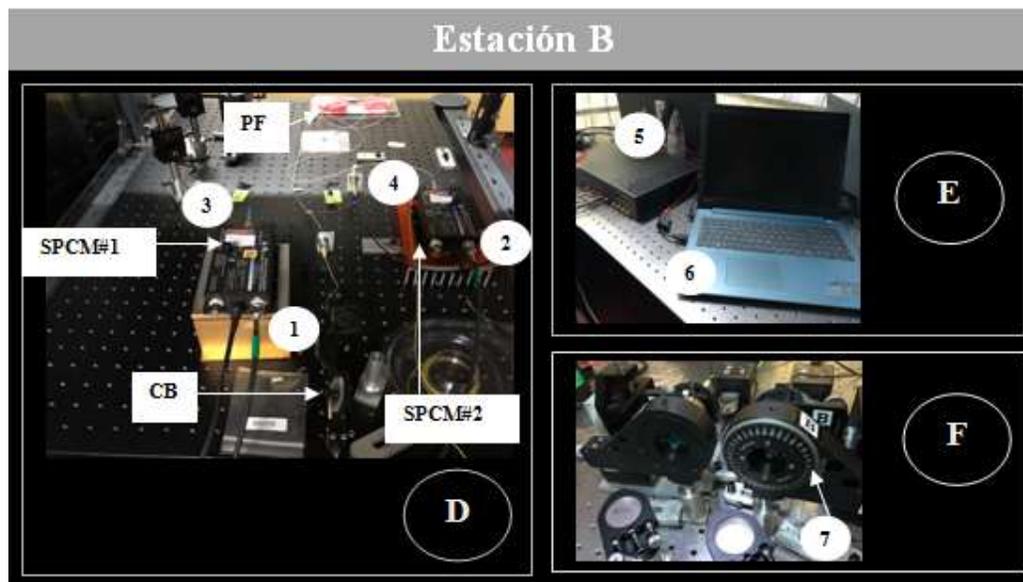


Figura 11. Fotografía del arreglo experimental de la Estación B: **D.1)** y **D.2):** Señal de los SPCM hacia el TDC de estación B, ch1 y ch2. **D.3)** y **D.4):** Entradas a los SPCM provenientes del polarizador a fibra. **D.CB:** Compensador de birrefringencia. **D.PF:** Polarizador de fibra. **E.6)** PC hijo B. **E.5)** TDC estación B **F.7)** Rotador motorizado PRM1Z8 estación B.

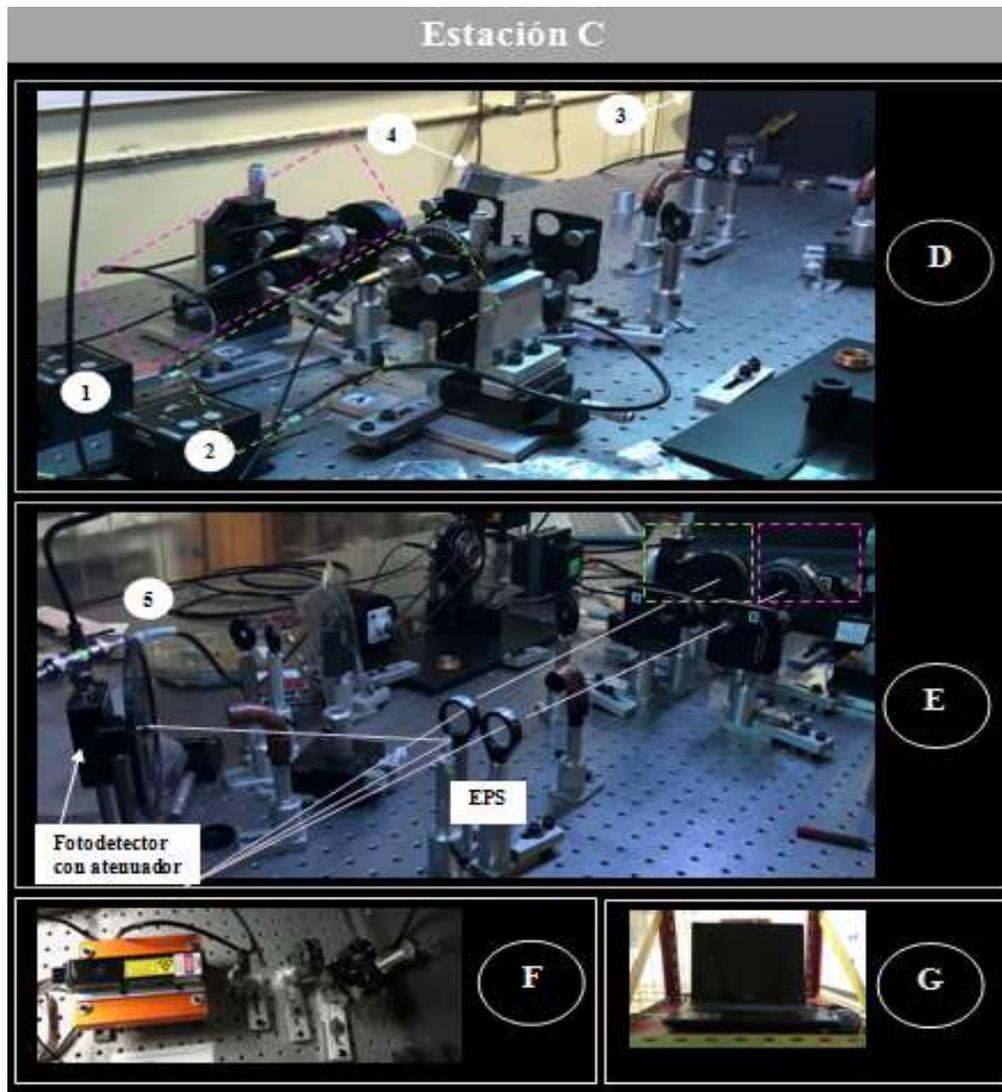


Figura 12. Fotografía del arreglo experimental de la estación C. En recuadros de líneas verde y magenta, se encuentran indicadas partes de la estación A y B respectivamente, correspondiente a los rotadores PRM1Z8. EPS: Espejo de primera superficie. E.5) Salida de fotodetector al ch3 de los TDC de las estaciones A y B. D.1) y D.2): Drivers y conexión USB de rotadores a PC madre. D.3) Pantalla. F) Diodo láser con su óptica de enfoque, permite modulación con generador de funciones en D.4). G) PC madre.

Se debe aclarar, que la mayoría de los experimentos de EPR trabajan con una ventana de coincidencias fija durante toda la medición por lo que no permiten evaluar las teorías que aprovechan el llamado *coincidence-loophole* (página 26 de [7]) y que se basan en la definición de una detección de una coincidencia que depende del tamaño de la ventana de coincidencias. Este experimento permite el estudio de estos tipos de teorías utilizando fotones entrelazados en polarización generados por fluorescencia paramétrica. Las coincidencias se determinan una vez

que termina la medición, con un procesamiento a posteriori variando la ventana de coincidencias a voluntad. Es por ello, que se requiere el registro del tiempo de detección de cada fotón (*time stamping*).

3.3 Experimento Láser sólido con modulación de pérdidas

En este experimento (figura 13), se empleó un láser con un cristal de Vanadato de Itrio dopado con iones de Neodimio (Nd: YVO₄) como medio activo. El material utilizado presenta ventajas por producir una gran ganancia en la amplificación óptica y por su amplio espectro de fluorescencia con una fuerte banda de absorción en 808 nm. El medio activo, es una lámina de 3mmx3mmx1mm con 1% de átomos de Neodimio con recubrimiento antirreflejo para 808 nm y alta reflectividad para 1064 nm en una cara y antirreflejo para 1064 nm en la otra, es bombeado por un diodo de 2 W de potencia máxima en 808 nm. Se utilizó una configuración de la cavidad en V. Sobre posicionadores angulares, se montó un espejo esférico de fondo con un revestimiento HR y otro plano de reflectividad 83%. El fotodiodo (figura 13.2), se ubicó detrás del espejo de salida.

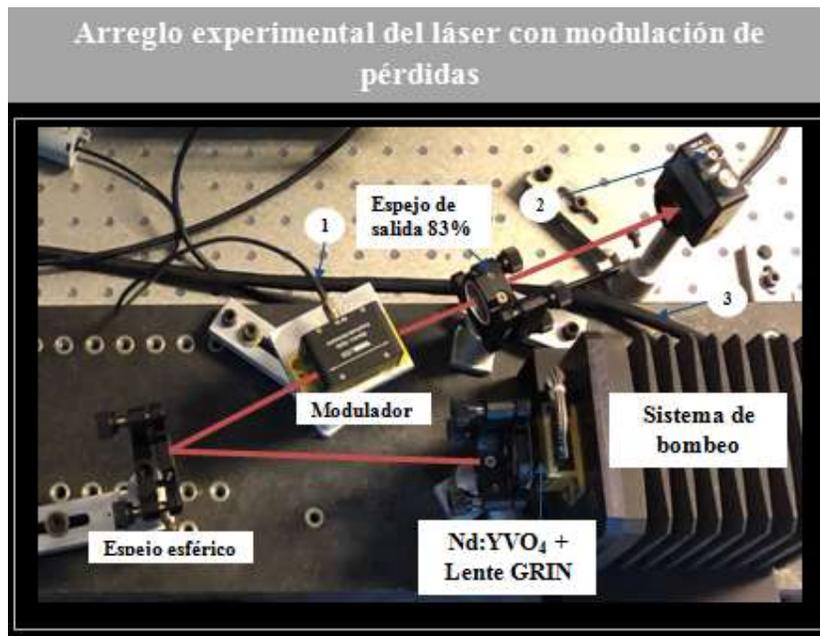


Figura 13. Fotografía del arreglo experimental del láser con modulación de pérdidas. 1) Conexión al amplificador-generador de funciones. 2) Conexión al PicoScope. 3) Conexión a la fuente del diodo.

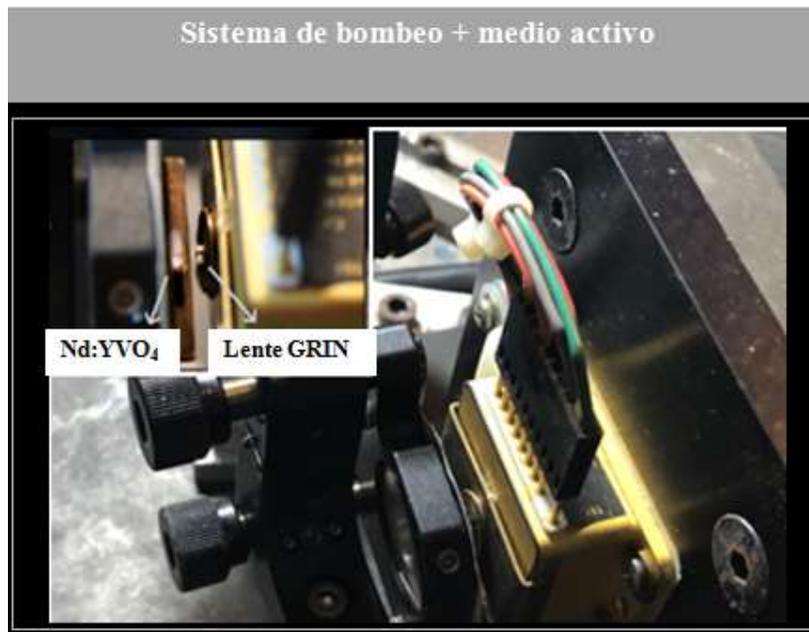


Figura 14. Fotografía de parte del sistema de bombeo + medio activo. En el recuadro se muestra una vista superior en detalle del medio activo y la lente GRIN.

El enfoque de la radiación de bombeo sobre el medio activo se logró con una lente de gradiente de índice (GRIN LENS) montada sobre una pieza metálica ajustada al armazón del diodo, de forma cilíndrica de radio 1.8 mm y largo 3.27 mm (figura 14). Sobre una de las ramas de la cavidad se colocó un modulador electro-óptico, que permite la variación de las pérdidas de la cavidad. Este, se encuentra conectado (figura 13.1) al amplificador (figura 15.1) y al generador de señales (figura 15.3). Se utilizó un diodo láser JDS Uniphase modelo SDL-2400 junto con un disipador de calor; una fuente de alimentación Spectra Diode Labs modelo SDL-805 (figura 15.2) el cual provee al diodo (figura 13.3) corriente eléctrica estable a través de un circuito electrónico realimentado.

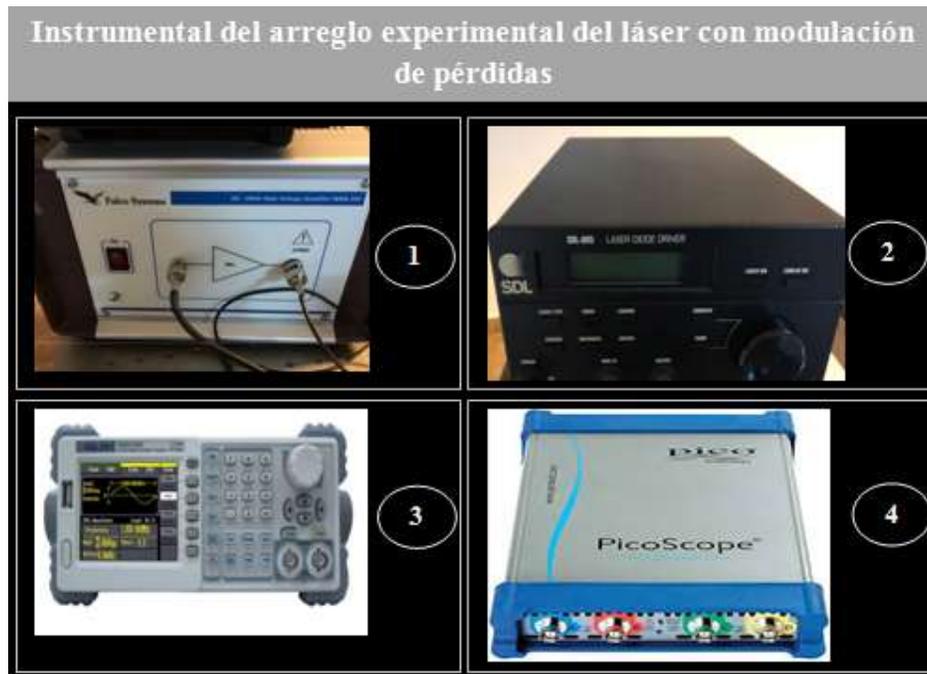


Figura 15. 1) Fotografía del amplificador Falcon Systems 50x. 2) Fotografía de la fuente del diodo Spectra Diode modelo SDL 805. 3) Imagen del generador de funciones SDG1025. 4) Imagen PicoScope modelo 6403B.

La emisión de salida del láser, por medio de un sensor de intensidad de luz como lo es el fotodetector de nuestro experimento, transforma una señal eléctrica proporcional a la intensidad de luz detectada. Para que estas señales analógicas puedan analizarse por computadora, es necesaria su conversión analógica a digital. Por supuesto, que una vez que se logran las señales digitales, se tendrán grandes volúmenes de datos, del orden de magnitud superior al millón de muestras por segundo. Estas, deberán procesarse para su análisis. En la figura 16, se muestra un esquema del arreglo experimental junto a dispositivos que permiten la adquisición de la señal de salida del láser para su procesamiento mediante una computadora.

Los datos se adquieren mediante un osciloscopio para PC (PicoScope 6403B) que se puede visualizar en la figura 15.4, posee 4 canales, resolución de conversión de tensión de 8 bits, un ancho de banda de 350 MHz, una frecuencia de muestreo de 5 GS/s (1 GS, 1 Gigasamples, aproximadamente 1 mil millones de muestras), una base de tiempo de 10 ns/div a 200 s/div, una memoria búfer de 512 MS (1 MS, 1 Megasamples, aproximadamente 1 millón de muestras), un

generador arbitrario de señal, una conexión USB 2.0 (soporta transferencias de datos arriba de 480 megabits por segundo) [22].

En una de las entradas del osciloscopio, a través de un conector BNC, se conecta la salida del fotodetector (figura 13.2). El fotodetector, mide la intensidad de radiación láser. Se utilizó uno de marca Thorlabs, el DET 36A. Este, es un detector de silicio polarizado con un rango de medición en las longitudes de onda entre 300 y 1100 nm y con un tiempo de respuesta (*rise time*) de 1 ns. El dispositivo, incluye además una batería de polarización interna de 12V.

El modulador electro-óptico, se encuentra modulado (figura 13.1) con una señal provista a través de un generador de funciones. El generador de señales utilizado es de la marca Siglent, modelo SDG1025 (figura 15.3). Tiene una tasa de muestreo de 125 MS/s, 14 bits de resolución. La máxima frecuencia de salida es de 25 MHz, 1 μ Hz de resolución. Tiene 2 canales de salida. Posee 5 tipos de señales de salida, incluye además señales arbitrarias.

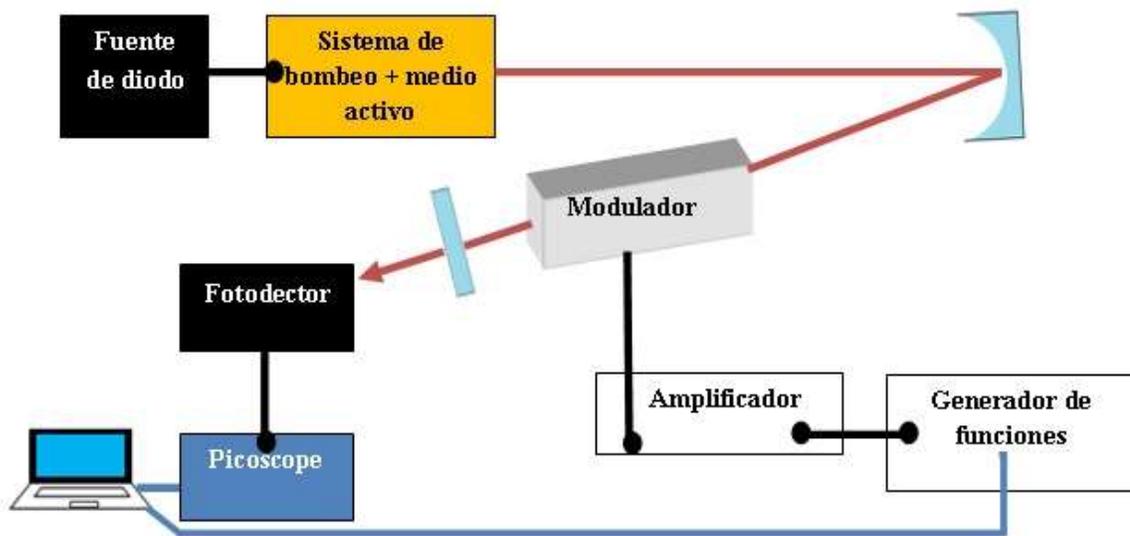


Figura 16. Esquema de la configuración del láser con cavidad en V y los dispositivos involucrados para la automatización y control de la señal de salida del láser.

A partir de estos dos arreglos experimentales que se han mencionado, que poseen al láser de estado sólido como elemento generador de las señales experimentales, se revelarán las propiedades físicas de dichos sistemas mediante su procesamiento basado en los fundamentos teóricos de las series temporales experimentales obtenidas respecto de estas señales. Para el caso de las señales

experimentales cuánticas, determinar la aleatoriedad de series temporales tiene implicancias cruciales en áreas como los fundamentos de la mecánica cuántica, así como las aplicaciones prácticas como la generación de números al azar. Para el caso de las señales experimentales clásicas, determinar el tipo de señal, además, abre las puertas a la implementación de sistemas de control y estabilización de las señales caóticas, conduciendo a láseres más compactos y eficientes.

CAPÍTULO 4:

AUTOMATIZACIÓN, CONTROL Y PROCESAMIENTO

4.1 Introducción

La realización de mediciones y control de dispositivos en forma automática durante un experimento es una herramienta de gran utilidad cuando se requiere sincronizar instrumental de laboratorio.

En ensayos del área de fotones entrelazados, debido a la sensibilidad a la luz de los SPCM, se requiere trabajar a oscuras durante la medición. Esto implica realizar al mismo tiempo varias acciones en un lugar carente de iluminación, tales como: manipular los ángulos de rotación de las láminas de polarización; configurar y adquirir mediante dos TDC al simultáneo como así también salvaguardar sus datos y configurar el generador de señales para la señal de modulación del láser. Indudablemente, es muy fácil equivocarse al manipular los instrumentos, más aún, si deben realizarse varias mediciones para diferentes ángulos de rotación de las láminas de polarización. Esto hace que se entorpezca la medición, debiendo desechar los datos obtenidos, teniendo que nuevamente repetir la experiencia. Tener en cuenta que hay experiencias que son de larga duración (3 hs aprox.). Todo lo anterior, hace que sea indispensable la automatización y control del sistema físico para la obtención de datos que luego serán procesados.

Al igual que lo comentado en el párrafo anterior, ocurre algo similar para el caso de los ensayos en el área clásica con el láser de modulación con pérdidas. Si bien este tipo de ensayo no se realiza a oscuras, la automatización y control del sistema, son de gran utilidad para el procesamiento de la señal en tiempo real. La necesidad de adquirir la señal de salida del láser y procesarla en tiempo real, no solo requiere del control del instrumental involucrado, sino que pretende realizar una automatización de la adquisición con el osciloscopio de PC en conjunto con el procesamiento de la señal. Permitiendo así, la detección del tipo de señal como así también logrando realimentar el sistema con la modificación de una variable del sistema físico, como, por ejemplo, la frecuencia de repetición del láser junto con el control del generador de funciones que se utiliza como señal del modulador.

Para el diseño y desarrollo de Software necesario para la automatización, control y procesamiento de los sistemas físicos que se han mencionado, se utilizaron las plataformas de programación LabVIEW versión 2017 (desarrollado por National Instruments) y MATLAB R2014a.

4.2. Automatización y control para señales cuánticas

La automatización y control de señales es esencial cuando se quiere sincronizar la adquisición de señales que se adquieren en mediciones de laboratorio de forma simultánea. En particular, la mayoría de los experimentos de óptica cuántica se basan en la detección de fotones individuales. Como se ha mencionado con anterioridad, esto necesariamente impone restricciones en las condiciones de trabajo: muchas de las mediciones deben realizarse en completa oscuridad. Durante los experimentos es necesario controlar una serie de instrumentos de forma individual y monitorear parámetros.

4.2.1 Comunicación wi fi entre las tres estaciones

Para la adquisición de señales cuánticas y control de instrumental aplicados a experimentos con fotones entrelazados en polarización, se desarrollaron programas en LabVIEW. En la figura 17, puede observarse un esquema de comunicación entre dos estaciones A (Alice) y B (Bob), con aproximadamente 60 m de separación entre ambas, y una estación C (central).

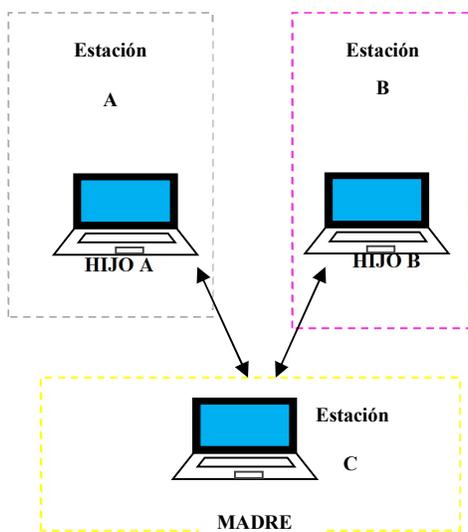


Figura 17. Esquema de comunicación entre las estaciones A, B y C.

Cada estación, A, B y C, posee una computadora portátil, las cuales se las nombró por analogía a como se llaman los procesos en programación que poseen orden jerárquico: proceso padre o madre y proceso hijo. Los hijos A y B, controlan la adquisición de señales de la estación A y B, y la madre, coordina las distintas actividades a desarrollar desde la Estación C. Desde ahora en adelante, nos referiremos a las tres computadoras como madre e hijo A o B respectivamente.

Al estar las estaciones separadas por 60 m aprox., para la sincronización en la adquisición de datos, fue necesario implementar una conexión inalámbrica (wi fi) entre las tres computadoras. Para ello, se seleccionó un router con ciertos requerimientos como el de ser lo suficientemente potente como para atravesar paredes u obstáculos en el laboratorio, para experimentos a posteriori en donde, las estaciones estuvieran mayormente distanciadas y se tuviera que alojar las estaciones A y B, fuera del laboratorio en otro sector físico, debiendo ser uno que tuviera alta potencia de señal como para atravesar obstáculos tales como paredes, armarios de laboratorio, estantes, mesas ópticas, etc. Para ello, se escogió un router rompe muros marca Tenda modelo AC6 como el de la figura 18, con las características que se muestran en la tabla 2.



Figura 18. Imagen de router wi fi alta potencia

Especificaciones técnicas			
1	Tipo de router: Router	10	Cantidad de antenas: 4
2	Tipo de conexión: inalámbrico	11	Nomenclatura de velocidad inalámbrica: AC1200
3	Velocidad inalámbrica: 1200 MB/s	12	Cantidad de puertos LAN: 3
4	Frecuencias: 5 GHz	13	Cantidad de puertos WAN: 1
5	Tipo de frecuencia: banda doble	14	Conectividad: wi fi;
6	Interfaces: 3 puertos LAN	15	1 puerto WAN
7	Estándares inalámbricos: IEEE 802.11ac/a/n	16	IEEE 802.11b/g/n
8	Altura: 49 mm	17	Profundidad: 141.5 mm
9	Ancho: 220 mm		

Tabla 2. Especificaciones técnicas del router wi fi de alta potencia

Con el router wi fi, se crea una red inalámbrica de trabajo, a la cual se la nombra como "LS", y solo se puede utilizar únicamente para las computadoras madre e hijos. A estas computadoras, se les designó una dirección IP (*Internet Protocol*) de trabajo (tabla 3):

Direcciones IP	
1	IP de la madre: 192.168.0.105
2	IP del hijo A: 192.168.0.197
3	IP del hijo B: 192.168.0.103

Tabla 3. Direcciones IP para la comunicación TCP/IP

Para generar una comunicación entre estas direcciones, se implementó una comunicación TCP/IP (*Transmission Control Protocol/Internet Protocol*) entre la madre y ambos hijos. En forma muy general, los hijos establecen la conexión con la madre y se quedan en estado ocioso a la espera para recibir un carácter de inicio de actividad por parte de su madre: un carácter "A" para el hijo A y un carácter "B" para el hijo B. Cuando la madre ha finalizado sus tareas en su estación, envía estos caracteres de inicio de actividad a sus hijos. Los hijos reciben dichos caracteres y comienzan a realizar su actividad. Al finalizar la actividad, envían un aviso a su madre: el hijo A envía el carácter "R" a su madre y el hijo B le envía el carácter "S". Se repite el ciclo hasta completar la totalidad de actividades.

Los primeros que inician la ejecución de los programas son los hijos y se quedan esperando a que inicie la madre. Luego de su inicio, se genera la secuencia comentada en el párrafo anterior. Tanto en la figura 19 como en la figura 20, se muestra el diagrama de flujo de la comunicación de los hijos y la madre.

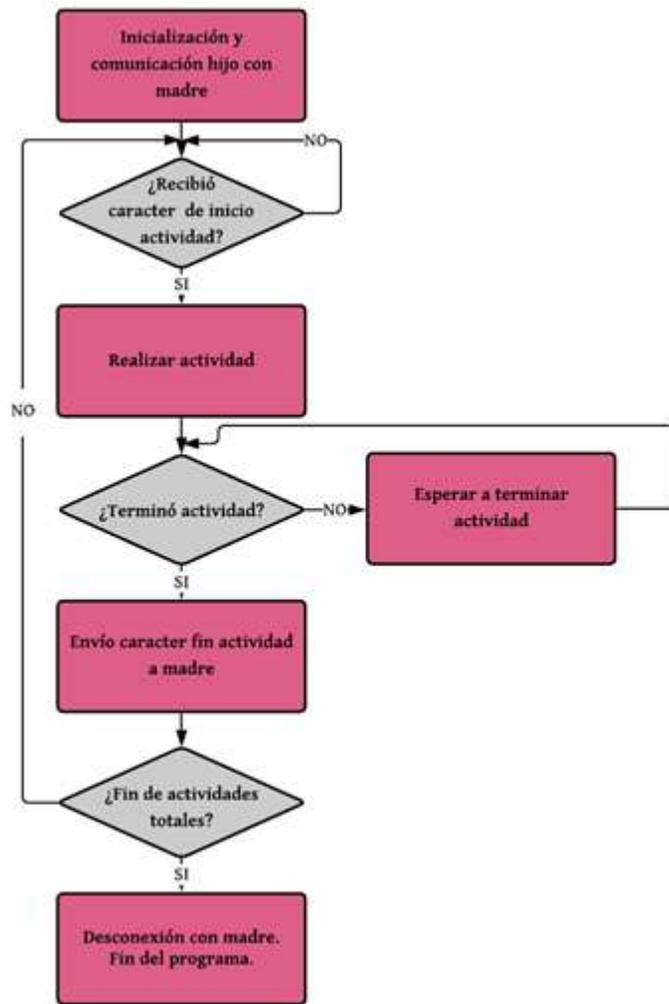


Figura 19. Diagrama de Flujo simplificado de comunicación tanto del hijo A como del hijo B

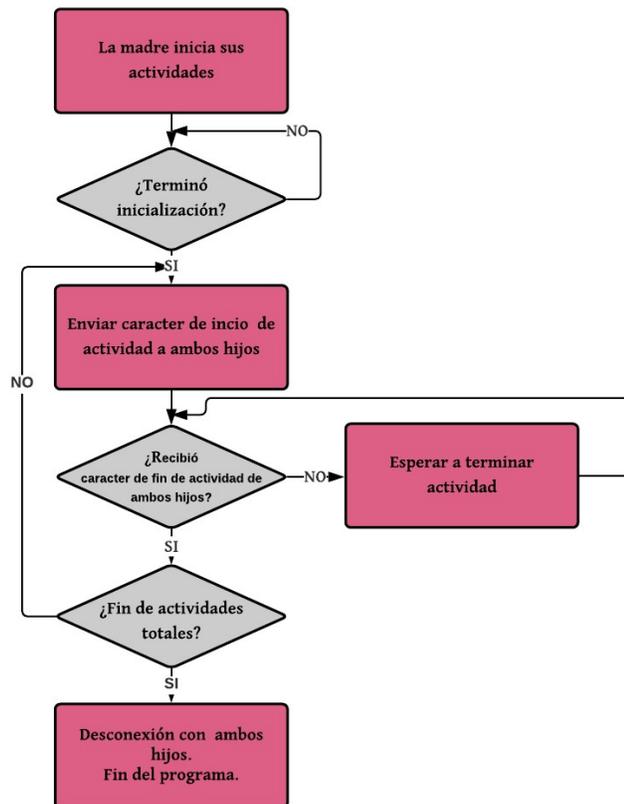


Figura 20. Diagrama de Flujo simplificado de la comunicación madre

4.2.2 Actividades realizadas por los hijos en cada estación A y B

En esta sección, se describirán en detalle las actividades que realizan ambos hijos. Como se mencionó con anterioridad, los hijos son los primeros que inician. En la figura 30, se muestra un diagrama de flujo más completa que en la sección anterior, referida a la comunicación genérica de ambos hijos (4.2.1).

A continuación, se enumerarán los pasos que realizan cada hijo al simultáneo:

1) Se realiza la lectura de los valores de las posiciones angulares de un archivo de texto. El archivo de texto se confecciona por el usuario con los valores angulares a los que desee rotar a la plataforma motorizada de cada estación. Estos ángulos, se deben escribir uno debajo del otro y en sistema sexagesimal. En la figura 21, se ejemplifica el listado de posiciones angulares de una estación en un archivo de texto para dos ángulos, uno de 45 grados y otro de 90 grados.

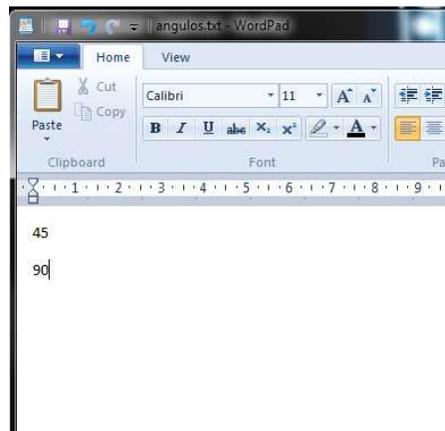


Figura 21. Captura archivo de texto con ángulos para los cuales se rotará al polarizador de media onda.

2) Para realizar la transferencia de datos desde el ID900 a la computadora de cada estación en particular, el fabricante provee un ejecutable "DataLinkTargetService.exe" (figura 22), que resuelve la sincronización de la transferencia de bloques de datos. Esta transferencia, se realiza por diferentes puertos, por ejemplo, las entradas 1 al 4 se corresponden con los puertos 5556 al 5559 de la dirección "tcp://localhost:6060" (en la figura 23, se muestra cómo se activa la comunicación a través de esta dirección). En la figura 24, se puede visualizar una porción del diagrama en bloques de la configuración del proceso de comunicación, en donde se indica cómo se activa y crea el archivo que contendrá los datos que provendrán de los puertos 5556 y 5557 que corresponderán al canal 1 y 2 respectivamente mediante el envío de la instrucción al localhost:6060. Para que el programa pueda funcionar correctamente, LabVIEW debe tener instalada una librería de comunicaciones de alto rendimiento orientada a mensajes "ZeroMQ Socket Library 3.3.1.102" de Martijn Jasperse, que puede descargarse e instalarse desde el VI package manager de LabVIEW.

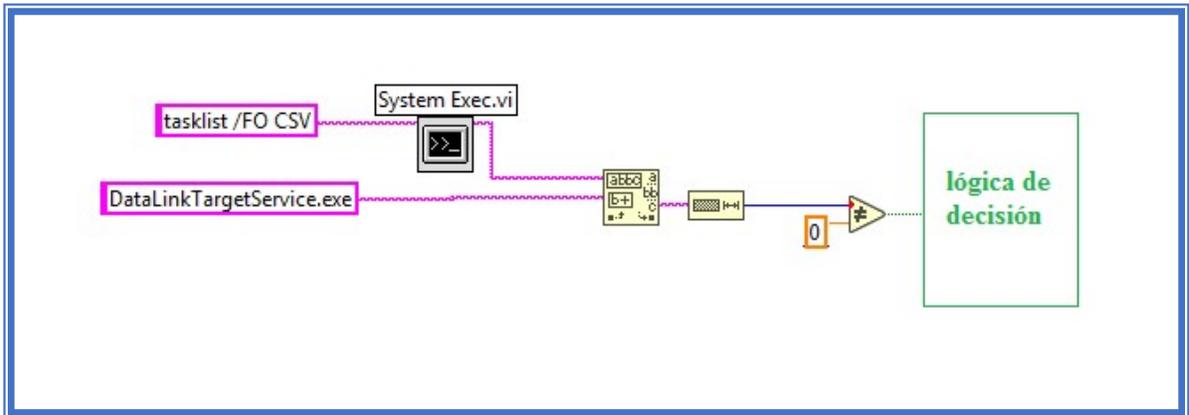


Figura 22. Parte del diagrama en bloques en LabVIEW para la ejecución del ejecutable para la transferencia de datos provisto por el fabricante.

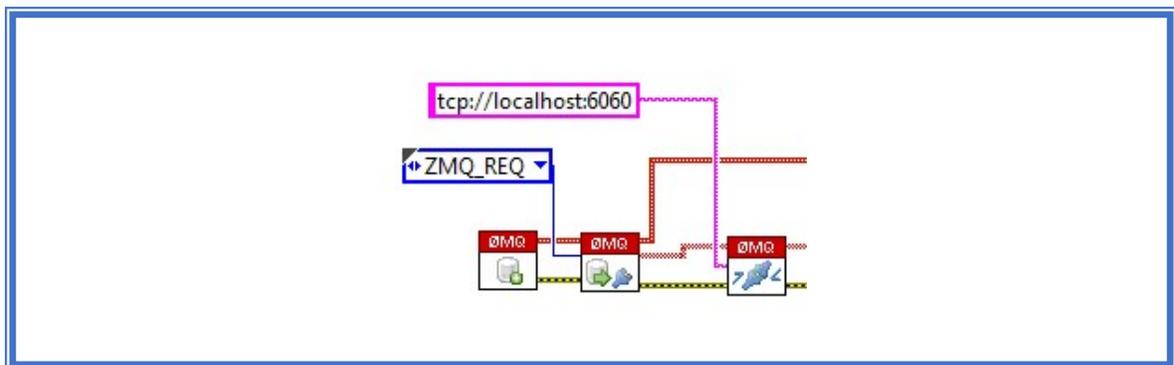


Figura 23. Parte del diagrama en bloques en LabVIEW de la comunicación para la transferencia de datos desde el ID900 con la computadora portátil por medio de la dirección "tcp://localhost:6060".

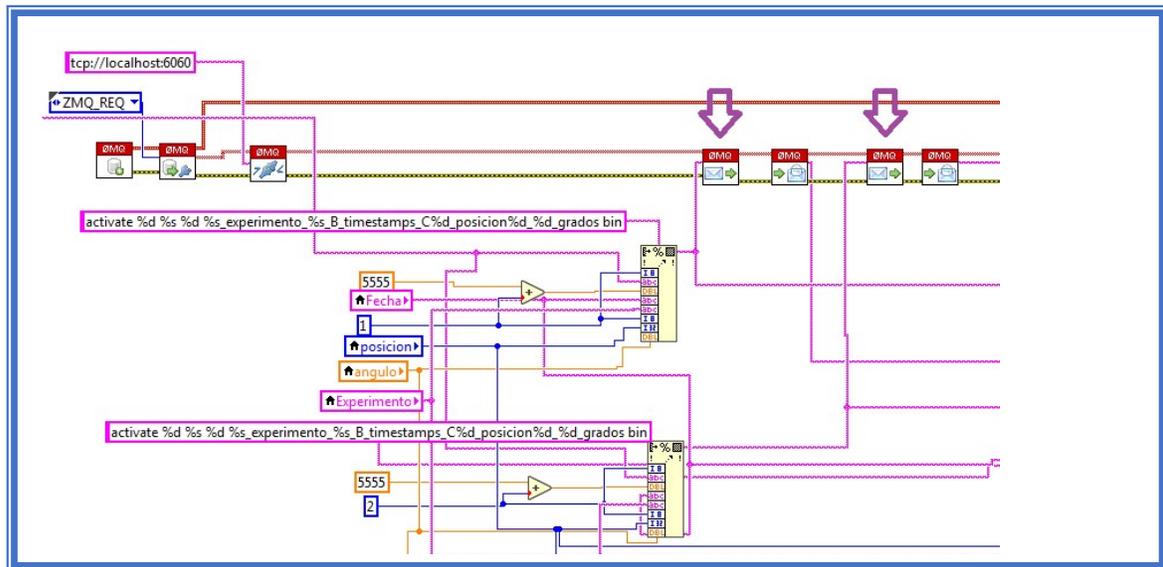


Figura 24. Parte del diagrama en bloques en LabVIEW para la configuración de puerto del canal 1 y 2. Las flechas en violeta indican el bloque que permite realizar el envío de dicho mensaje al localhost:6060.

3) Para cada posición angular, se generan archivos en formato BIN (uno por cada canal de entrada del ID900). En la sección 4.2.1.9, se explicará más en detalle respecto a la salvaguarda de datos. En esta etapa, mediante la comunicación con la IP del ID900 y por medio del puerto 5555 (ver figura 25), se habilitan los distintos canales de entrada con el comando `INPU#: ENAB ON` y se configuran los diferentes parámetros de adquisición: umbrales, con el comando de configuración `INPU#:THRE D mV`, el *delay* (retardo), con el comando de configuración `INPU#:DELAY D`, etc. Estos comandos utilizan sintaxis de la norma SPCI (*Standard comands for Programmable Instruments*) que provee el fabricante en su manual de usuario. El # se refiere al canal y debe reemplazarse por el respectivo número de canal. En el caso del canal 1, el # es igual a 1. La letra D, es una letra arbitraria que debe reemplazarse por un valor elegido de umbral, retardo, etc.

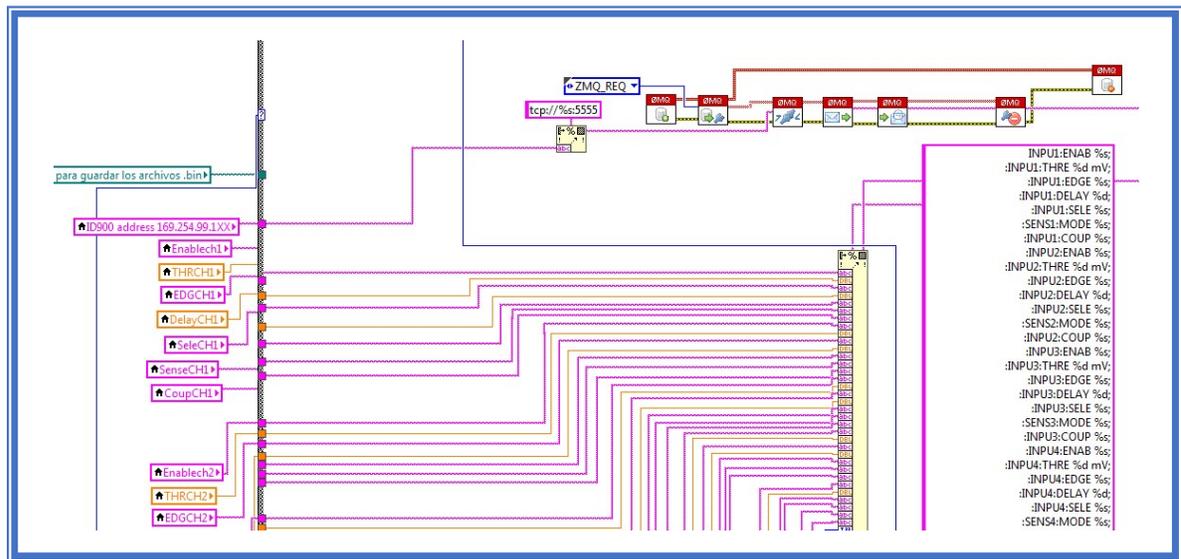


Figura 25. Parte del diagrama en bloques en LabVIEW de la configuración del ID900 mediante tcp://dirección IP del dispositivo:5555.

4) Se espera a recibir el carácter que les hará iniciar el *time stamping* por el puerto 8006 en el caso del hijo B (ver figura 26 y 27) y por el puerto 8005, en el caso del hijo A. Cuando lo recibe, termina la escucha del puerto y se desconecta.

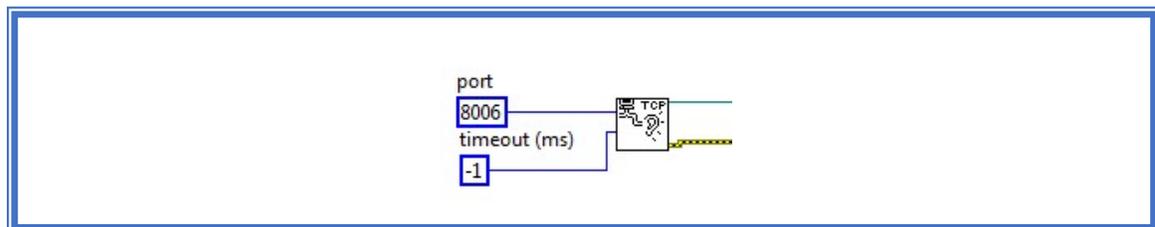


Figura 26. Esquema de bloques en LabVIEW para la conexión, programa escucha el puerto 8006 por donde la madre envía el carácter.

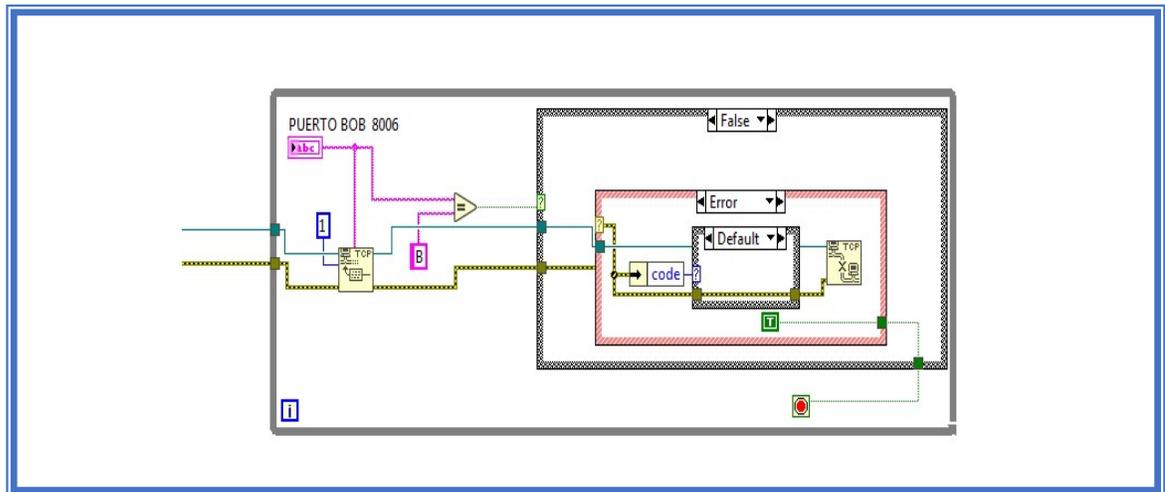


Figura 27. Esquema de bloques en LabVIEW para la espera de carácter B y desconexión del puerto 8006.

5) Se activa la adquisición de *time stamping* mediante el comando TSST#:DATA:RAW:SEND ON para los distintos canales de entrada seguido del comando TSGE8:ENAB ON. Notar que para que este comando responda se debe configurar previamente el generador interno del dispositivo TSGE8:ONES:PWID D000000000, en donde D es el valor en milisegundos y # el se debe reemplazar por el respectivo número de canal.

6) Se espera hasta que transcurra el tiempo de adquisición de *time stamping* (TSST#:DATA:RAW: SEND OFF) para cada canal.

7) Se conecta con la dirección IP de la madre (figura 28) y se envía el carácter de fin de *time stamping* por el puerto 8006 en el caso del hijo B (figura 29) y por el puerto 8005, en el caso del hijo A. Finalmente, se termina dicha conexión.

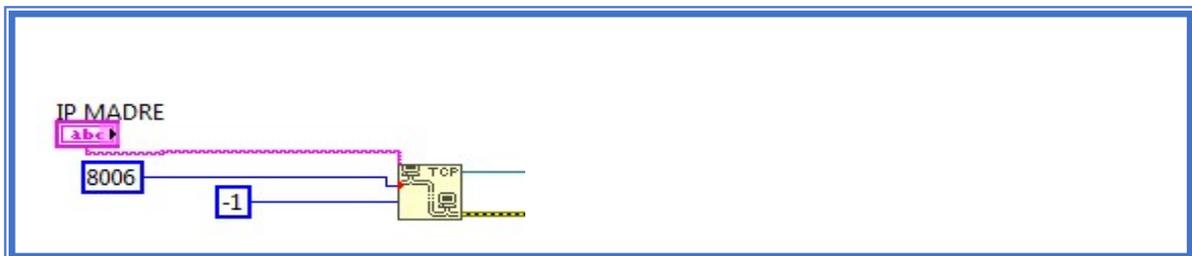


Figura 28. Parte de esquema de bloques en LabVIEW para la conexión con la IP MADRE y envío de caracteres por puerto 8006.

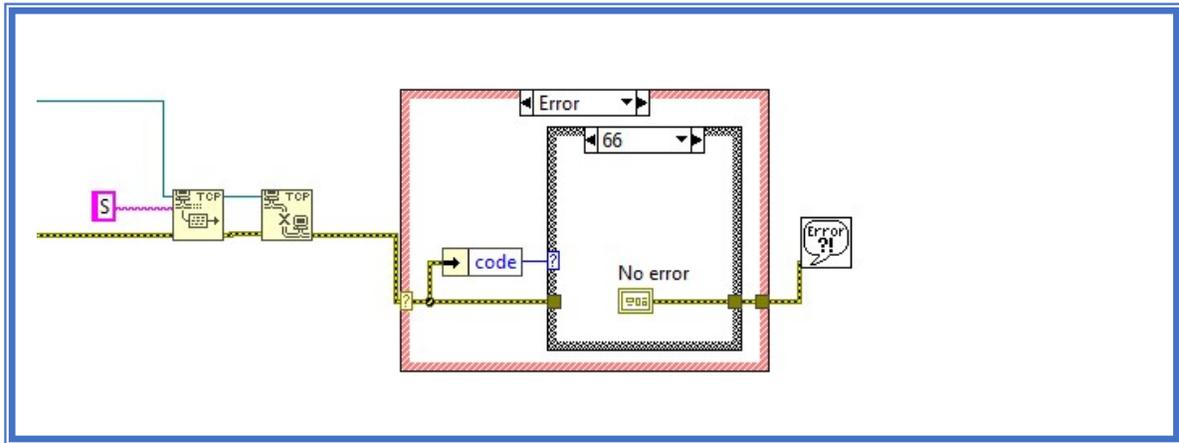


Figura 29. Parte de esquema de bloques en LabVIEW para el envío de carácter por puerto 8006 y desconexión con la madre.

8) Se cierran los archivos creados en el paso 3.

Los pasos 2 a 6 se repiten tantas veces como la cantidad de ángulos que haya que rotar. La madre se encarga de enviar la directiva para rotar las plataformas motorizadas. Una vez que lo hace, envía al simultáneo la señal de inicio de actividad a ambos hijos. Respecto a cada iteración, al iniciar, los hijos reciben de la madre la señal de inicio de actividad y al finalizar, los hijos envían un comando de fin de actividad hacia la madre, notificando que ya han culminado la iteración.

Cuando se han terminado de adquirir para todas las posiciones angulares, los hijos desactivan la comunicación del ID900 por medio de la dirección "tcp://localhost:6060". Además, se produce la desconexión TCP/IP con la madre.

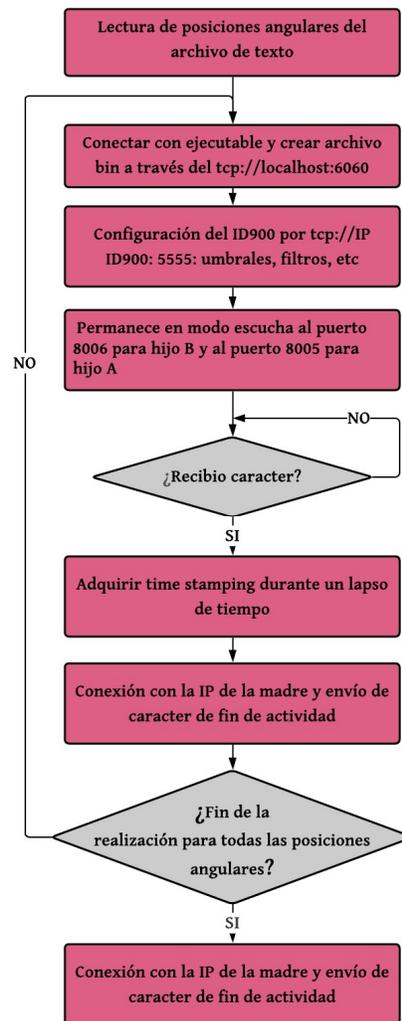


Figura 30. Diagrama de Flujo extendido de la comunicación genérica de los hijos.

4.2.3 Actividades realizadas por la madre en la estación C:

La madre inicia sus actividades después de que lo hacen sus hijos, es decir, primero se ejecuta el programa en LabVIEW de los hijos y luego el de la madre. En la estación C, se encuentran las láminas HWP que se montan en los rotadores motorizados PRM1Z8. La madre, es la encargada de coordinar las posiciones angulares.

A continuación, se explicarán los pasos que realiza la madre en cuanto a la comunicación. En la figura 34, se puede observar un diagrama de flujo con mayor detalle que el mencionado en la sección 4.2.1:

1) Enciende el láser en forma pulsada habilitando la señal que provee el generador de señales para modular el láser (figura 31).

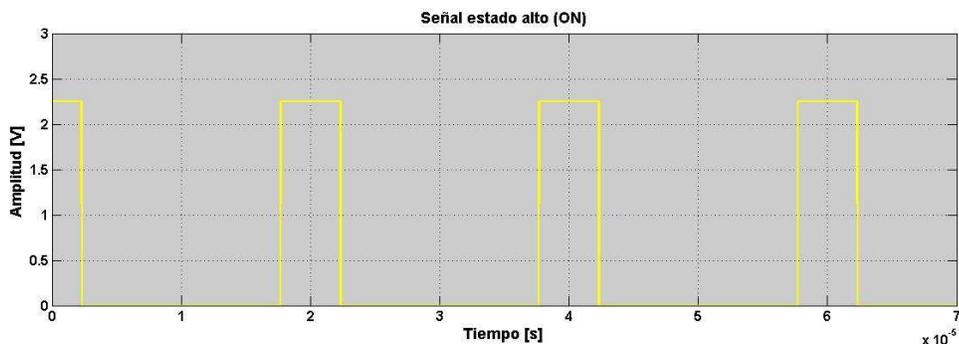


Figura 31. Señal de estado alto para modular el láser

2) Utiliza el comando HOME de los rotadores PRM1Z8 para llevar el motor a la posición de 0 grados. Este paso evita errores a posteriori en el posicionamiento angular con respecto a la lista de ángulos cargados. La comunicación de los rotadores se realizó mediante MATLAB a través de LabVIEW. En la sección 4.2.1.4, se explicará con más detalle el control de las plataformas motorizadas.

3) Realiza la lectura del listado de ángulos de ambos hijos (figura 32). Rota primero la plataforma motorizada para el ángulo correspondiente al hijo A y luego, gira a la posición angular del hijo B. Estos valores, se salvaguardan en un vector (arraylistado de la figura 33), que luego serán utilizados para acceder en cada ciclo a la posición y valor del ángulo al que se corresponde.

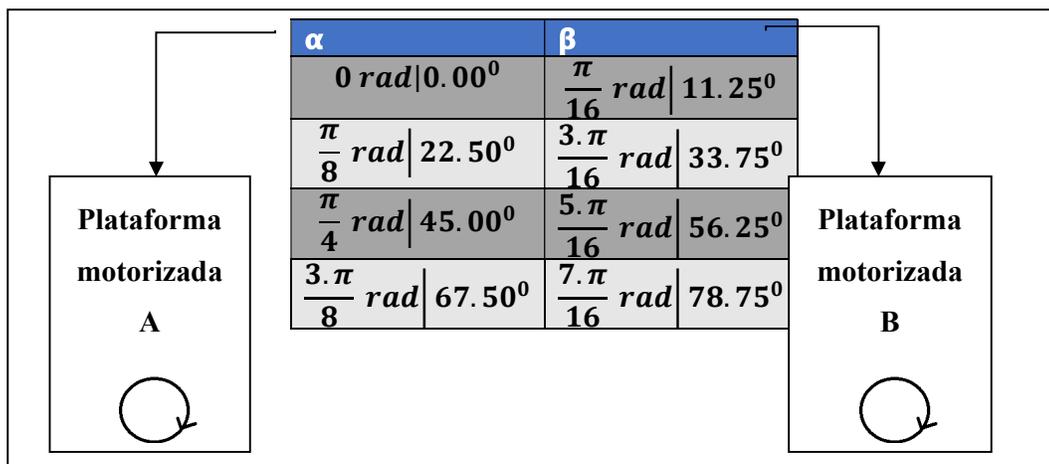


Figura 32. Listado de ángulos. Ejemplificación arbitraria.

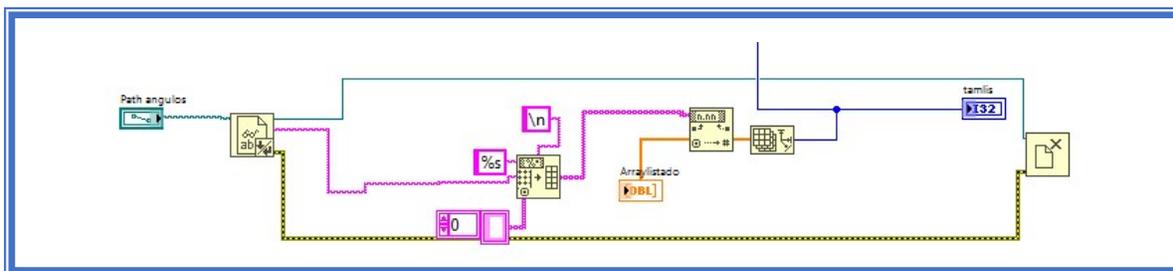


Figura 33. Lectura de archivo de ángulos y salvaguarda en un vector.

4) Una vez que ha realizado el movimiento de ambos rotadores, envía a sus dos hijos, el comando de inicio de actividad. Para que simultáneamente y en forma independiente, los hijos realicen la adquisición de señales.

Los pasos 2 a 3 se repiten tantas veces como cantidad de ángulos que haya que rotar. Una vez que finaliza el movimiento de las plataformas motorizadas con la cantidad total de posiciones de ambos hijos, la madre espera a que los hijos terminen su última adquisición y que se desconecten de ella. Luego, deshabilita la señal de modulación del láser, es decir, apaga el láser y termina el programa.

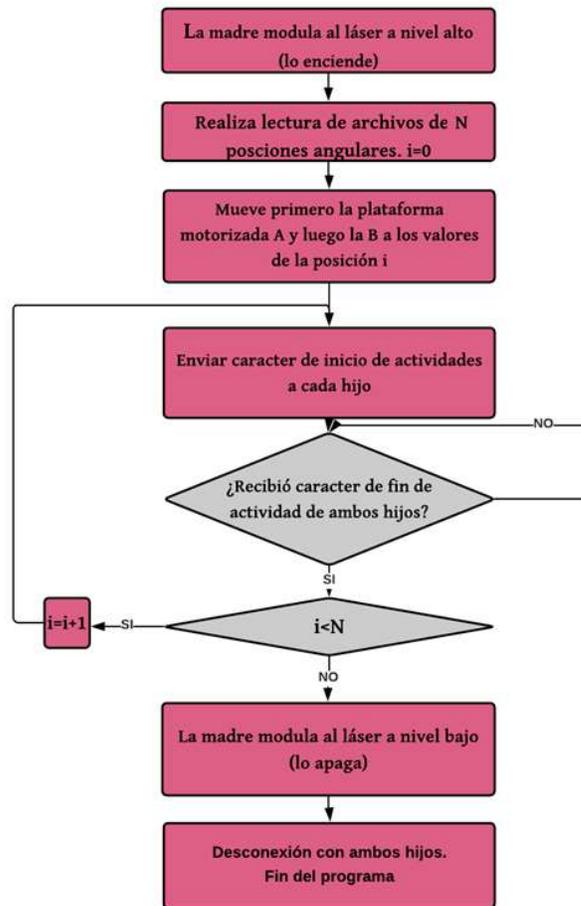


Figura 34. Diagrama de Flujo extendido de la comunicación madre

4.2.4 Programación de dispositivos:

4.2.4.1 Control de rotadores

Para poder utilizar el rotador o plataforma motorizada controlada desde la computadora se deben instalar los drivers de este desde la página web del fabricante [23].

Para el control de la plataforma motorizada, se utilizó el “MATLAB script node”, que permite ejecutar código de MATLAB a través de LabVIEW. La razón por la cual se utilizó este método de manipular mediante el Activex del fabricante a través de MATLAB, es porque el dispositivo presentaba pequeños desfases angulares si se empleaba directamente con el Activex implementado para LabVIEW del fabricante. Esto fue evidenciado en unas mediciones preliminares de laboratorio, cuando al levantar curvas de las mediciones, los valores obtenidos para ciertas posiciones angulares no concordaban con lo esperado. Si bien, a simple vista, aparentaba que el rotador se posicionaba en cierto ángulo, cada vez que se modificaba de posición,

se desfasaba unos grados de más en cada medición, casi imperceptibles al ojo humano, por lo cual, el Activex del fabricante en LabVIEW, generaba errores a lo largo de las mediciones. Sin embargo, en MATLAB, el Activex del fabricante funcionaba correctamente. Por lo cual, se logró implementar desde LabVIEW el control de dicho dispositivo ejecutándolo desde MATLAB. Todo esto, fue posible gracias al MATLAB script node de LabVIEW que como se mencionó, permite integrar códigos desde LabVIEW a través de MATLAB.

Si bien el Activex de MATLAB permite el control del dispositivo, fue necesario además el desarrollo de una lógica adicional para lograr la automatización de dicha plataforma. En el experimento de laboratorio, se posee dos plataformas motorizadas: una correspondiente a la estación A y otra correspondiente a la estación B.

En la Figura 35, se puede visualizar en forma simplificada el diagrama de flujo para el control de las plataformas motorizadas. Los códigos desarrollados para los bloques del diagrama de flujo que se señalan con un círculo a la derecha de los bloques que conforman el diagrama de flujo en dicha figura, que se numeran del 1 al 4, se pueden visualizar en el Apéndice B.

Básicamente, la lógica de control para las plataformas motorizadas es la siguiente:

Se inicia la conexión de la computadora con ambas plataformas motorizadas (A y B). Para ello, se debe identificar cada dispositivo, con su número de serie e iniciar el control como lo indica el código de figura B.1, del Apéndice B.

Se mueven las plataformas motorizadas A y B al origen, es decir, a la posición HOME de cero grados. Esta operación se realiza, ya que las plataformas motorizadas no poseen memoria, esto quiere decir que, si se han dejado las plataformas motorizadas en una posición, por ejemplo, 25 grados y luego, se apaga el dispositivo y se le proporciona la instrucción de moverse a la posición 30 grados, erróneamente, toma la posición 25 grados como la inicial y desplaza al motor a la posición 55 grados en vez de posicionarse en 30 grados. Es por ello, que para que las láminas de media onda que se encuentran montadas sobre estas plataformas motorizadas posean el correcto ángulo de polarización, se debe iniciar el proceso, siempre, en la posición HOME. En la figura B.2 del Apéndice B, se puede observar el código implementado.

Se leen los dos archivos de texto, en donde se encuentran, uno debajo de otro, los valores de los ángulos para cada plataforma motorizada (A y B). Para cada posición del archivo, se procede a

mover cada plataforma al valor del ángulo correspondiente. En la figura B.3 del Apéndice B, se puede observar el código implementado para la plataforma motorizada A.

Finalmente, una vez que se terminaron de ejecutar por las plataformas motorizadas las distintas posiciones angulares de ambos archivos de texto, se procede a desconectar los dispositivos con la computadora.

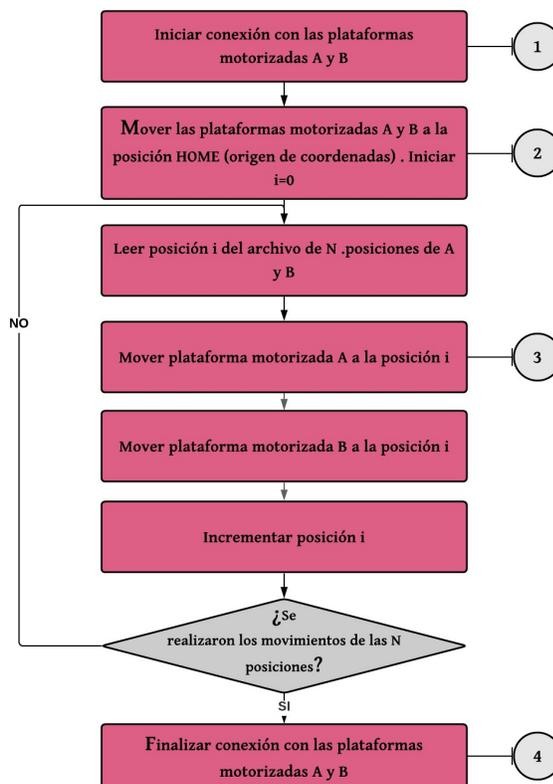


Figura 35. Diagrama de flujo simplificado del control de plataformas motorizadas

En cuanto a la codificación, tanto para los bloques del diagrama de flujo, señalados con 2 y 3, se requirió chequear los bits de estado, para confirmar si las plataformas motorizadas llegaron a la posición angular correspondiente. Para ello, se debió tener en cuenta el protocolo de comunicación del dispositivo [24].

4.2.4.2 Control del generador SDG:

La modulación del láser se realiza mediante una señal provista por el generador de funciones. Durante el proceso de medición, es importante que antes de la adquisición por parte de los TDC de ambas estaciones A y B, la señal del láser haya pasado el transitorio. En las primeras

mediciones, la adquisición comenzaba antes del encendido del láser, lo cual, provocaba errores en cuanto a la detección de los pulsos. Esto se solucionó activando la modulación del láser 10 segundos antes que comenzara el proceso de adquisición, y activando la modulación láser en estado bajo (láser apagado) con posterioridad, es decir, al finalizar todas las adquisiciones y salvaguardas correspondientes, un paso previo a la finalización del programa.

Para la modulación del láser, se utilizó el generador de funciones SDG1025, que se ha descrito en el capítulo 3. El control de este dispositivo se realiza por medio de interfaz USB, en combinación con NI-VISA (*National Instruments -Virtual Instrument Software Architecture*), que es un controlador de instrumentos que ofrece NI para la implementación de normas de entrada y salida de VISA. VISA, son normas para configurar, programar y depurar sistemas de instrumentación que comprenden interfaces GPIB (*General Purpose Interfaz Bus*), VXI (*VME eXtensions for Instrumentation*), PXI (*PC eXtensions for Instrumentation*), serie (RS232/RS485), Ethernet/LXI (*LAN based eXtensions for Instrumentation*) y/o interfaces USB [25].

La lógica de la programación de este dispositivo se muestra en la figura 36, en donde puede observarse un diagrama de flujo con los pasos que se comentan a continuación:

- a) La computadora se conecta con el dispositivo.
- b) Se configura la señal a nivel bajo, estado OFF, es decir, con las siguientes características amplitud 0.5V, *Duty* (ciclo de actividad) 20.3%, *DC Offset* (nivel de continua) 1V.
- c) En el programa en LabVIEW, se espera hasta que las plataformas motorizadas vayan a la posición HOME.
- d) Se configura la señal a nivel alto, estado ON, es decir, con las siguientes características amplitud 4.5V, *Duty* 20.3%, *DC Offset* 2.25V. En este paso, el láser comienza a modularse.
- e) Se espera hasta que los hijos A y B terminen de realizar la totalidad de *time stamping* para todos los ángulos del listado.
- f) Se configura nuevamente a estado OFF y la computadora se desconecta con el dispositivo.

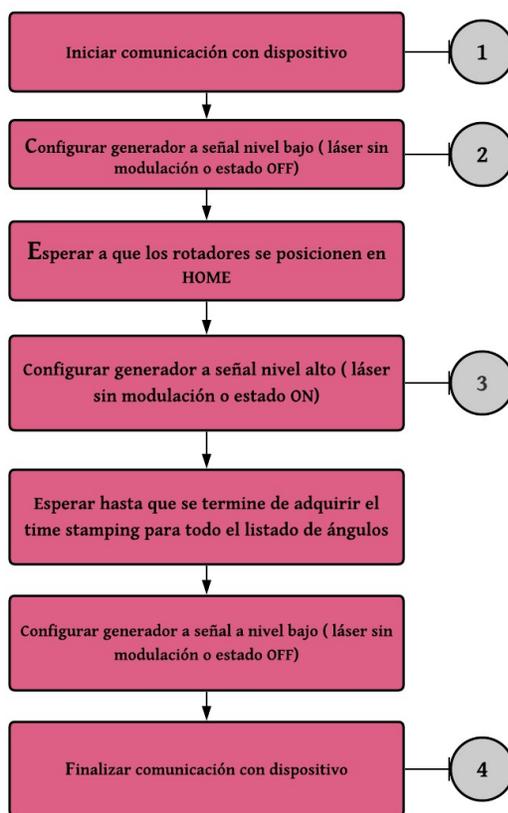


Figura 36. Diagrama de flujo del control con generador de señales

/-

Para la programación en LabVIEW del generador de señales es necesario incluir en el proyecto la librería Siglent SDG 1000 5000 800 Series.lvlib [26]. Dicha librería permite incluir bloques como el "initialize" (figura 37) que puede realizar la conexión con el generador de señales, con este, se realiza el punto 1 del diagrama de flujo de la figura 36; el "close" (figura 39) para efectuar la desconexión del dispositivo como el punto 4 del diagrama de flujo de la figura 36. Si bien existe el bloque para configurar el tipo de señal "configure waveform", se tuvo que modificar dicho bloque para incorporarle el duty cycle (ciclo de actividad). Para ello, se tuvo que editar el contenido del correspondiente bloque, ya que el bloque original, no lo contenía. Una vez modificado el bloque, se puede realizar tanto el punto 2 como 3 de la figura 36, insertando los valores correspondientes en cada caso (frecuencia, amplitud, etc.).

La computadora puede comandar el generador de señales mediante un cable USB tipo A/B a través de instrucciones que le envía mediante LabVIEW. El manual de usuario del generador de funciones se puede obtener de la página web del fabricante [27]. Para el encendido del láser, se lo debe modular con una señal estado ON (señal pulso, amplitud 4,5Vpp, offset 2,25V, ciclo de

actividad 20.3%, frecuencia 200kHz) por medio del generador de señales. Para el apagado del láser, se debe modular con una señal estado OFF (señal pulso, amplitud 0,5Vpp, offset 0V, ciclo de actividad 20.3%, frecuencia 200kHz) por medio del generador de señales. En la figura 38, se muestra el diagrama en bloques en LabVIEW para la configuración de ambos estados.

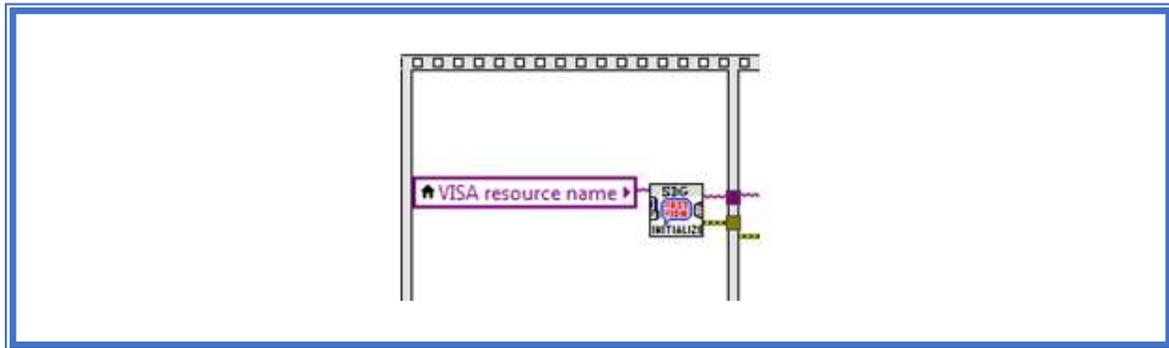


Figura 37. Diagrama en bloques de LabVIEW para la activación del generador de funciones

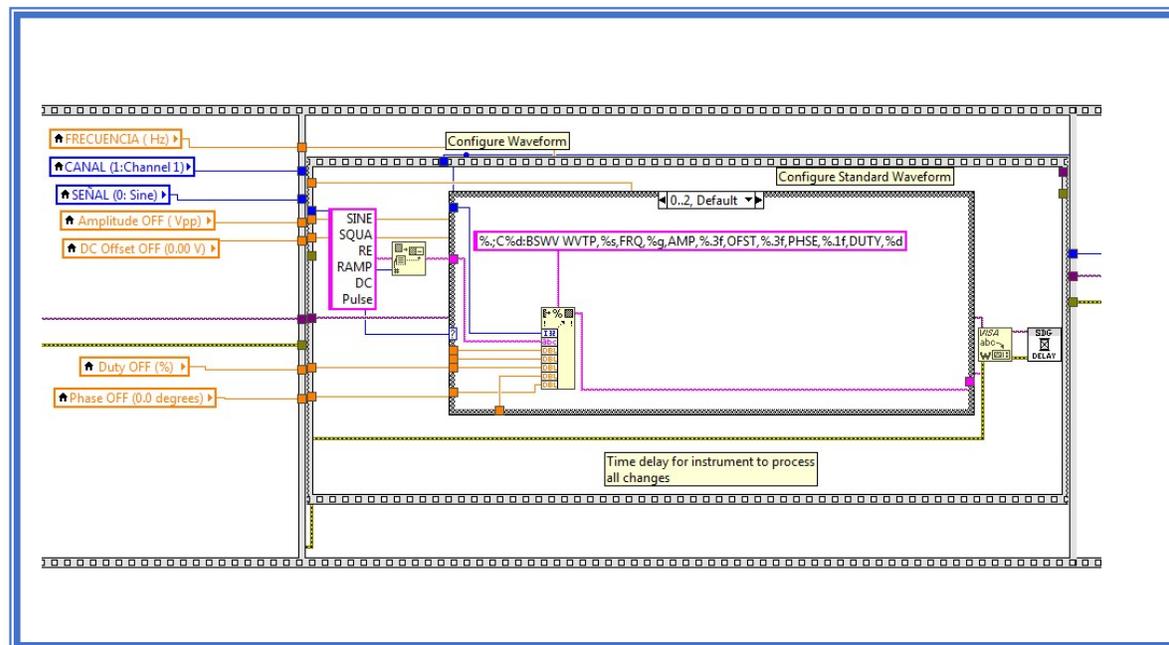


Figura 38. Diagrama en bloques de LabVIEW para la configuración del estado ON/OFF del generador de funciones

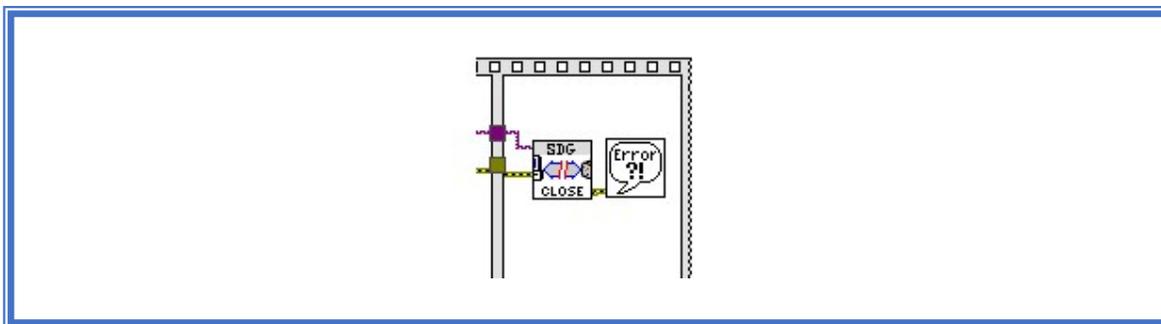


Figura 39. Diagrama en bloques de LabVIEW para la desactivación del generador de funciones

4.2.4.3 Control de ID900

Las principales características técnicas del ID900 se han especificado en el capítulo 3. Aquí, se detallará sobre el control y automatización del dispositivo en simultáneo para la adquisición de *time stamping* de dos estaciones A y B separadas por una distancia de aproximadamente 60m. Cada estación, emplea un TDC ID900.

Si bien el fabricante de los ID900 provee un software, cuando la medición difiere de la mera adquisición de valores, como intervalos temporales específicos, secuencias repetitivas o se quiere seguir un patrón en cuanto a la forma de adquisición de la señal, se hace necesario controlar el dispositivo en forma personalizada, es decir, poder programarlo. La implementación no fue sencilla, ya que el dispositivo, tal vez por ser de reciente lanzamiento, no poseía una versión de firmware estable, que debió ser cambiada más de una vez. Por otra parte, el soporte técnico de la empresa, así como el manual del equipo resultaron de poca ayuda. Esto generó muchos retrasos en la parte experimental, ya que una modificación o actualización, implicaba la edición de comandos por parte del fabricante como así también al cambio en cuanto a la transferencia de paquetes de datos por parte del dispositivo con la computadora y, por lo tanto, los desarrollos realizados, ya no resultaban compatibles con la versión del firmware utilizada y se debía realizar nuevamente los programas. En los programas desarrollados en su versión final, utilizaron la versión de firmware V.011 correspondiente a los ID900.

Al utilizar este TDC, se debe tener en cuenta, que sus entradas deben mantener un voltaje entre -3V a +3V con una carga de 50Ω , por lo cual, se debe tener la precaución de utilizar a la entrada de dichos canales atenuadores para no dañar las entradas o canales del dispositivo.

4.2.4.3.1 Comunicación de los ID900

Para poder realizar el envío de comandos desde las computadoras portátiles a los dispositivos ID900, se utilizó una librería de mensajería asincrónica llamada ZeroMQ. Para incorporar dicha librería al programa, se debe descargar dicho paquete e instalarlo en la aplicación [28].

La gestión de la transferencia de datos, desde el dispositivo hacia la computadora, se realiza mediante un ejecutable provisto por el fabricante. Este, se ejecuta desde el programa en LabVIEW.

La sintaxis de comandos para la comunicación desde la computadora al TDC se realiza por medio de normas SCPI (Standard Commands for Programmable Instruments). En el Apéndice E, se listarán los comandos provistos por el fabricante que fueron utilizados para el desarrollo de los programas de esta tesis.

4.2.4.3.2 Sincronización de relojes internos de ambos ID900

Para realizar una medición simultánea con dos TDC fue necesario, en primer término, sincronizar los relojes de ambos. Para ello, se configura a uno de los TDC como maestro (el que provee el reloj interno de referencia) y el otro como esclavo (que empleará el reloj del maestro). Esto es fundamental para las mediciones de coincidencias, donde se tiene que verificar la simultaneidad, de no realizar este paso, se observan los desfases en los tiempos de cada estación que pueden conducir a resultados erróneos por pérdidas de pulsos.

Para configurar al ID900 como maestro, se debe conectar la salida OUT del ID900 correspondiente a la estación B (figura 40.1) a la entrada IN (figura 41.1) del ID900 correspondiente a la estación A. Sin embargo, la conexión externa, no es suficiente para la configuración. Se lo debe indicar en el programa en LabVIEW con el siguiente comando por medio de TCP/IP al puerto 5555: DEVICE: SYNC INT, para el caso del maestro, y DEVICE: SYNC EXT, para el caso del esclavo.

El ID900 configurado como maestro, mostrará sus luces indicativas de color amarillo (figura 40.2) así también como el ID900 configurado como esclavo.



Figura 40. ID900 correspondiente a la estación B. Actúa como maestro:(1) Conexión de la parte trasera, BNC conectado en OUT. (2) Parte delantera.



Figura 41. ID900 correspondiente a la estación A. Actúa como esclavo:(1) conexión de la parte trasera, BNC conectado en INT. La parte delantera es igual a la Figura 40.2.

4.2.4.3.3 Salvaguarda de archivos

Durante un experimento, al manejar diversos archivos, es indispensable guardar los datos de forma ordenada [29]. Nombrar a los archivos de datos de forma distinta y con información extra en su nombre, disminuye el riesgo de confundirlos. En una medición se generan para cada uno de los 4 canales, 16 archivos de datos casi de la misma longitud, correspondientes a las diferentes combinaciones del parámetro angular. Este proceso se repite para cada medición. El volumen de archivos se vuelve, rápidamente, en inmanejable. Es por ello, que el programa desarrollado en LabVIEW, renombra cada archivo con los siguientes datos: fecha de realización del experimento, número del experimento o medición al que se refiere, referencia (si se trata de la estación A o B), canal al que pertenece, posición angular, y valor de dicho ángulo. Los archivos se guardan en formato BIN.

El archivo quedará conformado como sigue:

FECHA_experimento_MEDICIÓN_B_timestamps_CCANAL_posicionPOSICION_VALOR_grados.bin,

Por ejemplo, supongamos que la fecha del experimento (FECHA) es el día 11, mes 05 y año 20; que el experimento sea el primer experimento (MEDICIÓN) 1 de la estación B; que el archivo guarde los datos para el canal (CCANAL)1; realizando la adquisición de la posición (POSICION) 1; cuyo valor de ángulo (VALOR) es 45 grados; el tipo de archivo será binario (.bin):

11_05_20_experimento_1_B_tiemestamps_C1_posicon1_45_grados.bin,

4.2.4.4 Software desarrollado para el área de señales cuánticas

El control y la automatización de dispositivos involucrados durante los ensayos de laboratorio, para aquellos experimentos de óptica cuántica que requieren de trabajar en completa oscuridad, son una herramienta fundamental para trabajar de forma óptima durante el proceso de medición. Para ello, se desarrollaron programas bajo la plataforma LabVIEW 2017 versión 17.0 (64 bits). Las capturas en pantalla de los programas desarrollados se encuentran en el Apéndice C.

1. Software madre:

Descripción del software madre: El programa madre (ver captura de pantalla del software en la figura C.1 del Apéndice C), básicamente, realiza el control de la rotación de láminas de media onda para diferentes posiciones controladas por rotadores motorizados PRM1Z8 y envía instrucciones a las dos estaciones, A y B. El programa, permite el ingreso de archivos de listados angulares, para los cuales se realizará la rotación automática de las láminas de media onda. Además, admite el ingreso de las direcciones IP de las diferentes estaciones involucradas para la realización de la sincronización de la comunicación. También, habilita la configuración del modulador del láser, para la señal de estado alto (ON) y estado bajo (OFF). El estado actual se puede visualizar mediante una barra de progreso con su comentario correspondiente.

Descripción de los botones del programa madre. El programa se conforma por 4 bloques:

Archivos de posiciones del rotador: habilita a ingresar los número de serie de las plataformas motorizadas de las estaciones A y B (Alice y Bob) y sus archivos de posicione; Direcciones IP: permite ingresar las direcciones IP de las computadoras involucradas en el proceso; Generador de funciones: Admite el ingreso de parámetros para el nivel tanto alto como bajo para la

modulación del láser; Estado actual: actualiza el estado del programa mientras se encuentra en ejecución (actividades que se encuentra realizando).

En cuanto a algunos de los logros de este software que cabe destacar:

Permitió el control al simultáneo de dos rotadores motorizados por medio de comunicación USB, de cada estación (A y B) y de forma automática. Esto implicó un gran avance, ya que los experimentos en el laboratorio para experimentos relacionados con óptica cuántica que utilizan detectores muy sensibles a la luz obligan a trabajar a oscuras. Cometer un error, manipulando manualmente los ángulos de rotación en donde se insertan las láminas de media onda, es muy factible en plena oscuridad. El programa, contiene una herramienta que permite al usuario subir un archivo .txt con los ángulos a medir y este se encarga automáticamente de mover los rotadores a las posiciones deseadas.

Este software permite la sincronización de la comunicación entre dos computadoras remotas, a través de comunicación TCP/IP que se realiza por una red local del laboratorio de láseres sólidos del DEILAP. Esto notablemente es un gran avance, ya que lo usual era que varios operadores al simultáneo en cada computadora debieran ejecutar la adquisición de la señal del TDC. En este caso, la estación central que posee a la computadora madre, con su software madre, brinda aviso a las otras dos estaciones, evitando luego tener que corregir los datos en el desfasaje de tiempo de reacción de cada operador.

2. Software *time stamping* Alice/Bob

Descripción del Software *time stamping* Alice/Bob: este software *time stamping* Alice/Bob, ver captura de pantalla del software en Figura C.2 del Apéndice C, sincroniza la adquisición de *time stamping* con el programa madre, es decir, recibe el comando de acción de actividad desde el programa madre, e inicia la adquisición de la señal en modo *time stamping* durante un lapso de tiempo. Una vez que finaliza dicha adquisición, salvaguarda los datos y se comunica con el programa madre. El proceso de comunicación-adquisición se repite hasta terminar la totalidad de posiciones angulares a rotar. Cada una se ejecuta en una estación independiente. El programa del hijo B, es exactamente igual al hijo A, con la salvedad que este está configurado de forma distinta. Permite la configuración de cada TDC como maestro o esclavo, a fin de uniformar el clock de ambos dispositivos.

Descripción de los botones del software *time stamping* Alice:

Datos del ejecutable: permite indicar desde donde realizar la lectura del ejecutable del fabricante; Archivos de posiciones del rotador: habilita al usuario realizar el experimento con cualquier listado de ángulos seleccionado; Direcciones IP: se puede ingresar la dirección IP de la computadora madre; settings: se pueden variar los valores de configuración del ID900 (umbrales, habilitar o deshabilitar canales, etc.); estado actual: actualiza el estado del programa mientras se encuentra en ejecución (actividades que se encuentra realizando).

Los botones del hijo B son idénticos al hijo A con la diferencia, que tendrán algunos valores distintos de configuración.

En cuanto a algunos de los logros de este software que cabe destacar:

La sincronización de los *clocks* (relojes) de cada TDC, referido a un único *clock*, es decir, un *clock* de referencia. Esto permite que ninguno de los TDC se desfase en la medición cuando se detectan pulsos TTL en los canales de entrada. El programa permite configurar a cada dispositivo como maestro o esclavo. Esta precisión, es esencial en el procesamiento a posteriori, cuando se desea detectar las coincidencias de fotones.

Al igual que el software madre, este programa se comunica con la estación central o madre, a través de una comunicación TCP/IP en una red local del laboratorio de láseres sólidos del DEILAP. Por un puerto, recibe el comando que le envía la estación central (por ejemplo, comenzar a adquirir *time stamping*) y por otro puerto, le envía a la estación central su estado, si ya ha terminado la instrucción solicitada. A su vez, informa el estado actual al usuario (qué posición angular se encuentra adquiriendo, etc.).

3. Software de coincidencias:

Descripción del programa: este software, figura C.3 del Apéndice C, permite la visualización de las cuentas de fotones por segundo, como así también la gráfica y salvaguarda de la información. Este programa controla automáticamente el TDC.

En cuanto a algunos de los logros de este software que cabe destacar:

Este software resulta de mucha utilidad, para una pre-etapa de medición, por ejemplo, en la alineación del arreglo experimental, en donde se testea si la detección de cuentas se encuentra balanceada en los canales. También, sirve para contar las cuentas de oscuridad inicial. Se destaca por su práctica visualización, la cual permite a simple vista observar la variación de cuentas por

segundo que se obtiene de los distintos canales del TDC, mientras se realiza en tiempo real el ajuste fino del arreglo, es decir, mientras el usuario se encuentra efectuando la alineación del sistema.

4.3 Automatización, control y procesamiento para señales clásicas

Para el análisis dinámico de la señal de salida del láser con modulación de pérdidas, se realizó un programa en LabVIEW (ver en figura 42, en donde se puede visualizar el diagrama de flujo de la lógica de decisión de dicho programa). Este, adquiere la señal de salida del láser en tiempo real. Permite al usuario seleccionar el instante en el que se quiere analizar la señal: se adquiere señal, se realiza la detección de picos, se obtiene una serie temporal con los picos de amplitud de la señal. A partir de esta serie, se calcula la probabilidad algorítmica con el coeficiente de Kolmogorov, dicho coeficiente lo denotaremos con la letra K . A partir del valor de K , pueden surgir cuatro casos:

Caso 1, valor de K cercano entre 0.999 y 1.099: en este caso se sospecha que la serie temporal puede ser aleatoria. Para su corroboración, se ejecutan los 15 tests del NIST. Con los resultados de cada test, si analiza si es o no es aleatoria. Si los tests del NIST descartan aleatoriedad, es un resultado irrefutable, la serie temporal no es aleatoria. La aleatoriedad solo se puede refutar con certeza, es decir, se puede afirmar que no es aleatorio, mientras que para afirmar aleatoriedad solo se puede decir que es compatible con aleatoriedad. En el caso de refutar aleatoriedad, surge la duda si dicha serie es caótica, por lo cual, se evalúa el caso 2.

Caso 2, valor de K cercano a valor intermedio entre 0 y 1: en este caso se sospecha que la serie temporal puede ser caótica. Se calcula la dimensión de *embedding*. Si no posee dimensión de *embedding* finita, entonces no es caótica. En el caso de encontrar dimensión de *embedding* finita, se calcula los coeficientes de Lyapunov. En el caso que la sumatoria de los coeficientes de Lyapunov sea negativa y exista al menos un coeficiente positivo, entonces es una serie caótica. Si esto último no se cumple no se puede afirmar que la serie sea caótica y se ejecutan los 15 tests del NIST, para evaluar posibilidad de aleatoriedad.

Caso 3, valor de K cercano a 0: en este caso significa que la serie temporal es muy compresible, esto se puede deber a que dicha serie proviene de un proceso periódico o bien de una función, como por ejemplo una recta, la cual no es periódica. Para la corroboración de periodicidad, se aplica el primer test del NIST. Si el resultado arroja que no es aleatorio, entonces, la serie es periódica. En caso contrario, no es periódica.

Caso 4, valor de K no se encuentra comprendido en ninguno de los casos anteriores: en este caso no se puede decidir el tipo de señal a la que pertenece la serie temporal.

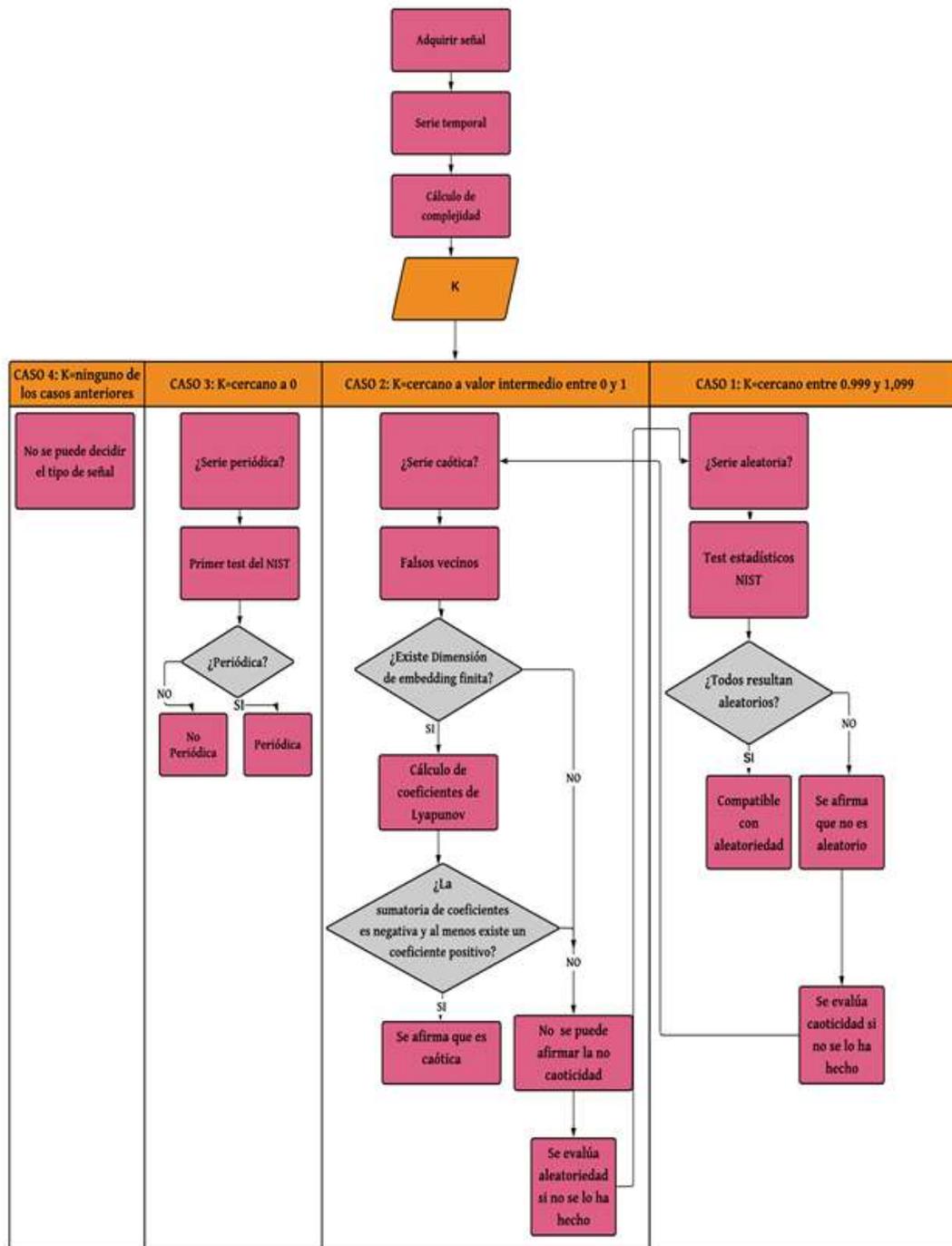


Figura 42. Diagrama de flujo de la lógica de decisión para el análisis del tipo de señal en una serie temporal.

4.3.1 Fotodetector

Como se vio en el capítulo 3, el fotodetector se ubica a la salida del espejo de salida (figura 13.2), transforma la emisión de salida del láser en una señal eléctrica proporcional a la intensidad de luz detectada. A través de un BNC, que se conecta a la salida del fotodetector, se obtiene la señal de salida del láser. Para que estas señales analógicas puedan ser interpretadas por la computadora, se utiliza un osciloscopio de PC, marca PicoScope modelo 6403B.

4.3.2 Control del PicoScope

El PicoScope, que es un osciloscopio digital, permite la adquisición de la señal que proviene de la salida del BNC del fotodetector y su muestreo. Esta señal muestreada, se envía a la PC por medio del puerto USB. Si bien el PicoScope posee un software del fabricante, resulta poco práctico para los casos en donde se quiere adquirir en tiempo real y procesar a la vez. Por lo cual, es necesaria la implementación de un programa propio. Para ello, se utilizó el kit de desarrollo SDK (*Software Development Kit*) que proporciona el fabricante para realizar software propio con el PicoScope [30]. Este kit incluye drivers para distintos sistemas operativos como así también códigos ejemplificativos que explican de forma básica cómo realizar interfaces con otros paquetes de software de terceros (LabVIEW, MATLAB, C, C#, etc.). Para la realización de los programas desarrollados, para la parte de la adquisición con el PicoScope, se utilizó como base el programa "Picoscope6000ExampleStreaming.vi" del kit de desarrollo provisto por el fabricante.

En la figura 43, se puede observar un diagrama de flujo para la adquisición en tiempo real y para el procesamiento del láser con modulación de pérdidas. Se inicia la comunicación del PicoScope con la PC (este bloque del diagrama se indica con un círculo con el número 1 en la figura 43). Su conexión en diagramas en bloques implementado en LabVIEW se puede visualizar mediante la figura 44.1 con el empleo del bloque "PS6000 Open". Luego, se procede a realizar la configuración del PicoScope (este bloque del diagrama se indica con un círculo con el número 2 en la figura 43 y su conexión en LabVIEW se visualiza en la figura 45), logrando configurar el rango de adquisición (máximo de tensión en Volts), el acoplamiento de señal (DC 1MOhm, DC 50Ohm, AC), el *offset* (nivel de continua) y el *trigger* (nivel de disparo). A continuación, si se desea realizar el procesamiento de la señal que se encuentra en adquisición en tiempo real, se debe obtener un bloque de datos de la señal en ese instante (este bloque del diagrama se indica con un círculo con el número 3 en el diagrama de la figura 43). Su diagrama en bloques implementado en LabVIEW se puede observar en la figura 46. Para la obtención de dicho bloque, se adquiere una tira de datos

utilizando el bloque "PS6000 Getstr" y se lo guarda en un vector. Con el bloque de datos (este paso se visualiza en con un círculo con el número 4 en el diagrama de la figura 43), se procede a analizar las muestras con la lógica de decisión que ya se ha visto con anterioridad (en la figura 42). En el caso que se quiera continuar con la adquisición de la señal en tiempo real al igual que su procesamiento, se debe ajustar la señal, variando como parámetro la frecuencia de la señal con el generador de funciones, y volviendo a realizar los pasos comentados con anterioridad. Si se desea abandonar el procesamiento, como se indica con el número 4 en el diagrama de flujo de la figura 43, se utiliza el bloque "PS600 Close" como se puede visualizar en la figura 44.2.

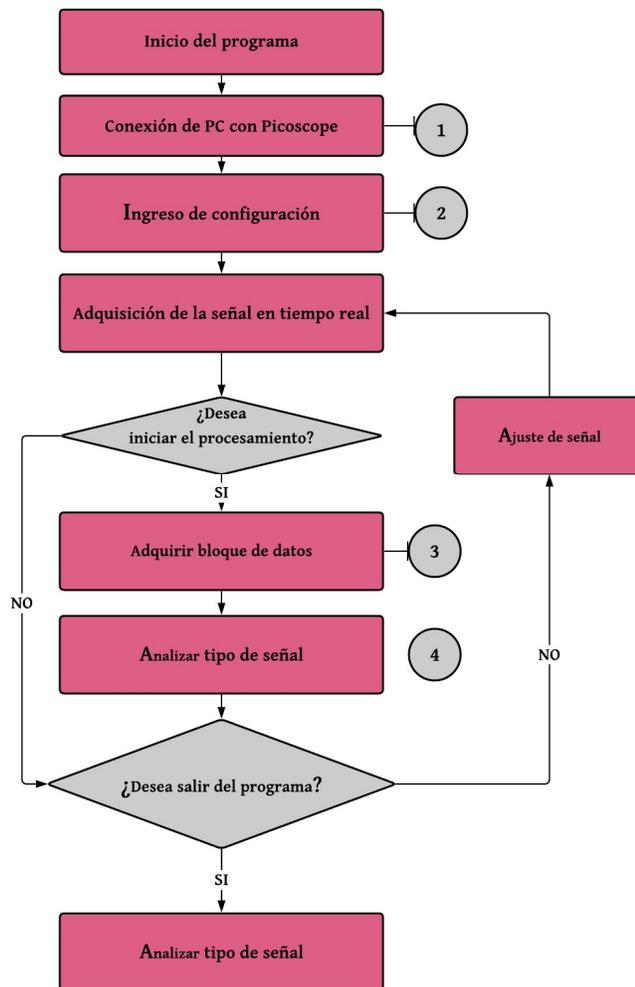


Figura 43. Diagrama de flujo del software de streaming para láser con modulación de pérdidas.

En la figura 44.1, se muestra en LabVIEW el diagrama en bloques para realizar la conexión y desconexión del PicoScope con la computadora a través del puerto USB.

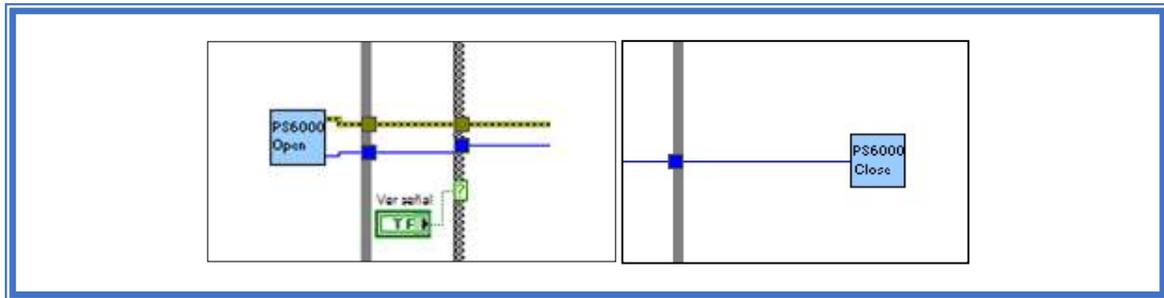


Figura 44. (1) Diagrama en bloques en LabVIEW de apertura de la comunicación de la PC con el PicoScope del kit SDK. (2) Diagrama en bloques en LabVIEW de cierre de la comunicación de la PC con el PicoScope del kit SDK.

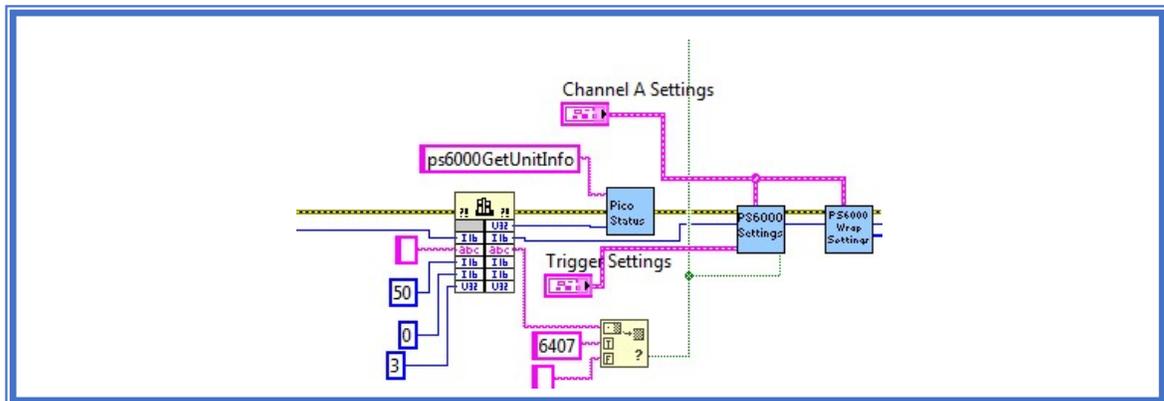


Figura 45. Diagrama en bloques en LabVIEW para la configuración del PicoScope del kit SDK.

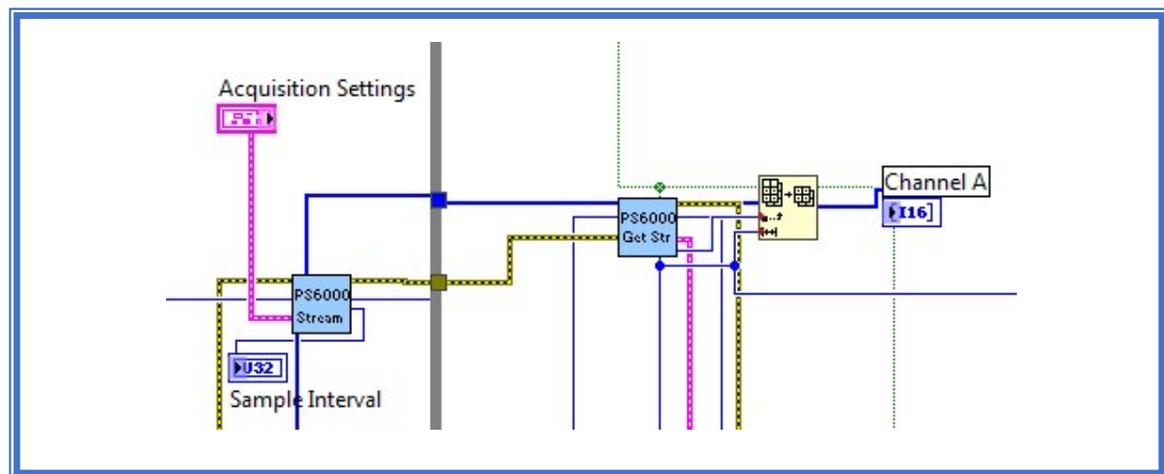


Figura 46. Diagrama en bloques en LabVIEW para la adquisición del bloque de datos del kit SDK.

El programa provisto por el fabricante "Picoscope6000ExampleStreaming.vi", posee la limitación que permite como máximo un buffer de 50 mil muestras, lo cual, para nuestro trabajo, es insuficiente para realizar el análisis de la señal. Por lo cual, se lo debió modificar, para guardar los datos asincrónicos, de forma tal de conformar nuestro bloque de datos de tamaño variable según el experimento en cuestión.

4.3.3 Control del generador de funciones

En cuanto al control del generador de funciones, para realizar el ajuste de la señal visto en el diagrama de flujo de la figura 43, se utilizan los mismos diagramas en bloques que ya se han explicado en la sección 4.2.4.2 para el caso de señales cuánticas. El ajuste de la señal consiste en la variación de un parámetro físico del sistema, la frecuencia de modulación del láser. Para ello, desde el programa, se ajusta el valor de dicha frecuencia y obtiene la adquisición de la señal experimental en tiempo real con dicha variación, permitiendo modificar la frecuencia y procesar la señal para la detección del tipo de señal experimental durante una medición.

4.3.4 Algoritmos de procesamiento

Implementación de algoritmo de filtrado, detección de picos y generación de series temporales:

El código fuente de los algoritmos de procesamiento desarrollados/utilizados para esta tesis, se encuentran en el Apéndice D. Para el caso de señales clásicas, para lograr el análisis de la señal, es necesaria la obtención de series temporales. En el capítulo 1, se han referido a estas series temporales.

Las señales que se obtienen a través del osciloscopio de PC poseen varias muestras por pulso del láser. En la figura 47, se muestra una señal de la salida del láser sin depurar. Como se puede observar, hay varias muestras que no son de nuestro interés. En la ampliación de la señal (dentro del círculo en color verde), se puede visualizar uno de los pulsos, conformado por varias muestras (puntos de la curva en color azul). De este pulso, para nuestro análisis, solo interesa el pico de amplitud para luego conformar la serie temporal.

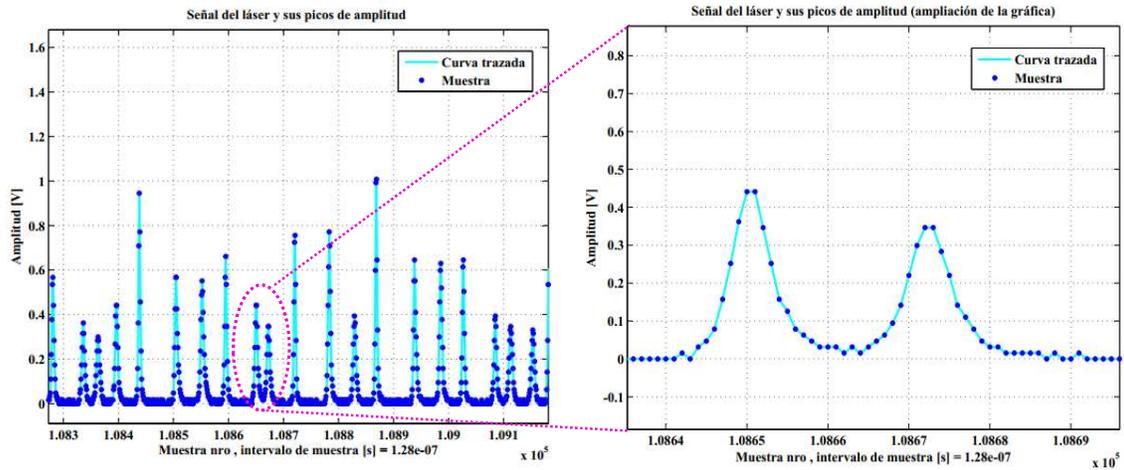


Figura 47. Señal de salida del láser experimental y ampliación de señal.

Es por ello que, como primer paso, se requiere un algoritmo que filtre todos aquellos puntos no deseados, como lo es no analizar la señal por debajo de un cierto umbral. En la figura 47, se podrían omitir las muestras por debajo de un umbral de 0.1 V, estudiando solo las muestras coloreadas en rojo en la figura 48 para la detección de picos.

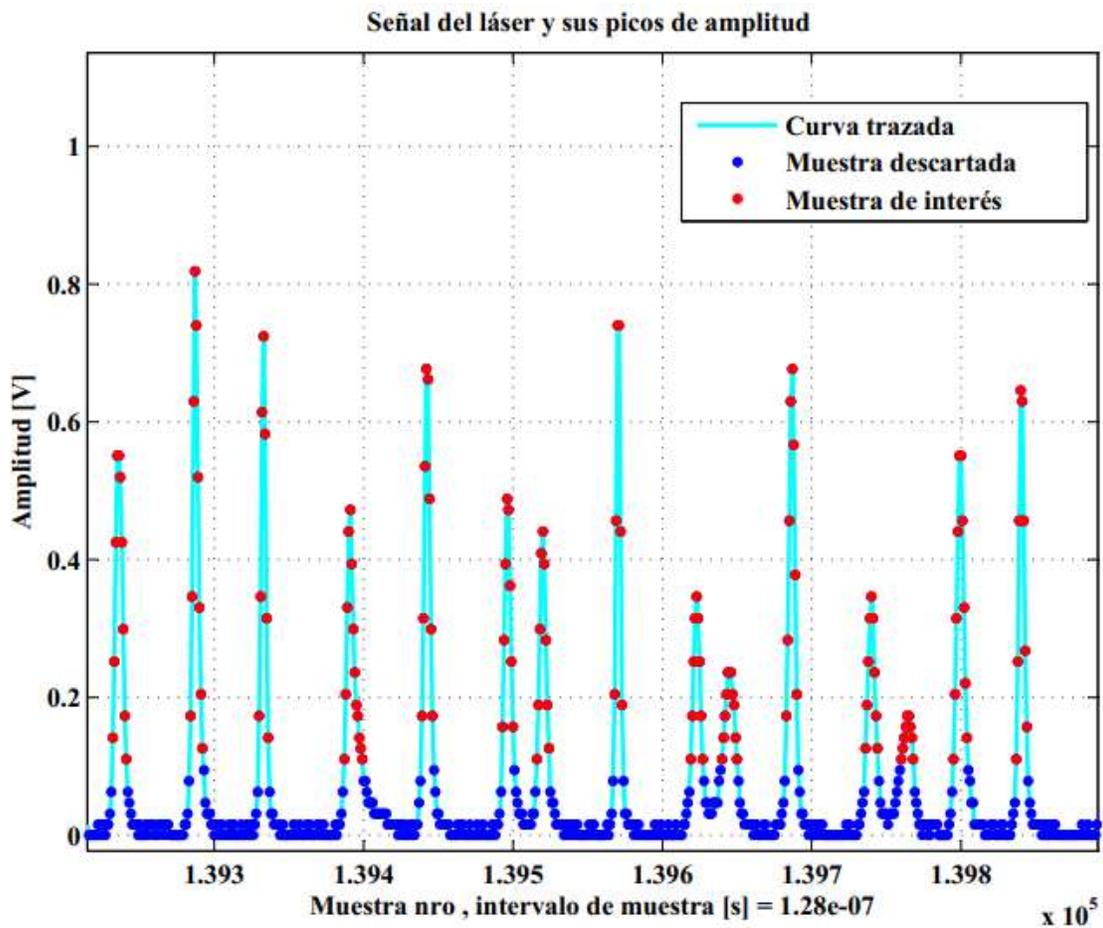


Figura 48. Puntos en rojo, son los puntos que serán de interés para la detección de picos.

Una vez que se tienen las muestras de interés, se procede a la detección de picos. Para ello, se utiliza la función de MATLAB `findpeaks()` con el conjunto de datos conformado por los puntos que fueron obtenidos luego de realizar el filtrado de umbral, con una variable a la cual nombramos `dist`, que modifica el valor de "MIN PEAK DISTANCE" de dicha función de MATLAB, es decir, permite la detección de picos, tan cerca como el valor o cantidad de muestras de valor `dist` haya entre picos. Este valor al igual que el valor de umbral, deben ser elegidos por el usuario.

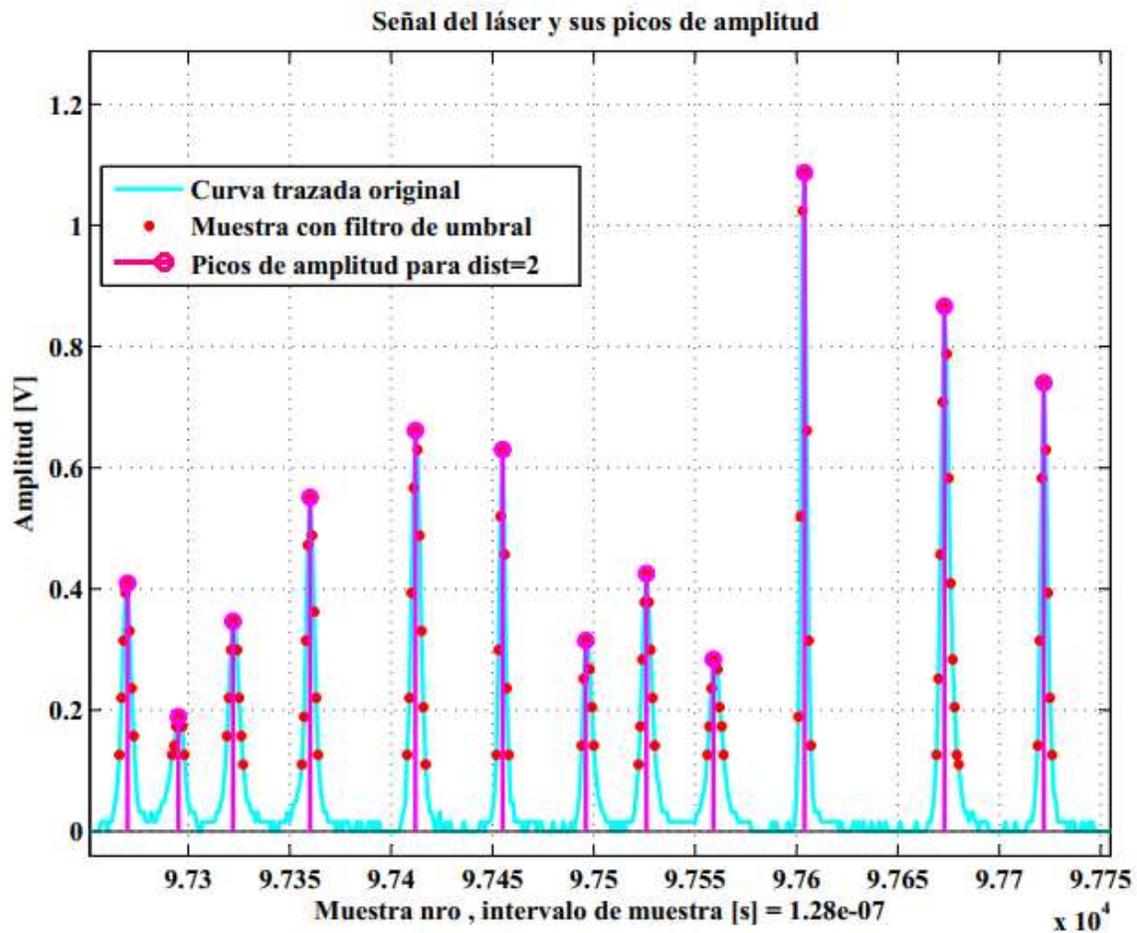


Figura 49. Picos de amplitud de señal en color magenta.

Una vez que se obtienen los picos de amplitud, se conforman las series temporales como se lo mencionó en el capítulo 1 con los picos de señal que se señalan en la figura 49. Las figuras de series temporales se pueden visualizar en las figuras 3 y 4. Básicamente, los dos listados de series temporales quedaran conformados como las tablas 4 y 5. En la tabla 4, se pueden observar los primeros 12 elementos de la serie temporal de picos amplitud de señal. En la columna 1, se puede visualizar cuando ocurre cada pico de amplitud, y en la columna 2, el respectivo valor de amplitud. En la tabla 5, se pueden observar los primeros 12 elementos de la serie temporal de diferencias de tiempo entre picos amplitud de señal. En la columna 1, se puede visualizar el número de pico consecutivo, y en la columna 2, el respectivo valor entre diferencia de tiempo entre picos consecutivos.

Valores de la serie temporal de amplitud de picos de señal		
	1	2
1	1.9200e-06	0.5984
2	9.4720e-06	0.4094
3	1.7280e-05	0.6142
4	2.3296e-05	0.5039
5	2.7392e-05	0.7717
6	3.5840e-05	0.6614
7	4.1984e-05	0.6772
8	4.7232e-05	0.8504
9	5.5680e-05	0.5827
10	6.2080e-05	0.8819
11	7.0016e-05	0.3780
12	7.7696e-05	0.6299

Tabla 4. Primeros 12 elementos de la serie temporal de picos de señal. En la columna 1, se encuentra el tiempo del pico de amplitud y en la columna 2, la intensidad.

Valores de la serie temporal de amplitud de picos de señal		
	1	2
1	1	7.5520e-06
2	2	7.8080e-06
3	3	6.0160e-06
4	4	4.0960e-06
5	5	8.4480e-06
6	6	6.1440e-06
7	7	5.2480e-06
8	8	8.4480e-06
9	9	6.4000e-06
10	10	7.9360e-06
11	11	7.6800e-06
12	12	5.8880e-06

Tabla 5. Primeros 12 elementos de la serie temporal de diferencias de tiempo entre picos de amplitud. En la columna 1, se encuentra el número de pico consecutivo y en la columna 2, el valor entre diferencia de tiempo entre picos consecutivos.

2. Algoritmo complejidad de kolmogorov:

En el capítulo 2, se hizo referencia de la aleatoriedad algorítmica y de una aproximación para calcularla. En el Apéndice D, sección D.4, se presenta el código utilizado para nuestra tesis. Este código ha sido desarrollado por S.Faul et. al [21] basado en el artículo [12]. Se debe tener

en cuenta, que este código, requiere como parámetro, el ingreso de una secuencia binarizada. Por lo cual, se convierte a las series temporales experimentales a formato binario mediante el código de kolbin.m cuyo código se encuentra en el Apéndice D, sección D.4.

3. Algoritmo para la realización de los tests del NIST:

En el capítulo 2, se mencionaron en detalle los diferentes tests del NIST para el discernimiento del tipo de señal aplicando tests de aleatoriedad estadística. En el Apéndice A, se encuentran los códigos fuente de cada test en particular. Estos se han implementado en MATLAB. En la actualidad no existen desarrollos corroborados de dicho código en el mencionado entorno de programación. Los códigos desarrollados se basan en los códigos implementados en C por el NIST y se ha verificado su correcto funcionamiento en base a los resultados provistos por el NIST para cada test con evaluación del resultado de la probabilidad estadística que arroja cada test con las tiras de datos largas provistas por el NIST. En internet, existen algunos códigos implementados en MATLAB pero con errores, no satisfacen los resultados contrastados con los valores provistos por el NIST. Por lo cual, para nuestro trabajo de tesis, fue indispensable el desarrollo de dichos tests en MATLAB, ya que hemos utilizado dicho entorno de programación para la realización del procesamiento de las señales experimentales de la salida del láser.

Si bien se desarrollaron los distintos tests del NIST, resultaba poco práctico el testeado de la aleatoriedad de una serie temporal, ejecutando cada test (en total 15), cada vez que se analizaban los datos. Por lo cual, fue necesario, previo a la implementación de los tests del NIST en LabVIEW, generar una aplicación fácil de operar, que permitiera: subir los datos y aplicarle automáticamente todos los tests del NIST, y que arrojara: los distintos valores de probabilidad como así también si dependiendo de ese valor, se trataba, para cada test en particular, de una secuencia de datos experimentales random (aleatorio) o no random (no aleatorio).

GUI o *app* de tests del NIST:

Descripción breve: Algoritmos de los 15 tests del NIST codificados en MATLAB, herramienta que actualmente no existe implementada en dicho entorno de programación. Se implementó una GUI (*graphical user interface*) o *app*, como interfaz gráfica para que cualquier usuario sin conocimiento de programación pueda ejecutar dichos tests. Permite el ingreso de una serie de datos que luego binariza respecto de la mediana y calcula los tests del NIST (tests que evalúan aleatoriedad estadística). En el caso de no querer evaluar un test específico se lo puede seleccionar para obviarlo en la ejecución del programa. El programa arrojará los valores de

probabilidades, el resultado (si es *random* o no *random*) y si no se puede aplicar el test por no cumplir la muestra de datos con los requerimientos de tamaño. La captura de pantalla de dicha aplicación se encuentra en la Figura C.5 del Apéndice C.

En cuanto a algunos de los logros de este software que cabe destacar:

En la actualidad, como se mencionó con anterioridad, no se encuentran desarrollados en MATLAB los tests del NIST. Algunos códigos que circulan por internet presentan errores y no se encuentran verificados, lo cual, no son confiables para utilizar en una medición. Para esta *app*, se implementaron los 15 tests del NIST, que se corroboraron con los resultados de estadística provistos por el manual del NIST, en donde se testearon varias series de datos con millones de muestras (número pi binario, número en binario, etc.), verificando los resultados de probabilidad con los obtenidos en nuestro software. Poseer estos tests en MATLAB, permiten luego poder automatizar y controlar a nuestro gusto el procesamiento de datos experimentales propios. Sin duda, una herramienta muy poderosa para experimentos tanto para señales cuánticas como clásicas.

4. Algoritmos de decisión para detección de señales caóticas:

TISEAN es un proyecto de software para el análisis de series temporales basadas en la teoría de los sistemas dinámicos determinísticos no lineales [31].

En el capítulo 2, se ha visto sobre la dimensión de *embedding* y sobre los coeficientes de Lyapunov. El método de falsos vecinos es un método para determinar la mínima dimensión de *embedding* [32]. El paquete de TISEAN, posee un algoritmo "false_nearest" para el cálculo de falsos vecinos como así también los coeficientes de Lyapunov "lyap_spec". El funcionamiento de ambos algoritmos se describe en la página web de los desarrolladores de TISEAN [33]. Dichos algoritmos pueden ejecutarse a través de un ejecutable (.exe). Ambos, se utilizaron para adaptarlo al programa desarrollado en LabVIEW. En el Apéndice F, se detalla cómo se implementaron dichos paquetes en LabVIEW para nuestro trabajo de tesis.

5. Algoritmo para la detección de tipo de señales:

Para poder realizar la clasificación del tipo de señal de la salida del láser para el caso del láser con modulación de pérdidas: primero se debe adquirir la señal, obtener un bloque de datos, del cual, se obtendrá una serie temporal. Se realizará el cálculo de la complejidad de Kolmogorov, al cual denominaremos valor K. Luego, se analizará el valor de dicha complejidad según el rango

en el que se encuentre comprendido. Por ejemplo, si el valor de K es cercano a 0, podría tratarse de una señal periódica y deberá confirmarse realizando el primer test del NIST. En cambio, si el valor es cercano a 1, podría tratarse de una señal random y deberán realizarse los 15 tests del NIST para confirmarlo. Si el valor de K fuese un valor intermedio entre 0 y 1, podría tratarse de una señal caótica, y se deberá calcular los falsos vecinos para determinar el valor de la dimensión m finita, y en caso de haberla, calcular los coeficientes de Lyapunov. Si la sumatoria de estos coeficientes es negativa, y al menos, existe un coeficiente positivo, entonces es caótica. En el caso que el valor de K no estuviese comprendido por ninguno, entonces, los datos no son concluyentes como para poder decidir a qué tipo de señal corresponde. En la figura 45, se muestra el diagrama de flujo para decidir el tipo de señal.

4.3.5 Software de streaming para láser con modulación de pérdidas:

Descripción del programa: este programa permite adquirir la señal de salida del láser en tiempo real mostrándola en pantalla. Además, habilita el procesamiento de un bloque de datos elegido durante la adquisición: calcula el coeficiente de Kolmogorov; dependiendo del rango en el cual se encuentra comprendido el coeficiente de Kolmogorov, se aplican los tests de NIST y o el cálculo de la dimensión de *embedding* y coeficientes de Lyapunov según el diagrama de flujo de la figura 43.

Descripción de botones: La pantalla principal se divide en cuatro bloques: la configuración (settings), en donde se pueden ingresar los valores de configuración para el cálculo de la dimensión de *embedding* con la prueba de falsos vecinos, los valores de rango para la clasificación de señales (periódica, caótica y aleatoria), los valores de configuración de adquisición del PicoScope y los tests del NIST; el estado, en donde se puede visualizar el progreso del procesamiento que se encuentra en ejecución como así también una línea de texto en donde se hará un breve comentario de la actividad en ejecución; gráficas y tablas, en donde se puede visualizar la adquisición en tiempo real del PicoScope y las gráficas de la dimensión de *embedding* y tablas de Lyapunov.

RESULTADOS

A continuación, se mostrarán algunos resultados experimentales que se obtuvieron a partir del trabajo de tesis. Se expondrá un paso a paso de las mediciones experimentales realizadas en ambos sistemas físicos mencionados en los capítulos anteriores.

Para el arreglo experimental para el área cuántica, visto en el capítulo 3 sección 3.2, se realizaron mediciones bajo las siguientes condiciones:

Se utilizó el siguiente instrumental de laboratorio:

TDC ID900
Fotodetector PDA100A2 Amplified detector 320-1100 nm (Thorlabs) con atenuador
Osciloscopio Digital Tektronix DPO 4102B-L
Generador de Funciones SDG1025
SPCM-AQR-13-FC de Perkin-Elmer
Plataforma motorizada PRM1Z8 Thorlabs

Para el experimento, se utilizó la configuración de señal para la modulación del láser, que se presenta en la tabla 6. Se observan dos estados: para el estado ON, el láser se encuentra "encendido", para el estado OFF, el láser se encuentra "apagado".

Estado	Tipo de señal de modulación del láser	Frecuencia [Hz]	Amplitud [V]	Offset	Duty o ciclo de actividad [%]
ON	Pulso	200000	4.5	2.25	20.3
OFF	Pulso	200000	0.5	1	20.3

Tabla 6. Tabla de valores de configuración de señal de modulación del láser en estado ON y OFF, para el experimento del arreglo experimental de señales cuánticas.

Para la adquisición con el método *time stamping*, se utilizaron 16 posiciones angulares para rotar a las láminas de media onda montadas en plataformas motorizadas de la estación A, ángulo α , y de la estación B, ángulo β . En la tabla 7, se indican las posiciones angulares utilizadas tanto de α como de β expresadas en radianes y en grados sexagesimales:

Posición angular nro	Plataforma motorizada correspondiente a la estación A, ángulo α : [radianes grados sexagesimales]	Plataforma motorizada correspondiente a la estación B, ángulo β : [radianes grados sexagesimales]
1	0 rad 0.00 ⁰	$\frac{\pi}{16}$ rad 11.25 ⁰
2	0 rad 0.00 ⁰	$\frac{3.\pi}{16}$ rad 33.75 ⁰
3	0 rad 0.00 ⁰	$\frac{5.\pi}{16}$ rad 56.25 ⁰
4	0 rad 0.00 ⁰	$\frac{7.\pi}{16}$ rad 78.75 ⁰
5	$\frac{\pi}{8}$ rad 22.50 ⁰	$\frac{\pi}{16}$ rad 11.25 ⁰
6	$\frac{\pi}{8}$ rad 22.50 ⁰	$\frac{3.\pi}{16}$ rad 33.75 ⁰
7	$\frac{\pi}{8}$ rad 22.50 ⁰	$\frac{5.\pi}{16}$ rad 56.25 ⁰
8	$\frac{\pi}{8}$ rad 22.50 ⁰	$\frac{7.\pi}{16}$ rad 78.75 ⁰
9	$\frac{\pi}{4}$ rad 45.00 ⁰	$\frac{\pi}{16}$ rad 11.25 ⁰
10	$\frac{\pi}{4}$ rad 45.00 ⁰	$\frac{3.\pi}{16}$ rad 33.75 ⁰
11	$\frac{\pi}{4}$ rad 45.00 ⁰	$\frac{5.\pi}{16}$ rad 56.25 ⁰
12	$\frac{\pi}{4}$ rad 45.00 ⁰	$\frac{7.\pi}{16}$ rad 78.75 ⁰
13	$\frac{3.\pi}{8}$ rad 67.50 ⁰	$\frac{\pi}{16}$ rad 11.25 ⁰
14	$\frac{3.\pi}{8}$ rad 67.50 ⁰	$\frac{3.\pi}{16}$ rad 33.75 ⁰
15	$\frac{3.\pi}{8}$ rad 67.50 ⁰	$\frac{5.\pi}{16}$ rad 56.25 ⁰
16	$\frac{3.\pi}{8}$ rad 67.50 ⁰	$\frac{7.\pi}{16}$ rad 78.75 ⁰

Tabla 7. Tabla de valores de posiciones angulares correspondientes a los rotadores de las estaciones A y B.

Si bien ya se han mostrado las plataformas motorizadas en el arreglo experimental del capítulo 3 en la figura 50, se muestra una fotografía más en detalle de estas plataformas correspondientes a las estaciones A y B. Las mismas, se moverán de forma automática mediante el control USB comandadas a través de la computadora de la estación central C.



Figura 50. Fotografía de plataformas motorizadas de la estación A y B con láminas de media onda montadas.

En la figura 51, se muestra la computadora de la estación B con el programa de *time stamping*, listo para utilizarse en el experimento. Cada computadora tendrá su respectivo programa en ejecución. A modo ilustrativo, se muestra el programa en ejecución de la PC del hijo B. Abajo de la computadora, se visualiza el TDC ID900 de dicha estación. En los primeros canales, se conectan las salidas de los SPCM y en el tercer canal, la salida del fotodetector con atenuador.



Figura 51. Fotografía de la computadora con el TDC ID900 de la estación B, con el programa de *time stamping*.

Se debe destacar, que en este experimento es esencial respetar el orden de los pasos para efectuar la medición. Trabajar en forma metódica, reduce el riesgo de dañar el instrumental de laboratorio, ya que algunos son sensibles a la luz, y utilizarlos con la luz encendida puede perjudicarlos. En la tabla 8, se muestran los pasos que deben ejecutarse desde la configuración de los dispositivos, cerciorándose de las correspondientes direcciones IP como así también de la correcta identificación de las plataformas motorizadas con su número de serie.

Paso Nro	Descripción
1	Si fuese la primera medición, y las computadoras involucradas no estuvieran encendidas, como primer paso, se debe cerciorar de encender las tres computadoras de las estaciones (A, B y C) y dejarlas un tiempo prudencial para que inicie correctamente su sistema operativo.
2	Si fuese la primera medición, se debe asegurar que las tres computadoras estén conectadas a la misma red de trabajo designada como "LS".
3	Se debe revisar que las IP sean las correspondientes a las que figuran en el software, caso contrario, se debe reingresar el correcto número de IP en el software: IP computadora estación A: 192.168.0.197 IP computadora estación B: 192.168.0.103 IP computadora estación C: 192.168.0.105
4	Se debe abrir el programa (.vi) de LabVIEW en cada computadora.
5	Una vez abierto cada programa, se debe revisar que las configuraciones (estado ON/OFF, umbrales, disparo, etc.) sean acordes al experimento. En este paso, se debe asegurar que estén bien cargados los archivos de los ángulos. Se debe tener precaución que el último ángulo salvaguardado en el .txt se encuentre guardado sin el <i>enter</i> . La computadora madre debe tener los dos archivos de ángulos cargados en las computadoras de las estaciones en A y B. Se debe tener en cuenta los números de serie correspondientes a cada estación: S/N rotador estación A: 83854419 S/N rotador estación B: 83854377 IDQ900 estación A: Dirección IP: 169.254.99.155, S/N: 1734055T010 IDQ900 estación B: Dirección IP: 169.254.99.154, S/N: 1734054T010
6	Se debe encender los TDC ID900
7	Se debe asegurar que el generador de funciones esté encendido y tenga conectado el terminal BNC del láser.
8	Se debe encender la UPS (<i>Uninterruptible Power Supply</i>).
9	Se debe apagar las luces del laboratorio.
10	Se debe encender los SPCM.
11	Se debe correr solamente el programa (.vi) de las computadoras de las estaciones A y B (presionar RUN).
12	Se debe encender la llave de la fuente del láser.
13	Se debe correr el programa (.vi) de la computadora de la estación C (presionar RUN).
14	El programa de la estación C, inicia la comunicación

Tabla 8. Tabla de valores de pasos a realizar el inicio del proceso de medición para el experimento de señales cuánticas.

Luego de haber realizado los 13 pasos que figuran en la tabla 8, las tres computadoras se encuentran listas para realizar el control y automatización del sistema. Lo que primero realiza, es ubicar a las plataformas motorizadas a la posición HOME. A su vez, configura la señal de modulación del láser a su estado ON. Para proceder a la adquisición de archivos de *time stamping* con los cuales se obtendrán a posteriori, resultados experimentales del sistema físico en cuestión. Como resultado experimental, se obtienen 64 archivos por cada computadora (hijo A y B) correspondientes a las estaciones A y B. Para la obtención de estos, se realiza el procedimiento que se muestra en el diagrama de la figura 52.

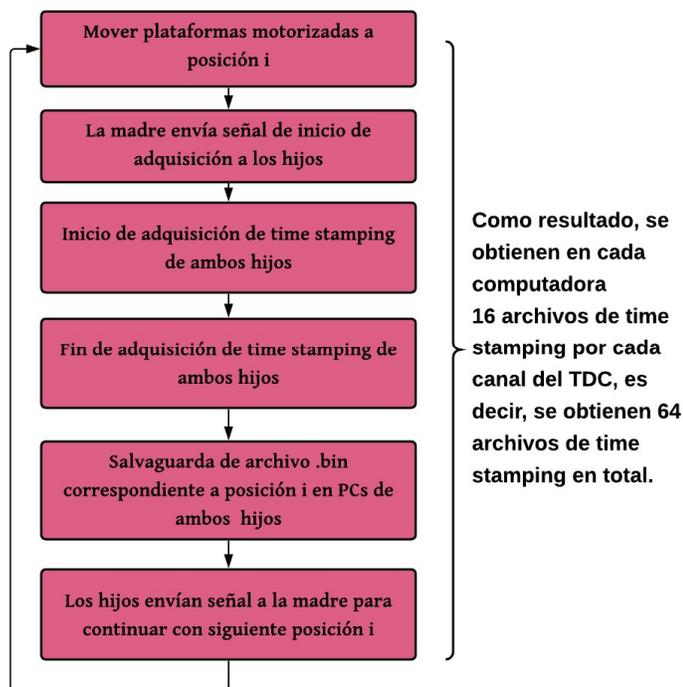


Figura 52. Diagrama para la adquisición de archivos *time stamping*

Una vez que se termina el proceso de la figura 52, se modula a la señal del láser en estado OFF y se termina el programa, finalizando la comunicación entre las tres computadoras de las tres estaciones A, B y C. Con estos 64 archivos salvaguardados en las computadoras de las estaciones A y B, se analizan las coincidencias detectadas. En base al cálculo que se detalla en el apéndice G, se obtuvo un valor del parámetro de Bell $S_{CHSH} = 2.35 \pm 0.01$.

En la figura 53, se muestran las curvas de coincidencias con registro *time stamping* variando la orientación de las láminas de media onda montadas en los rotadores motorizados, α correspondiente al ángulo en la estación A y β correspondiente al ángulo en la estación B. El alto contraste de las curvas indica que el sistema se encuentra bien alineado y que la fuente emite pares de fotones entrelazados en polarización alcanzando un valor del parámetro de Bell $S_{CHSH} = 2.67 \pm 0.01$.

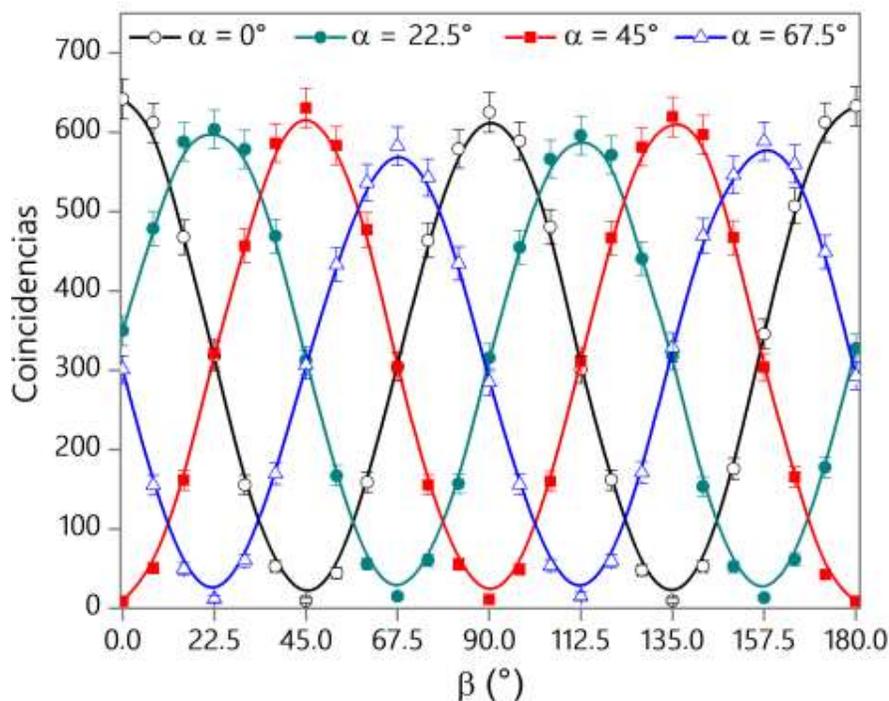


Figura 53. Curvas de contraste obtenidas con registro *time stamping*

Como se pudo ver, se logró la implementación para el control y automatización de un arreglo experimental para experimentos de desigualdad de Bell, único en su tipo en el país. Este desarrollo, permitió obtener propiedades físicas relevantes del sistema como así también permitió trabajar a oscuras de forma sistematizada, reduciendo errores en la manipulación del instrumental.

En cuanto al área clásica, respecto al arreglo experimental visto en el capítulo 3 sección 3, se realizaron mediciones bajo las siguientes condiciones:

Se utilizó el siguiente instrumental de laboratorio:

Fuente del diodo Spectra Diode modelo SDL 805
Amplificador Falcon Systems 50x
Osciloscopio de PC PicoScope modelo 6403B
Osciloscopio Digital Tektronix DPO 4102B-L
Generador de Funciones SDG1025
Fotodetector Thorlabs DET36A

A continuación, se mostrarán los resultados de cuatro mediciones realizadas con el arreglo experimental correspondiente a señales clásicas de diferentes características. En particular, se detallará cómo es paso a paso una de las mediciones (medición nro 1), en la cual, se presenta una señal de carácter no aleatorio.

Para la medición nro 1, se utilizó la configuración de señal para la modulación del láser, que se presenta en la tabla 9. Se observa, además la corriente de la fuente del diodo utilizada en el experimento.

Medición nro 1	
Frecuencia de modulación	180kHz
Corriente de fuente del diodo	1837mA

Tabla 9. Tabla de valores de configuración de señal de modulación del láser y corriente de la fuente del diodo, para el experimento correspondiente al arreglo experimental de señales clásicas de la medición nro 1.

En la tabla 10, se muestran los pasos que deben ejecutarse desde la configuración de los dispositivos, cerciorándose de las correspondientes direcciones IP como así también de la correcta identificación de las plataformas motorizadas con su número de serie.

Se realizaron los siguientes pasos en el proceso de medición:

Paso Nro	Descripción
1	Configuración de la corriente de la fuente de alimentación del diodo.
2	Configuración de la frecuencia de modulación del láser.
3	Adquisición de señal con el osciloscopio de PC (PicoScope).
4	Se muestra la adquisición de la señal en tiempo real en pantalla.
5	Se presiona el botón adquirir, para adquirir el <i>streamming</i> .
6	Se presiona el botón <i>stop</i> (parar), para detener el <i>streamming</i> .
7	Búsqueda o ajuste de parámetros umbral y dist. Se modifican valores y se repite el paso 5 y 6 , hasta encontrar los parámetros para filtrar la señal.
8	Se presiona continuar.
9	Calcula complejidad de kolmogorov, valor K.
10	Se verifica el tipo de señal según sección 4.3, figura 45.
11	Si no se desea salir del programa, se ajusta la señal y se repiten los pasos 5,6,7,8,9 y 10.

Tabla 10. Tabla de los pasos a realizar para la adquisición y procesamiento de señales clásicas de un láser con modulación de pérdidas.

En la figura 54, se muestra las señales de modulación del láser a una frecuencia de 180.2kHz y la señal de salida del láser detectada por un fotodetector. El arreglo experimental correspondiente junto con la PC adquiriendo se puede visualizar en la figura 55.



Figura 54. Captura del osciloscopio de la señal de la modulación del láser en color amarillo y salida del láser en color celeste de la medición nro 1.



Figura 55. Fotografía del arreglo experimental del láser con modulación de pérdidas, adquiriendo con el PicoScope y visualizando la señal con la computadora en tiempo real.

Al ejecutar el software (captura de pantalla del software en apéndice C, figura C.4) para láser con modulación de pérdidas se puede visualizar la adquisición en tiempo real de la señal de la salida del láser. En la figura 56, se observa una captura de pantalla de la gráfica correspondiente al software desarrollado de adquisición en tiempo real de la señal de salida del láser con modulación de pérdidas.

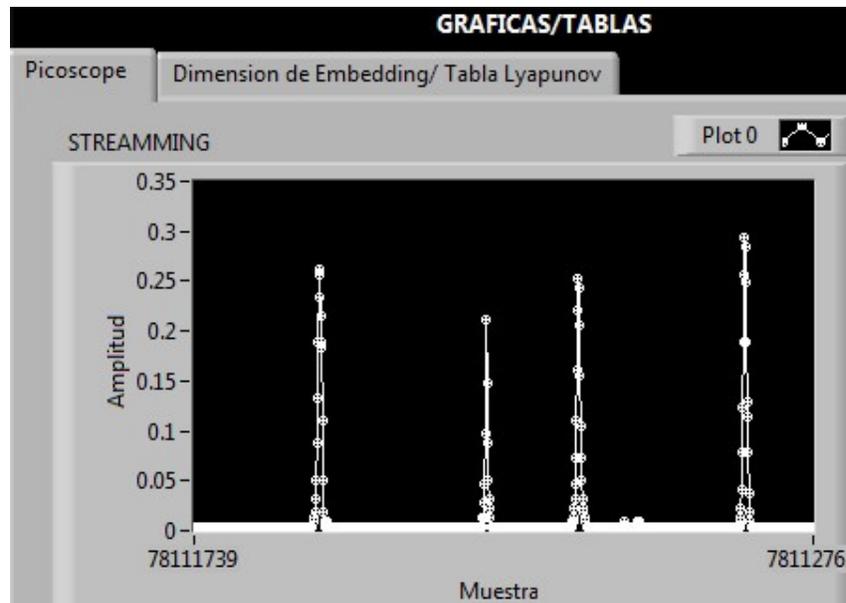


Figura 56. Captura de pantalla de la gráfica correspondiente al software desarrollado de adquisición en tiempo real de la señal de salida del láser con modulación de pérdidas de la medición nro 1.

El software adquiere la señal de salida del láser. Luego, se realiza el ajuste de parámetros para la detección de picos de amplitud con el botón "Ajustar serie" y configurando los valores de "dist" correspondiente a "MINPEAKDISTANCE" de MATLAB, la separación mínima de los picos (entre muestras, cada muestra en este caso se adquiere a 102.4ns) y el "umbral", para filtrar los datos menores a ese valor de referencia. Para esta señal en particular, se realizó un ajuste con la variable $dist=5$ y $umbral=0.05V$. La gráfica correspondiente se obtiene a través de una *Figure* (ventana gráfica) de MATLAB a través de LabVIEW. Dicha ventana con la gráfica de picos de amplitud se puede visualizar en la figura 57, en donde se muestra todo el bloque de señal a analizar, y en círculos verdes, la detección de picos de amplitud. En la figura 58, se puede observar, la misma gráfica de la figura 57 pero con una ampliación de señal.

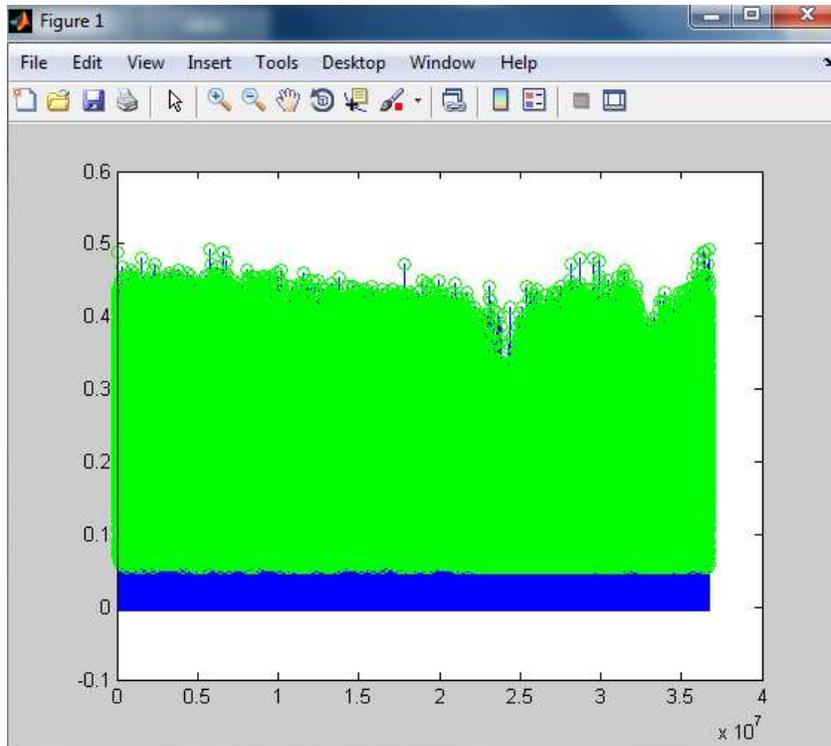


Figura 57. Captura de pantalla de la gráfica que arroja el programa en LabVIEW a través de MATLAB correspondiente a picos detectados de la medición nro 1.

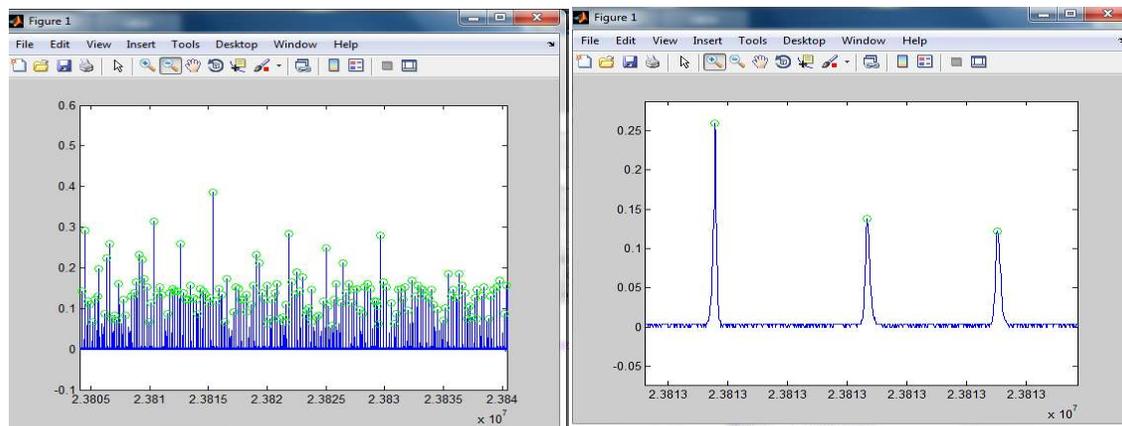


Figura 58. Captura de pantalla de la gráfica que arroja el programa en LabVIEW a través de MATLAB correspondiente a picos detectados de la medición nro 1 (la de la derecha es la misma señal, pero observada más en detalle).

En la señal de la figura 57, se obtienen en total 143692 picos de amplitud. En la tabla 11.1, se observan las primeras 13 posiciones con los valores de los picos de amplitud de la serie temporal. En la tabla 11.2, se puede observar, la binarización de dicha serie temporal de picos de amplitud. Con dicha serie temporal binarizada, el programa procede a calcular el coeficiente

de Kolmogorov. Dicho coeficiente K resulta 0.947. A partir de ese resultado de K, siguiendo la lógica del diagrama de flujos de la sección x del capítulo 4, se aplican los Tests del NIST.

	1		1
1	0.4213	1	1
2	0.2441	2	1
3	0.1063	3	0
4	0.2165	4	1
5	0.1575	5	0
6	0.2205	6	1
7	0.3740	7	1
8	0.2008	8	0
9	0.2323	9	1
10	0.2598	10	1
11	0.2598	11	1
12	0.1811	12	0
13	0.2165	13	1

Tabla 11. 1) Listado de 13 primeros picos de amplitud de la serie temporal en unidades de V de la medición nro 1.
 2) Listado de 13 primeros picos de amplitud de la serie temporal binarizada de la medición nro 1.

En la tabla 12, se muestran los resultados que arroja el software, en los pasos referidos a la tabla 11.

<859461 96 ++PASO Nro	Resultados de medición nro 1	
7	Umbral=0.05V , dist=5	
9	Complejidad de Kolmogorov= 0.947	
10	Tests del NIST	
	Test nro	P-value
	1	2.763E-32
	2	0
	3	0
	4	4.093E-236
	5	0.00319
	6	8.635E-113
	7	0
	8	No aplica
	9	0
	10	No aplica
	11	No aplica
	12	0
	13	0 (reverse y foward)
	14	No aplica
	15	No aplica
	Resultado	
	1	No random
	2	No random
	3	No random
	4	No random
	5	No random
	6	No random
	7	No random
	8	No se puede aplicar n<1E6
	9	No random
	10	No se puede aplicar n<1E6
	11	No se puede aplicar m<[log2n]-2), con m=16
	12	No random
	13	No random
	14	No se puede aplicar n<1E6
	15	No se puede aplicar n<1E6
	Dimensión de embedding=no aplica	
	Coefficientes de Lyapunov= no aplica	
	Tipo de señal= no aleatoria	

Tabla 12. Tabla de resultados de la medición nro 1 al aplicar el software para láser con modulación de pérdidas.

Del resultado del análisis del software, surge que dicha señal es aleatoria según el punto de vista de la probabilidad algorítmica al calcular el coeficiente de Kolmogorov y arrojando un valor cercano a 1. Sin embargo, no es concluyente solo esa prueba para aseverar el carácter de dicha señal. Al realizar los tests del NIST, para evaluar la aleatoriedad desde el punto de vista estadística, nos arroja la no aleatoriedad en los distintos tests. Por lo tanto, dicha señal experimental, se puede afirmar que se trata de una señal no aleatoria. En el caso que los tests hubieran arrojado aleatoriedad, solo se puede decir que la secuencia es compatible con la aleatoriedad, ya que no

existe un conjunto completo de tests que puedan afirmar con certeza la aleatoriedad de una
secuencia. Sin embargo, la no aleatoriedad se puede afirmar.

Con el mismo procedimiento anterior, se obtuvieron señales periódicas (medición nro 2), señales caóticas (medición nro 3) y señales aleatorias (medición nro 4). Dichas gráficas de señal experimental se mostraron en el capítulo en las figuras 6,7 y 8. A continuación, se muestran los resultados de dichas mediciones en particular.

Para el caso de la medición nro 2 (ver tabla 13 de resultados experimentales), se obtuvo una complejidad de Kolmogorov menor a 0.5. El aspecto periódico de la señal se observó en el capítulo 1 en la figura 6. Debido al rango de valor de la complejidad, indicativo de una señal de dichas características, se aplicaron los tests del NIST, observando que el primer test, resultaba no aleatorio. Discerniendo, para este caso, que se trataba de una señal periódica.

Medición nro 2			
Frecuencia de modulación	92.53kHz		
Corriente de fuente del diodo	1930Ma		
PASO nro	Resultados de medición nro 2		
7	Umbral=0.4V, dist=5		
9	Complejidad de Kolmogorov= 0.426		
10	Tests del NIST		
	Test nro	P-value	Resultado
	1	0	No random
	2	0	No random
	3	0	No random
	4	4.826E-260	No random
	5	0.500e-107	No random
	6	0	No random
	7	8.806e-36	No random
	8	No aplica	No se puede aplicar n<1E6
	9	No aplica	No se puede aplicar n<(Q+K)*L
	10	No aplica	No se puede aplicar n<1E6
	11	No aplica	No se puede aplicar m<[log2n]-2), con m=16
	12	0	No random
	13	0 (reverse y foward)	No random
	14	No aplica	No se puede aplicar n<1E6
	15	No aplica	No se puede aplicar n<1E6
	Dimensión de <i>embedding</i> =no aplica. Coeficientes de Lyapunov= no aplica. Tipo de señal= periódica.		

Tabla 13. Tabla de resultados de la medición nro 2 al aplicar el software para láser con modulación de pérdidas.

Para el caso de la medición nro 3 (ver tabla 14 de resultados experimentales), se obtuvo una complejidad de Kolmogorov mayor a 0.5 y menor a 1, el cual, puede indicar que puede tratarse de una señal caótica. El aspecto de la señal se observó en el capítulo 1 en la figura 7, se puede apreciar un semblante con los adjetivos de señal brusca, turbulenta, etc. Debido al rango de valor de la complejidad, se procedió al cálculo de falsos vecinos, para la detección de dimensión de *embedding*, este arrojó un valor finito. Al existir dimensión de *embedding* finita, se calcularon los coeficientes de Lyapunov. Éstos arrojaron que la sumatoria era negativa y al menos un coeficiente era positivo. Resultando, de esta forma una señal del tipo caótica. La caoticidad se puede afirmar. Sin embargo, el caso contrario, la no caoticidad, no se puede afirmar. Se deben realizar tests adicionales para indagar si se trata de otro tipo de señal y se realizan los tests del NIST, para descartar aleatoriedad de la señal.

Medición nro 3	
Frecuencia de modulación	151kHz
Corriente de fuente del diodo	1930mA
PASO nro	Resultados de medición nro 3
7	Umbral=0.4V, dist=5
9	Complejidad de Kolmogorov= 0.927
10	Falsos vecinos y coeficientes de Lyapunov
	Dimensión de <i>embedding</i>=12 Coefficientes de Lyapunov= sumatoria negativa (-1.183) y al menos un coeficiente positivo. Tipo de señal= caótica.

Tabla 14. Tabla de resultados de la medición nro 3 al aplicar el software para láser con modulación de pérdidas.

Para el caso de la medición nro 4 (ver tabla 15 de resultados experimentales), se obtuvo una complejidad de Kolmogorov cercana a 1, el cual puede indicar que puede tratarse de una señal aleatoria. El aspecto de la señal se observó en el capítulo 1 en la figura 8. Debido al rango de valor de la complejidad, se procedió al cálculo de los tests del NIST. Resultando, de esta forma una señal compatible con aleatoriedad.

Medición nro 4			
Frecuencia de modulación	90kHz		
Corriente de fuente del diodo	1602mA		
PASO Nro	Resultados de medición nro 4		
7	Umbral=0.5V, dist=40		
9	Complejidad de Kolmogorov= 1.071		
10	Tests del NIST		
	Test nro	P-value	Resultado
	1	0.455	Es <i>random</i>
	2	0.437	Es <i>random</i>
	3	0.0906	Es <i>random</i>
	4	0.500	Es <i>random</i>
	5	No aplica	No se puede aplicar $n < 38 * Q * M$
	6	No aplica	$n < 1000$
	7	0.995	Es <i>random</i>
	8	No aplica	No se puede aplicar $n < 1E6$
	9	No aplica	No se puede aplicar $n < (Q+K) * L$
	10	No aplica	No se puede aplicar $n < 1E6$
	11	No aplica	No se puede aplicar $m < [\log_2 n] - 2$, con $m = 16$
	12	No aplica	$m \geq \log_2(n) - 5$
	13	0.353 (reverse) y 0.860 (foward)	No <i>random</i>
	14	No aplica	No se puede aplicar $n < 1E6$
	15	No aplica	No se puede aplicar $n < 1E6$
	Dimensión de <i>embedding</i> = no aplica. Coeficientes de Lyapunov= no aplica. Tipo de señal= aleatoria.		

Tabla 15. Tabla de resultados de la medición nro 4 al aplicar el software para láser con modulación de pérdidas.

Como resultados experimentales, tanto de señales cuánticas como clásicas, se pudo extraer información relevante de los sistemas físicos complejos. Por una parte, en la parte del experimento de desigualdades de Bell, se logró, la adquisición de fotones en forma sincronizada de dos estaciones independientes (A y B), se logró obtener mediante el método de *time stamping*, listas de datos, que permitieron arribar al parámetro de entrelazamiento S del sistema, develando de esta forma una propiedad física relevante del sistema, fundamental para la mecánica cuántica. Tanto la automatización como control del sistema fueron incorporados al Laboratorio de Láseres Sólidos del DEILAP, siendo una herramienta de gran utilidad para la reducción de errores en cuanto a la manipulación de instrumentos de laboratorio durante los ensayos a oscuras, además permitió la implementación de un sistema de adquisición sincronizado de tres estaciones mediante comunicación wi fi. Cabe destacar que no existen experimentos de este tipo con automatización y control en el país. Por otro lado, en la parte del experimento de señales clásicas, para láseres de estado sólido con modulación de pérdidas de construcción propia, se pudo lograr la adquisición en tiempo real de la señal experimental, como así también la implementación de algoritmos que permiten determinar el carácter de la señal.

CONCLUSIONES

Este trabajo de tesis contribuyó al desarrollo de novedosas herramientas tecnológicas de gran aporte que son únicas en su tipo en el país tanto para el estudio de la mecánica cuántica como en el área clásica. Esto implicó el desafío de trabajar con arreglos experimentales de sistemas físicos complejos, obteniendo las propiedades fundamentales de una señal física mediante el hallazgo del carácter de una señal calculando el valor de entrelazamiento y discerniendo entre señales caóticas y aleatorias. Esta tesis, permitió generar conocimiento de técnicas experimentales sobre la base de fundamentos teóricos de sistemas físicos complejos, lo cual, es de gran importancia debido a que todas estas técnicas de control, automatización y procesamiento abren puertas a trabajos a futuro de gran interés científico. Además, se destaca una enorme contribución al crecimiento académico-profesional de la tesista como así también a su metodología de trabajo adquirida durante la tesis.

En el transcurso de esta tesis se logró la automatización y control de dos sistemas físicos complejos: uno para señales cuánticas y otro para señales clásicas. Para el primer caso, se pudo implementar un sistema de adquisición de señales en forma sincronizada de dos estaciones independientes (A y B), como así también se consiguió el control automatizado de la rotación de láminas de media onda para experimentos de las desigualdades de Bell con dos estaciones separadas a 60 m aprox., alcanzando un arreglo experimental automatizado y controlado único en su tipo en el país. Para el segundo caso, se consiguió la automatización de la adquisición en tiempo real de una señal experimental proveniente de la salida de un láser de estado sólido de construcción propia con modulación de pérdidas como así también el procesamiento de un bloque de datos extraído de dicha señal y su ajuste mediante la variación de un parámetro del sistema, como la frecuencia de modulación del láser, mediante el control por software desarrollado para esta tesis. Este tipo de ajuste abre puertas para futuras investigaciones que se comentarán en "recomendaciones o consideraciones a futuro" de esta tesis.

A partir de los mencionados sistemas físicos complejos, se pudieron obtener señales experimentales, empleando los métodos de adquisición *time stamping* y series temporales, mediante la extracción de datos experimentales.

Con la implementación de algoritmos de procesamiento desarrollados, se alcanzó a discernir propiedades físicas de los sistemas como así también el carácter de la señal bajo estudio. Además,

se desarrollaron programas por computadora fáciles de utilizar por cualquier usuario para tanto experimentos en el área cuántica como así también para el área clásica.

Como mejora operativa en el laboratorio, en el área cuántica, sin duda, las implementaciones contribuyeron a trabajar de forma sistematizada y ordenada, de forma tal, que no se perdiera ninguna información de la configuración experimental durante el proceso de medición, ya que es muy fácil equivocarse o mezclar archivos para su posterior procesamiento, cuando un experimento puede involucrar 64 archivos por estación. Esto, se logró mediante el nombramiento automático, que salvaguarda los datos de *time stamping* nombrándolos con datos como, por ejemplo, a qué estación corresponde dicha medición, a qué posición angular y valor correspondiente en grados sexagesimales, etc. Además, debido a la sensibilidad de instrumental de laboratorio empleado, se deben realizar las mediciones a oscuras, la sistematización y control de los dispositivos involucrados redujo los errores de manipulación de instrumental. Tener en cuenta que un experimento puede requerir 3hs de medición, a oscuras, por lo cual es muy fácil equivocarse si no se cuenta como un sistema automatizado y controlado como el que se desarrolló en esta tesis, que involucra a tres estaciones (A, B y C), sincronismo de comunicación wi fi entre tres computadoras como de adquisición de datos con dos TDC ID900 con resolución a 10ps, manipulación de plataformas motorizadas, adquisición de datos provenientes de los SPCMs, etc.

A partir de esta tesis, surgieron consideraciones para el futuro para continuar con esta línea de trabajo. Como se ha mencionado en esta tesis, las inestabilidades en la frecuencia de repetición de la clase de láseres (*jitter*) del artículo [1], responde al caos determinístico en baja dimensión. Este hallazgo, más allá de su interés básico, para el área de sistemas dinámicos no lineales, tiene implicaciones prácticas, pues abre la puerta al uso de nuevos y posiblemente más eficientes métodos de estabilización por control del caos.

Las técnicas de control del caos, que convierten una trayectoria caótica en un movimiento regular, han sido con éxito demostradas teóricamente en distintos trabajos. Estas técnicas pueden clasificarse en dos grupos: técnicas de control de realimentación y técnicas de control *non-feedback*. Las técnicas de retroalimentación se basan en el método de control desarrollado por Ott et al [34]. Por ejemplo, una técnica de realimentación proporcional ocasional se aplicó a un sistema caótico por primera vez por Roy et al. [35]. Dado que esta técnica requiere la detección del grado de desviación entre el estado de referencia periódico inestable y el estado caótico en tiempo real, la configuración experimental del sistema de control de retroalimentación proporcional debe ser

un bucle cerrado y por lo tanto tiende a ser relativamente complejo. En las otras técnicas de control *non-feedback* la estabilización de la órbita puede lograrse más fácilmente al menos en teoría. Dado que sólo se requiere una pequeña modulación de un parámetro para controlar el sistema, debido a su naturaleza caótica, un circuito abierto más simple experimentalmente ofrece una alternativa más alcanzable.

El principio implicado en técnicas de control *non-feedback* para la supresión de oscilaciones caóticas es que un sistema se estabiliza a una de las órbitas periódicas que existen en la región de bifurcación antes de la transición al caos mediante la aplicación de una pequeña modulación cerca de la frecuencia resonante de la bifurcación. El período de la oscilación periódica obtenida como resultado del control de caos depende en gran medida de la estructura original del atractor caótico y sus bifurcaciones. Por lo tanto, la oscilación caótica no puede estabilizarse a un patrón periódico arbitrario que no existe en la bifurcación. En el trabajo de Otsuka et al. [36] para un microchip láser con retroalimentación, la salida caótica se estabilizó sólo a una oscilación de período 2 porque se determinó ese período en el diagrama de bifurcación. Por lo tanto, la estructura caótica de un atractor y sus bifurcaciones limitan el número de patrones periódicos a la que la oscilación caótica puede ser estabilizada.

Control y caos en la misma frase parecería contradictorio, pero justamente es el carácter determinístico del caos que hace que sea a la vez predecible y estructurado en el espacio de fase. Esa estructura en el espacio de fase contiene dentro de sí una cantidad muy grande de órbitas cuasi periódicas, topológicamente inestables, y por las cuales el sistema puede pasar a medida que alteramos sus parámetros de control. Es factible, entonces que un ajuste cuidadoso de estos mismos parámetros, utilizando las direcciones estables e inestables conocidas del atractor, permita estabilizar alguna órbita que de otra forma sería inestable. Existen distintos métodos de control que cambian ligeramente el campo vectorial añadiendo grados de libertad dinámicos al sistema. Esto hace que los parámetros anteriormente constantes, pasen a ser dependientes del tiempo y de este modo se puede estabilizar una órbita deseada, inicialmente inestable. Otro tipo de esquema de control remonta su linaje a la teoría de control estándar donde se linealiza el sistema en un entorno de la región a estabilizar. En este enfoque una fuerza externa que actúa sobre las variables dinámicas se añade al campo vectorial mientras que los parámetros de la dinámica originales permanecen inalterados.

Estos métodos también pueden ser vistos como la sincronización de la dinámica original empleando fuerzas externas diseñadas para colocar el sistema en una determinada sub-variedad

del atractor. Estas fuerzas actúan ya sea en los parámetros o las variables dinámicas en el campo de vectores del sistema original, amplían el número de grados de libertad en la dinámica, y cambian la evolución de las órbitas.

En el control mediante la alteración de los parámetros del sistema, el objetivo se está moviendo en una órbita inestable del atractor caótico. El método tiene dos elementos relevantes, el primero es una aplicación de métodos de la teoría de control relativamente estándar comenzando con una linealización alrededor de las órbitas, y el segundo es una idea innovadora sobre cómo utilizar las propiedades de la dinámica no lineal para lograr el control linealizado de forma rápida. Una de las claves para el método es el conocimiento detallado del espacio de fase y la dinámica del sistema. Esto se puede extraer de las observaciones sobre el sistema mediante la adquisición y procesamiento de las señales de este. Una característica atractiva del método es que utiliza pequeños cambios en los valores de parámetros para lograr el control una vez que el sistema ha sido conducido a la vecindad dónde sea apropiada la linealización. La necesidad de cambios de parámetros sólo pequeños proviene de la utilización del colector estable de las órbitas regulares a la que se elige para mover el sistema. Si se cambian los parámetros por grandes cantidades, la estructura de espacio de fases del sistema podría ser cambiado sustancialmente, así, el método se centra en la determinación de la estructura caótica en base a la observación de su dinámica.

Para consideraciones futuras de esta tesis, sería interesante, dado que se ha logrado el procesamiento e identificación de señales caóticas, como así también el ajuste de la señal mediante la variación de la frecuencia de modulación, poder encarar control de señales caóticas experimentales, una vez determinada su existencia. El control del caos es, para definirlo simplemente, la estabilización de una órbita periódica inestable mediante la inyección de una pequeña señal en alguna de las variables del sistema. Desde el primer método propuesto, el OGY (por las iniciales de sus autores Ott, Gregori, Yorke) que data de 1990, numerosas propuestas teóricas han sido desarrolladas para generar control de señales caóticas en diferentes sistemas desde circuitos electrónicos a láseres. Sin embargo, la implementación experimental es mucho más escasa y casi nula en láseres sólidos con cavidades complejas como las que se desarrollaron en esta Tesis. Por otra parte, la estabilización en tiempo real de la señal de láseres sólidos abre un abanico de aplicaciones, ya que, si bien en la actualidad estos láseres son ampliamente utilizados en una gran variedad de aplicaciones, sus inestabilidades hacen que, en ciertos usos, donde la estabilidad es crítica o dónde la frecuencia de repetición determina el *clock* del sistema, estén

vedados. Esto puede tener amplia repercusión técnica al poder reemplazar láseres mucho más complejos y costosos por estos láseres sólidos más simples y compactos.

Por otro lado, una aplicación que podría derivarse del sistema automatizado desarrollado en el trabajo de tesis podría implementarse para el control de movimiento de vehículo autónomo: emitiendo un pulso láser, detectando el rebote de este, para el cálculo de distancias y, mediante un umbral de detección, realizando el control de las distancias en conjunto con el movimiento del motor.

Para el caso de señales cuánticas, sería atractivo el estudio del sistema, llevando las estaciones A y B, a una distancia de separación física mayor, con las estaciones A y B en lugares físicos distantes, y no en un mismo ambiente, como se realizó en este trabajo de tesis. Con el sistema implementado de sincronización de red wi fi, esto posibilitaría la realización de dichas mediciones.

PUBLICACIONES ORIGINADAS POR ESTA TESIS

A continuación, se detallarán las publicaciones originadas por esta tesis:

Se participaron en los siguientes congresos nacionales de la especialidad con referato, presentando los siguientes artículos:

- Nonaka, M., Agüero, M., Bonazzola, C., Kovalsky, M., Hnilo, A. (2017). *Searching for Chaos in Laser Time Series*. 2017 XVII Workshop on Information Processing and Control (RPIC), Mar del Plata, Argentina. IEEE Xplore, doi: 10.23919/RPIC.2017.8214323.
- Nonaka, M., Agüero, M., Kovalsky, M., Hnilo, A. (2017). *A Novel Time Stamping Data Acquisition System Using a Digital Oscilloscope for Single Photon Counting*. XVII Workshop on Information Processing and Control (RPIC), Mar del Plata, Argentina. IEEE Xplore, d.o.i: 10.23919/RPIC.2017.8214324.
- Nonaka, M., Ferrero, H., Agüero, M., Kovalsky, M. (2016). *Diseño e implementación de sistema automático de micromaquinado con láseres de nanosegundos*. Actas del V Congreso Argentino de Ingeniería Mecánica, Santiago del Estero. ISBN-13: ISBN 978-987-1676.
- Nonaka, M., Kovalsky, M., Agüero, M., Hnilo, A. (2015). *Software de control de movimiento de plataforma motorizada a dos ejes para micromaquinado con láser de imágenes 2-D*. Actas de las VIII Jornadas Argentinas de Robótica, Editorial: CEIT, 1/2015. ISBN-13: 978-987-1978-19-9.

Se participaron en los siguientes congresos nacionales con la presentación de posters que se mencionan:

- Nonaka, M., Agüero, M., Hnilo, A., Kovalsky, M. (2019). Presentación de poster: *Desarrollo de sistema automatizado de medición de fotones únicos*. 104^a Reunión Nacional de Física AFA, septiembre 2019, Santa Fe.
- Nonaka, M., Kovalsky, M., Hnilo, A., Agüero, M. (2019). Presentación de poster: *Aleatoriedad y entrelazamiento: un estudio experimental*. 104^a Reunión Nacional de Física AFA, Santa Fe.
- Agüero, M., Nonaka, M., Kovalsky, M., Hnilo, A. (2017). Presentación de poster: *Conteo de fotones con registro time stamping empleando osciloscopio digital para PC aplicado a experimentos de óptica cuántica*. 102^a Reunión Nacional de Física AFA, septiembre 2017, La Plata.
- Santos, N., Villanueva, A., Nonaka, M., Bonazzola, C., Agüero, M., Hnilo, A. (2015). Presentación de póster: *Implementación de controladores manuales de polarización en*

fibras ópticas monomodo. 100^a Reunión Nacional de Física AFA, septiembre de 2015, San Luis.

Se presentó ponencia en Terceras Jornadas Doctorales, UTN FRBA:

- Nonaka, M., Kovalsky, M. (2019). *Procesamiento en tiempo real de series temporales provenientes de sistemas físicos complejo*. Terceras Jornadas Doctorales UTN.FRBA en Procesamiento de Señales e Imágenes.

Se presentaron artículos en revistas nacionales:

- Nonaka, M., Kovalsky, M. (2020). *Desarrollo de sistema automatizado aplicado a arreglo experimental con láser de estado sólido bombeado por diodos láser con modulación de pérdidas para el procesamiento de señales experimentales*. REVISTA TECNOLOGÍA Y CIENCIA. ISSN 1666-6933/DOI: <https://doi.org/10.33414/rtyc.39.162-177.20>
- Nonaka, M., Agüero, M., Hnilo, A., Kovalsky, M. (2020). *Desarrollo en labVIEW y MATLAB para el control y automatización de experimentos en óptica cuántica*. ANALES AFA, vol.31, Nro.2, julio 2020. ISSN 1850-1168/ DOI: <https://doi.org/10.31527/analesafa.2019.31.2.62>.

Se presentó artículo en revista internacional de alto impacto:

- Nonaka, M., Kovalsky, M., Agüero, M., Hnilo, A. (2021). *Testing how different levels of entanglement affect predictability in practical setups*. Quantum Inf Process 20, 165 (2021). DOI: <https://doi.org/10.1007/s11128-021-03110-3>

BIBLIOGRAFÍA

- [1] Kovalsky, M., Hnilo, A. (2010). *Chaos in the pulse spacing of passive Q-switched all-solid state lasers*. Optics Letters, vol. 30, nro 20, pgs 3498 -3500.
- [2] Departamento de física - UBA. *Láser de Nd:YAG. Cavidades de oscilación-Modos transversales*. Apuntes de Laboratorio 5.
- [3] Lacapmesure, A., Mirón Granese, N. *Eventos Extremos en la dinámica de un láser de Nd:YVO4 con modulación de pérdidas: aspectos experimentales*. Trabajo de Laboratorio 6 y 7.
- [4] Koechner, W. (2006). *Solid-State Laser Engineering*. 6ta ed. Ed. Springer.
- [5] Silfvast, W. (2004). *Laser Fundamentals*. 2da ed. Ed. Cambridge, pg 4.
- [6] Khalkhal, E., Rezaei Tavirani, M., Reza Zali, M., Akbari, Z. (2019). *The evaluation of Laser Application in surgery: A Review Article*. Journal of Lasers in Medical Sciences. Review Article, pgs 104-111.
- [7] Agüero, M. (2013). *Fotones en estados entrelazados: mediciones con resolución temporal*. Bblioteca Digital, FCEN-UBA. Tesis Doctoral.
- [8] Mirón Granese, N., Lacapmesure, A., Agüero, M., Kovalsky, M., Hnilo, A., Tredicce, J. (2016). *Extreme events and crises observed in all-solid-state laser with modulation of losses*. Optics Letters. vol.41, prob.13, pgs 3010-3012.
- [9] Motter, A., Campbell, D (2013). *Chaos at Fifty*. Physics Today.vol. 66, pags 27-33.
- [10] Kobayashi, H., Mark, B., Turin, W. (2012). *Probability, random processes, and statistical analysis*. Ed. Cambridge. pg 10.
- [11] Downey, R., Hirschfeldt, D. (2010). *Algorithmic Randomness and complexity*. Ed. Springer. pg. xxi.
- [12] Kaspar, F., Schuster, H. (1987). *Easily calculable measure for the complexity of spatiotemporal patterns*. Physical Review A. vol.36, Num.2, pgs 842-848.
- [13] Rukhin, A., Soto, J., Nechvatal, J., Smid, M., Barker, E., Leigh, S., Levenson, M., Vangel, M., Banks, D., Heckert, A., Dray, J., Vo, S. (2010). *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*. National Institute of Standards and Technology. Special Publication 800-22. Rev. 1a.
- [14] Li, M., Vitányi, P. (2008). *An Introduction to Kolmogorov Complexity and Its Applications*. Editorial Springer, 3ra Edición.
- [15] Zenil, H. (2013). *Complejidad y aleatoriedad*. Revista Ciencia.
- [16] Caparrini, F. *El Teorema de Takens: cómo predecir un sistema a partir de información parcial*. Disponible en pg web: <<http://www.cs.us.es/~fsancho/?e=127>>.
- [17] Abarbanel, H. (1996). *Analysis of Observed Chaotic Data*. Ed. Springer Science-Business Media, 1ra ed., EE.UU.
- [18] Takens, F. (1981). *Detecting strange attractors in turbulence*. Dynamical Systems and Turbulence, Lecture Notes in Mathematics, vol. 898, Springer-Verlag, pgs 366–381.
- [19] Mañé, M. (1981). *On the dimension of the compact invariant sets of certain nonlinear maps*. Dynamical Systems and Turbulence (1981), Lecture Notes in Mathematics, vol. 898. Springer-Verlag. pág. 230–242.

- [20] D. T. Mihailović, G. Mimić, N. Drěsković, Ilija Arsenić (2015). *Kolmogorov Complexity Based Information Measures Applied to the Analyss el of Different River Flow Regimes*. Entropy, vol.17.
- [21] Faul, S. (2005). *kolmogorov.m*. Disponible en <<https://la.mathworks.com/matlabcentral/fileexchange/6886-kolmogorov-complexity>>.
- [22] Manual del PicoScope. Disponible en <[https://www.picotech.com/download/manuals/PicoScope6000 ABSerie-UsersGuide-en-1.pdf](https://www.picotech.com/download/manuals/PicoScope6000ABSerie-UsersGuide-en-1.pdf)>
- [23] Página web de Thorlabs. Disponible en <www.thorlabs.com>
- [24] Manual de programación. Disponible en <Página web: https://www.thorlabs.com/software/apt/APT_Communications_Protocol_Rev_15.pdf>
- [25] Información disponible en página web <<https://www.ni.com/es-cr/shop/select/ni-visa>>
- [26] Descargas de librería. Disponible en página web <<https://siglenta.com/service-and-support/firmware-software/waveform-generators/#sdg1000-series>>
- [27] Información de generador de funciones. Disponible en página web <<https://www.siglent.eu/product/1140255/siglent-sdg1025-25mhz-function-arbitrary-waveform-generator>>
- [28] Librería: ZeroMQ Socket Library. Versión: 3.3.1.102. Autor: Martijn Jasperse. JKI V Package. Disponible en para descarga en VI package manager.
- [29] Nowogrodzki, A. (2020). *Eleven Tips for working with large data sets*. Nature, vol. 577.
- [30] Software Development Kit. Disponible para descarga en página web del PicoScope: <<https://www.picotech.com/downloads>>.
- [31] Hegger, R., Kantz, H., Schreiber, T (1999). *Practical implementation of nonlinear time series methods: The TISEAN package*. CHAOS 9, 413.
- [32] Kennel, M., Brown, R., Abarbanel, H. (1992). *Determining embedding dimension for phase-space reconstruction using a geometrical construction*. Phys. Rev. A 45, 3403.
- [33] Paquete de TISEAN. Disponible para descarga en la página web: <https://www.pks.mpg.de/~tisean/Tisean_3.0.1/index.html>
- [34] Ott, E., Grebogi, C., Yorke, J. (1990). *Controlling Chaos*. Physical Review Letters, vol. 64, nro 11, pg 1196.
- [35] Roy, R., Murphy, T., Maier, T., Gills, Z., Hunt, E. (1991). *Dynamical Control of Chaotic Laser: Experimental Stabilization of Globally Coupled System*. Physical Review Letters, vol. 68, pg 1259.
- [36] Otsuka, k., Chern, J., Lih, J. (1997). *Experimental supression of chaos in a modulated multimode laser*. Optics Letters, vol. 22, pg. 292.
- [37] Código fuente de tests del NIST SP 800-22. Disponible en página web: <<https://csrc.nist.gov/projects/random-bit-generation/documentation-and-software>>
- [38] Descripción del programa false_nearest del paquete TISEAN. Disponible en página web: <https://www.pks.mpg.de/~tisean/TISEAN_2.1/docs/docs_c/false_nearest.html>
- [39] Descripción del programa lyap_spec del paquete de TISEAN. Disponible en página web: https://www.pks.mpg.de/~tisean/TISEAN_2.1/docs/docs_c/lyap_spec.html

[40] Sano, M., Sawada, Y. (1985). *Measurement of the Lyapunov spectrum from a chaotic time series*. Phys. Rev. Lett. **55**, pg. 1082.

APÉNDICE A

CÓDIGOS DE LOS TESTS DEL NIST DESARROLLADOS EN MATLAB

En este apéndice, se mostrará el código en MATLAB de los 15 tests del NIST que se han desarrollado en la tesis basados en el código base en el lenguaje de programación en C [37]. Todos los códigos implementados han sido verificados con datos empíricos provistos por el NIST [13], arrojando idénticos resultados y corroborando de esta forma que los códigos desarrollados en MATLAB son confiables para utilizarlos en el análisis de secuencias aleatorias.

Se implementaron 15 funciones de MATLAB, las cuales corresponderán a los 15 tests del NIST. Sus argumentos y valores retornados por cada uno se explicarán en cada código en particular.

A.1. Código fuente correspondiente al Test 1: The Frequency (Monobit) Test

La función `[nr,noaplicable,p_value,n] = fr1(pat)`, permite el cálculo del test1. Como argumento se le debe ingresar en la variable `pat`, el path o ruta del archivo en donde se guarda la secuencia binaria a testear. Esta secuencia binaria, se encuentra formada por unos y ceros, guardados en una sola columna y varias filas. La función `fr1` calcula la probabilidad `Pvalue` y devuelve dicho valor en la variable `p_value`. Además, nos informa el estado de las variables `nr` y `noaplicable`. En el caso que la secuencia resulte random `nr` será igual a 0, en el caso que sea no random `nr` será igual a 1. En el caso que este test no se pueda aplicar por limitaciones en los requisitos como, por ejemplo, tamaño de la muestra, `noaplicable` será igual a 1. En caso contrario, es decir, si se puede aplicar, `noaplicable` será igual a 0. Además, la función retorna el valor de la longitud de la secuencia, `n`. Todas estas variables, se implementaron para poder realizar un control en el procesamiento de series temporales experimentales.

```
function [nr,noaplicable,p_value,n] = fr1( pat )

% NIST Test Suite en MATLAB
% Traducción del .c del NIST a MATLAB por Myriam Nonaka
% 22-05-2019
%#####
% Descripción:
% Test 1 (The Frequency (Monobit)Test)
%#####
%
% Variables utilizadas:
%
% n= La longitud del archivo de unos y ceros a testear.
% epsilon= Es la secuencia de datos del archivo a testear.
%
%-----
```

% IMPORTANTE: Formato de datos ingresados para epsilon debe ser de vector de unos
% y ceros en una sola columna con i filas. Serie binaria.

```

fileID=fopen(pat);
epsilon = fread(fileID,'int32'); %depende cómo se guarda el archivo a leer, puede ser
'double'
epsilon = epsilon';
n=length(epsilon);
nr=-1;
p_value=-1;
if n>=100
sum=0.0;
for i=1:n
sum=sum+(2*epsilon(i)-1);
end
s_obs=abs(sum)/sqrt(n);
f=s_obs/sqrt(2);
p_value=erfc(f);
%Decision rule
if p_value<.01
disp('Test 1. No es random. p_value<0.01' );
p_value
nr=1;
else
disp('Test1. Es random. p_value>0.01');
p_value
nr=0;
end
noaplicable=0;
else
noaplicable=1;
disp('1-Frequency Test. NO SE PUEDE APLICAR n<100');
end
fclose(fileID);
end

```

A.2. Código fuente correspondiente al Test 2: The Frequency Test within a Block

La función `[nr, noaplicable, p_value,n,chi_squared,N] = FB1(MM,pat)`, permite el cálculo del test 2. Como argumento se le debe ingresar en la variable `pat`, el path o ruta del archivo en donde se guarda la secuencia binaria a testear y el valor de `MM`, la longitud de cada bloque a testear. Esta secuencia binaria, se encuentra formada por unos y ceros, guardados en una sola columna y varias filas. La función `FB1` calcula la probabilidad `Pvalue` y devuelve dicho valor en la variable `p_value`. Además, nos informa el estado de las variables `nr` y `noaplicable`. En el caso que la secuencia resulte random `nr` será igual a 0, en el caso que sea no random `nr` será igual a 1. En el caso que este test no se pueda aplicar por limitaciones en los requisitos como, por ejemplo,

tamaño de la muestra, noaplicable será igual a 1. En caso contrario, es decir, si se puede aplicar, noaplicable será igual a 0. Además, la función retorna el valor de la longitud de la secuencia, n; el valor de chi_squared y el N. Todas estas variables, se implementaron para poder realizar un control en el procesamiento de series temporales experimentales.

```
function [nr, noaplicable, p_value,n,chi_squared,N ] = FB1( MM,pat )

% NIST Test Suite en MATLAB
% Traducción del .c del NIST a MATLAB por Myriam Nonaka
% 22-05-2020
%#####
% Descripción:
% Test 2 Frequency Test within a Block
%#####
%
% Variables utilizadas:
%
% M= La longitud de cada bloque
% n= La longitud del archivo de unos y ceros a testear.
% epsilon= Es la secuencia de datos del archivo a testear.
%
%-----
% Formato de datos ingresados para epsilon debe ser de vector de unos
% y ceros en una sola columna con i filas. Serie binaria.
fileID=fopen(pat);
epsilon = fread(fileID,'int32');
fclose(fileID);
n=length(epsilon);
nr=-1;
p_value=-1;
chi_squared=-1;
N=-1;
if n>=100
M=MM; %M=128
N=floor(n/M);
sum=0.0;
for i=1:1:N
blockSum=0;
for j=1:1:M
blockSum=blockSum+epsilon(j+(i-1)*M);
end
pi=double(blockSum)/double(M);
v=pi-0.5;
sum=sum+v*v;
end

chi_squared=4.0*M*sum;
```

```
p_value=gammainc(chi_squared/2,N/2,'upper'); %gammainc(z, nu) = 1 - igamma(nu,
z)/gamma(nu)
if p_value<.01
disp('2-Block Frequency Test. No es random. p_value<0.01' );
p_value
nr=1;
else
disp('2-Block Frequency Test. Es random. p_value>0.01');
p_value
nr=0;
end
noaplicable=0;
else
disp('2-Block Frequency Test. NO SE PUEDE APLICAR n<100');
noaplicable=1;
end
end
```

A.3. Código fuente correspondiente al Test 3: Runs Test

La función [nr,noaplicable, p_value,n,pi] = ru1(pat), permite el cálculo del test 3. Como argumento se le debe ingresar en la variable pat, el path o ruta del archivo en donde se guarda la secuencia binaria a testear. Esta secuencia binaria, se encuentra formada por unos y ceros, guardados en una sola columna y varias filas. La función ru1 calcula la probabilidad Pvalue y devuelve dicho valor en la variable p_value. Además, nos informa el estado de las variables nr y noaplicable. En el caso que la secuencia resulte random nr será igual a 0, en el caso que sea no random nr será igual a 1. En el caso que este test no se pueda aplicar por limitaciones en los requisitos como, por ejemplo, tamaño de la muestra, noaplicable será igual a 1. En caso contrario, es decir, si se puede aplicar, noaplicable será igual a 0. Además, la función retorna el valor de la longitud de la secuencia, n y el valor de pi. Todas estas variables, se implementaron para poder realizar un control en el procesamiento de series temporales experimentales.

```
function [nr,noaplicable, p_value,n,pi ] = ru1( pat )

% NIST Test Suite en MATLAB
% Traducción del .c del NIST a MATLAB por Myriam Nonaka
% 22-05-2020
%#####
% Descripción:
% Test 3 Runs Test
%#####
%
% Variables utilizadas:
%
% n= La longitud del archivo de unos y ceros a testear.
```

```

% epsilon= Es la secuencia de datos del archivo a testear.
%-----
% Formato de datos ingresados para epsilon debe ser de vector de unos
% y ceros en una sola columna con i filas. Serie binaria.

fileID=fopen(pat);
epsilon = fread(fileID,'int32'); %depende cómo se guarda el archivo a leer, puede ser
'double'
epsilon = epsilon';
n=length(epsilon);
nr=-1;
p_value=-1;
pi=-1;
if n>=100
S=0;
for k=1:1:n
if epsilon(k)
S=S+1;
end
end
pi=double(S/n);
if abs(pi-0.5)>abs(2.0/sqrt(n))
p_value=0.0;
else
V=1;
for k=2:1:n
if epsilon(k)~=epsilon(k-1)
V=V+1;
end
end
erfc_arg=abs(V-2.0*n*pi*(1-pi))/(2.0*pi*(1-pi)*sqrt(2*n));
p_value=erfc(erfc_arg);
end
if p_value<.01
disp('3-Runs Test. No es random. p_value<0.01' );
p_value
nr=1;

else
disp('3-Runs Test. Es random. p_value>0.01');
p_value
nr=0;
end
noaplicable=0;
else
noaplicable=1;
disp('3-Runs Test. NO SE PUEDE APLICAR n<100');
end
fclose(fileID);
end

```

A.4. Código fuente correspondiente al Test 4: The Longest Run of Ones in a Block

La función [nr,noaplicable, p_value,n,nu,M] = LR1(pat), permite el cálculo del test 4. Como argumento se le debe ingresar en la variable pat, el path o ruta del archivo en donde se guarda la secuencia binaria a testear. Esta secuencia binaria, se encuentra formada por unos y ceros, guardados en una sola columna y varias filas. La función LR1 calcula la probabilidad Pvalue y devuelve dicho valor en la variable p_value. Además, nos informa el estado de las variables nr y noaplicable. En el caso que la secuencia resulte random nr será igual a 0, en el caso que sea no random nr será igual a 1. En el caso que este test no se pueda aplicar por limitaciones en los requisitos como, por ejemplo, tamaño de la muestra, noaplicable será igual a 1. En caso contrario, es decir, si se puede aplicar, noaplicable será igual a 0. Además, la función retorna el valor de la longitud de la secuencia, n; el valor de nu y M. Todas estas variables, se implementaron para poder realizar un control en el procesamiento de series temporales experimentales.

```
function [ nr,noaplicable, p_value,n,nu,M ] = LR1( pat)

% NIST Test Suite en MATLAB
% Traducción del .c del NIST a MATLAB por Myriam Nonaka
% 22-05-2020
%
%#####
% Descripción:
% Test 4 Test for the Longest Run of Ones in a Block
%#####
%
% Variables utilizadas:
%

% n= La longitud del archivo de unos y ceros a testear.

% epsilon= Es la secuencia de datos del archivo a testear.
%
% M= La longitud de cada bloque.
%-----
% Formato de datos ingresados para epsilon debe ser de vector de unos
% y ceros en una sola columna con i filas. Serie binaria.

n=length(epsilon);
nr=-1;
p_value=-1;
M=-1;
nu=[0 0 0 0 0 0];
if n<128
```

```

disp('4-Longest Run of Ones. NO SE PUEDE APLICAR n<128');
noaplicable=1;
else
if n<6272
    K=4;
    M=8;
    V(1)=1; V(2)=2; V(3)=3; V(4)=4;
    pi(1) = 0.21484375;
    pi(2) = 0.3671875;
    pi(3) = 0.23046875;
    pi(4) = 0.1875;
else
if n<750000
    K=6;
    M=128;
    V(1)=4; V(2)=5; V(3) = 6; V(4) = 7; V(5) = 8; V(6) = 9;
    pi(1) = 0.1174035788;
    pi(2) = 0.242955959;
    pi(3) = 0.249363483;
    pi(4) = 0.17517706;
    pi(5) = 0.102701071;
    pi(6) = 0.112398847;
else
    K = 7;
    M = 10000;
    V(1) = 10; V(2) = 11; V(3) = 12; V(4) = 13; V(5) = 14; V(6) = 15; V(7) = 16;
    pi(1) = 0.0882;
    pi(2) = 0.2092;
    pi(3) = 0.2483;
    pi(4) = 0.1933;
    pi(5) = 0.1208;
    pi(6) = 0.0675;
    pi(7) = 0.0727;
end
end

N=floor(n/M);
for i=1:1:N
v_n_obs=0;
run=0;
for j=1:1:M
    if (epsilon( ((i-1)*M)+j)) ==1
        run=run+1;
        if run>v_n_obs
            v_n_obs=run;
        end
    else
        run=0;
    end
end
end
end

```

```

if v_n_obs<V(1)
    nu(1)=nu(1)+1;
end
for j=1:1:K
    if v_n_obs==V(j)
        nu(j)=nu(j)+1;
    end
end
if v_n_obs>V(K)
    nu(K)=nu(K)+1;
end
end

chi2=0.0;
for i=1:1:K
    chi2=chi2+ (((nu(i) - N * pi(i)) * (nu(i) - N * pi(i))) / (N * pi(i)));
end
p_value = gammainc(chi2/2,(K-1)/2,'upper');
if p_value<.01
    disp('4-Longest Run Of One. No es random. p_value<0.01' );
    p_value
    nr=1;
else
    disp('4-Longest Run Of One. Es random. p_value>0.01');
    p_value
    nr=0;
end
noaplicable=0;
end
fclose(fileID);
end

```

A.5. Código fuente correspondiente al Test 5: Binary Matrix Rank Test

La función, `[nr, noaplicable,p_value,n,N,F_32,F_31,chi_squared] = BM1(MM,QQ,pat)`, permite el cálculo del test 5. Como argumento se le debe ingresar en la variable `pat`, el path o ruta del archivo en donde se guarda la secuencia binaria a testear, el valor de `MM`, el número de filas de cada matriz; el valor de `QQ`, el número de columnas de cada matriz. Esta secuencia binaria, se encuentra formada por unos y ceros, guardados en una sola columna y varias filas. La función `BM1` calcula la probabilidad `Pvalue` y devuelve dicho valor en la variable `p_value`. Además, nos informa el estado de las variables `nr` y `noaplicable`. En el caso que la secuencia resulte random `nr` será igual a 0, en el caso que sea no random `nr` será igual a 1. En el caso que este test no se pueda aplicar por limitaciones en los requisitos como, por ejemplo, tamaño de la muestra, `noaplicable` será igual a 1. En caso contrario, es decir, si se puede aplicar, `noaplicable` será igual a 0. Además,

la función retorna el valor de la longitud de la secuencia, n; el valor de N, F_32, F_31 y chi_squared. Todas estas variables, se implementaron para poder realizar un control en el procesamiento de series temporales experimentales.

```
function [nr, noaplicable,p_value,n,N,F_32,F_31,chi_squared ] = BM1( MM,QQ,pat )

% NIST Test Suite en MATLAB
% Traducción del .c del NIST a MATLAB por Myriam Nonaka
% 23-05-2020
% BM
%#####
% Descripción:
% Test 5 Binary Matrix Rank Test
%#####
%
% Variables utilizadas:
%
% n= La longitud del archivo de unos y ceros a testear.
% epsilon= Es la secuencia de datos del archivo a testear.
%
% M= Número de filas de cada matriz.
% Q= Número de columnas de cada matriz.
%-----
% Formato de datos ingresados para epsilon debe ser de vector de unos
% y ceros en una sola columna con i filas. Serie binaria.

fileID=fopen(pat);
epsilon = fread(fileID,'int32'); %depende cómo se guarda el archivo a leer, puede ser
'double'
epsilon =epsilon';

n=length(epsilon);
M=MM; %M=32
Q=QQ; %Q=32
nr=-1;
p_value=-1;
N=-1;
F_32=-1;
F_31=-1;
chi_squared=-1;
if n>=(38*M*Q)
N=floor(n/(M*Q));
epsilon2=epsilon(1:(M*Q*N)); %se van los bits finales
e_matrix=vec2mat(epsilon2,Q);
r=32;
product=1;
for i=0:1:r-1
```

```

    product=product*((double(1)-(power(2,i-Q)))*(double(1)-(power(2,i-M)))/
(double(1)-power(2,i-r)));
end
    p_32 = (power(double(2),(r*(Q+M-r)-(M*Q)))) * product;
    r=31;
    product = 1;
for i=0:1:r-1
    product=product*(double(1)-(power(double(2),(i-Q))))*(double(1)-power(double(2),(i-
M)))/(double(1)-power(double(2),(i-r)));
end
    p_31 = power(double(2),(r*(Q+M-r)-(M*Q))) * product;
    p_30=double(1)-(p_32+p_31);
    F_32=0;
    F_31=0;
    F_30=0;
for k=0:1:N-1 %para cada matriz de 32x32
    matrix=e_matrix(1+M*k:M+M*k,:);
    R=rank(gf(matrix));
    if R==32
        F_32=F_32+1;
    end
    if R==31
        F_31=F_31+1;
    end
end
    F_30=double(N)-(F_32+F_31);
    chi_squared=((power((F_32 - N*p_32),2)/double(N*p_32)) + (power(F_31 -
N*p_31,2)/ double(N*p_31)) + (power(F_30 - N*p_30,2)/double(N*p_30)));
    arg1=-chi_squared/double(2);
    p_value=exp(arg1);
    disp('el pvalue del binary');
    p_value=exp(arg1)
n=length(epsilon);
N=floor(n/(M*Q));

    if p_value<.01
disp('5-Binary Matrix Rank Test. No es random. p_value<0.01');
    p_value
nr=1;
    else
disp('5-Binary Matrix Rank Test. Es random. p_value>0.01');
    p_value
nr=0;
    end
noaplicable=0;
    else
disp('5-Binary Matrix Rank Test. NO SE PUEDE APLICAR n<38*Q*M');
noaplicable=1;
    end
fclose(fileID);
end

```

A.6. Código fuente correspondiente al Test 6: Discrete Fourier Transform (Spectral) Test

La función, `[nr,noaplicable, p_value,n,N_1,N_o,d] = dFT1(pat)`, permite el cálculo del test 6. Como argumento se le debe ingresar en la variable `pat`, el path o ruta del archivo en donde se guarda la secuencia binaria a testear. Esta secuencia binaria, se encuentra formada por unos y ceros, guardados en una sola columna y varias filas. La función `dFT1` calcula la probabilidad `Pvalue` y devuelve dicho valor en la variable `p_value`. Además, nos informa el estado de las variables `nr` y `noaplicable`. En el caso que la secuencia resulte random `nr` será igual a 0, en el caso que sea no random `nr` será igual a 1. En el caso que este test no se pueda aplicar por limitaciones en los requisitos como, por ejemplo, tamaño de la muestra, `noaplicable` será igual a 1. En caso contrario, es decir, si se puede aplicar, `noaplicable` será igual a 0. Además, la función retorna el valor de la longitud de la secuencia, `n`; el valor de `N_1`, `N_o` y `d`. Todas estas variables, se implementaron para poder realizar un control en el procesamiento de series temporales experimentales.

```
function [nr,noaplicable, p_value,n,N_1,N_o,d ] = dFT1( pat )

% NIST Test Suite en MATLAB
% Traducción del .c del NIST a MATLAB por Myriam Nonaka
% 23-05-2020
%
%#####
% Descripción:
% Test 6 Discrete Fourier Transform (Spectral) Test
%#####
%
% Variables utilizadas:
%
% n= La longitud del archivo de unos y ceros a testear.
% epsilon= Es la secuencia de datos del archivo a testear.
%-----
% Formato de datos ingresados para epsilon debe ser de vector de unos
% y ceros en una sola columna con i filas. Serie binaria.

format long
fileID=fopen(pat);
epsilon = fread(fileID,'int32'); %depende cómo se guarda el archivo a leer, puede ser
'double'
fclose(fileID);
n=length(epsilon);
nr=-1;
p_value=-1;
N_1=-1;
N_o=-1;
```

```

d=-1;
if n>=1000
for i=1:1:n
X(i)=2*(epsilon(i))-1;
end
S=fft(X,n);
m(1,1)=sqrt(power(real(S(1,1)),2)+power(imag(S(1,1)),2));
for i=2:1:floor(n/2)
m(:,i)=sqrt(power(real(S(1,i)),2)+(power(imag(S(1,i)),2)));
end
count=0;
upperBound=sqrt(2.995732274*n);
for i=1:1:floor(n/2)
if m(:,i)<upperBound
count=count+1;
end
end
percentile=double(count/(n/2)*100);
N_1=double(count); %numeros de picos menores que h=sqrt(3*n)
N_o=double(0.95*n/2);
deno=sqrt((n*0.95*0.05)/4.0);
num=(N_1-N_o);
d=num/deno;
p_value=double(erfc(abs(d)/sqrt(2.0)))
if p_value<.01
disp('6-Discrete Fourier Transform Test. No es random. p_value<0.01' );
p_value
nr=1;
else
disp('6-Discrete Fourier Transform Test. Es random. p_value>0.01');
p_value
nr=0;
end

noaplicable=0;
else
disp('6-Discrete Fourier Transform Test. NO SE PUEDE APLICAR n<1000');
noaplicable=1;
end
end

```

A.7. Código fuente correspondiente al Test 7: Non-overlapping Template Matching Test

La función `[nr,noaplicable, n,p_value,m,chi2,M,N] = NOMT1(pat)`, permite el cálculo del Test 7. Como argumento se le debe ingresar en la variable `pat`, el path o ruta del archivo en donde se guarda la secuencia binaria a testear. Esta secuencia binaria, se encuentra formada por unos y ceros, guardados en una sola columna y varias filas. La función `NOMT1` calcula la probabilidad `Pvalue` y devuelve dicho valor en la variable `p_value`. Además, nos informa el estado de las

variables nr y noaplicable. En el caso que la secuencia resulte random nr será igual a 0, en el caso que sea no random nr será igual a 1. En el caso que este test no se pueda aplicar por limitaciones en los requisitos como, por ejemplo, tamaño de la muestra, noaplicable será igual a 1. En caso contrario, es decir, si se puede aplicar, noaplicable será igual a 0. Además, la función retorna el valor de la longitud de la secuencia, n; el valor de m, chi2, M y N. Todas estas variables, se implementaron para poder realizar un control en el procesamiento de series temporales experimentales.

```
function [nr,noaplicable, n,p_value,m,chi2,M,N ] = NOMT1( pat )
```

```
% NIST Test Suite en MATLAB
% Traducción del .c del NIST a MATLAB por Myriam Nonaka
% 23-05-2020
%#####
% Descripción:
% Test 7 Non-overlapping Template Matching Test
%#####
%
% Variables utilizadas:
%
% n= La longitud del archivo de unos y ceros a testear.
% epsilon= Es la secuencia de datos del archivo a testear.
% m= Longitud de cada template
% B=secuencia de m-bit template
% M=longitud de bits de cada substring de epsilon a ser testeado
%-----
% Formato de datos ingresados para epsilon debe ser de vector de unos
% y ceros en una sola columna con i filas. Serie binaria.

fileID=fopen(pat);
epsilon = fread(fileID,'int32'); %depende cómo se guarda el archivo a leer, puede ser
'double'
epsilon = epsilon';
n=length(epsilon);
N=8; %BLOCKS 8
M = floor(n/N);
K=5;
%secuencia
B=[0 0 0 0 0 0 0 1]; %TEMPLATE
m=length(B);
nr=-1;
p_value=-1;
chi2=-1;
if N<=100 && M>(n*0.01)
lambda= (M-m+1)/(2^m);
varWj=M*( (1/(2^m)) - (2*m-1)/(2^(2*m)) );
```

```

sum=0;
for k=1:1:K+1
    nu(k)=0;
end
for i=1:1:N
    W_obs=0;
    for j=1:1:M-m+1
        match=1;
        for k=1:1:m
            if B(k)~= epsilon((i-1)*M+(j-1)+k)
                match=0;
                break;
            end
        end
        if match==1
            W_obs=W_obs+1;
            j=j+m-1;
        end
    end
    Wj(i)=W_obs;
end
sum=0;
chi2=0.0;
for i=1:1:N
    chi2=chi2+ double(power((double(Wj(i))-lambda)/power(varWj,0.5),2));
end
p_value = gammainc(chi2/2,N/2,'upper')
if p_value<.01
disp('7-Non-overlapping Template Matching Test. No es random. p_value<0.01' );
p_value
nr=1;
else

disp('7-Non-overlapping Template Matching Test. Es random. p_value>0.01');
p_value
nr=0;
end
noaplicable=0;
else
disp('7-Non overlapping test. NO SE PUEDE APLICAR N<100 . M<=0.01*N');
noaplicable=1;
end
fclose(fileID);
end

```

A.8. Código fuente correspondiente al Test 8: Overlapping Template Matching Test

La función, [nr,noaplicable, p_value,n,nu,B,chi2,m] = ot1(pat), permite el cálculo del test 8. Como argumento se le debe ingresar en la variable pat, el path o ruta del archivo en donde se

guarda la secuencia binaria a testear. Esta secuencia binaria, se encuentra formada por unos y ceros, guardados en una sola columna y varias filas. La función `ot1` calcula la probabilidad `Pvalue` y devuelve dicho valor en la variable `p_value`. Además, nos informa el estado de las variables `nr` y `noaplicable`. En el caso que la secuencia resulte random `nr` será igual a 0, en el caso que sea no random `nr` será igual a 1. En el caso que este test no se pueda aplicar por limitaciones en los requisitos como, por ejemplo, tamaño de la muestra, `noaplicable` será igual a 1. En caso contrario, es decir, si se puede aplicar, `noaplicable` será igual a 0. Además, la función retorna el valor de la longitud de la secuencia, `n`; el valor de `nu`, `B`, `chi2` y `m`. Todas estas variables, se implementaron para poder realizar un control en el procesamiento de series temporales experimentales.

```
function [nr,noaplicable, p_value,n,nu,B,chi2,m ] = ot1( pat )

% NIST Test Suite en MATLAB
% Traducción del .c del NIST a MATLAB por Myriam Nonaka
% 23-05-2020
%#####
% Descripción:
% Test 8 Overlapping Template Matching Test
%#####
%
% Variables utilizadas:
%
% n= La longitud del archivo de unos y ceros a testear.
% epsilon= Es la secuencia de datos del archivo a testear.
% m= Longitud de cada template

% B=secuencia de m-bit template
% M=longitud de bits de cada substring de epsilon a ser testeado
%-----
% Formato de datos ingresados para epsilon debe ser de vector de unos
% y ceros en una sola columna con i filas. Serie binaria.

fileID=fopen(pat);
epsilon = fread(fileID,'int32'); %depende cómo se guarda el archivo a leer, puede ser
'double'
fclose(fileID);
n=length(epsilon);
nr=-1;
p_value=-1;
nu=-1;
B=-1;
chi2=-1;
m=-1;
if n>=1000000
```

```

M=1032;
K=5;
nu=[0 0 0 0 0];
N=floor(n/M);
B=[1 1 1 1 1 1 1 1];
m=length(B);
lambda=double((M-m+1)/power(2,m));
eta=double(lambda/2);
sum=0.0;
for i=0:1:5-1 %calculo de probabilidades
u=i;
if u==0
pr=exp(-eta);
else
sum1=0;
for l=1:1:u
sum1=sum1+exp(-eta-(u)*log(2)+(l)*log(eta)-gammaln((l+1))+gammaln(u)-
gammaln(l)-gammaln(u-l+1));
end
pr=sum1;
end
pi(i+1)= pr;
sum=sum+pi(i+1);
end
pi(5+1)=1-sum;
for i=1:1:N
W_obs=0;
for j=1:1:M-m+1
match=1;
for k=1:1:m
if B(k)~=epsilon((i-1)*M+j+(k-1))
match=0;
end
end
if match==1
W_obs=W_obs+1;
end
end
if W_obs<=4
nu(W_obs+1)=nu(W_obs+1)+1;
else
nu(6)=nu(6)+1;
end
end
sum=0;
chi2=0.0;
for i=1:1:K+1
chi2=chi2+( double(power((double(nu(i))-double(N*pi(i))),2)) / double(N*pi(i)) );
sum=sum+nu(i);
end

```

```

p_value=gammainc(double(chi2/2),double(K/2),'upper');
    if p_value<.01
disp('8-Overlapping Template Matching Test. No es random. p_value<0.01' );
p_value
nr=1;
    else
disp('8-Overlapping Template Matching Test. Es random. p_value>0.01');
p_value
nr=0;
    end
noaplicable=0;
    else
disp('8-Overlapping Test.. NO SE PUEDE APLICAR n<1000000');
noaplicable=1;
    end
end
end

```

A.9. Código fuente correspondiente al test 9: Maurer's Universal Statistical Test

La función, [nr, noaplicable, p_value,n,c,K,sum,evL,Q] = MU1(LL,pat), permite el cálculo del test 9. Como argumento se le debe ingresar en la variable pat, el path o ruta del archivo en donde se guarda la secuencia binaria a testear y el LL, longitud de cada bloque. Esta secuencia binaria, se encuentra formada por unos y ceros, guardados en una sola columna y varias filas. La función MU1 calcula la probabilidad Pvalue y devuelve dicho valor en la variable p_value. Además, nos informa el estado de las variables nr y noaplicable. En el caso que la secuencia resulte random nr será igual a 0, en el caso que sea no random nr será igual a 1. En el caso que este test no se pueda aplicar por limitaciones en los requisitos como, por ejemplo, tamaño de la muestra, noaplicable será igual a 1. En caso contrario, es decir, si se puede aplicar, noaplicable será igual a 0. Además, la función retorna el valor de la longitud de la secuencia, n; el valor de c, K, sum,evL y Q. Todas estas variables, se implementaron para poder realizar un control en el procesamiento de series temporales experimentales.

```

function [nr, noaplicable, p_value,n,c,K,sum,evL,Q ] = MU1( LL,pat )

% NIST Test Suite en MATLAB
% Traducción del .c del NIST a MATLAB por Myriam Nonaka
% 23-05-2020
%#####
% Descripción:
% Test 9 Test Maurer´s Universal Statistical Test
%#####
%
% Variables utilizadas:
% L= longitud de cada bloque
% Q= El número de bloques de la secuencia de inicialización

```

```
% n= La longitud del archivo de unos y ceros a testear.
% epsilon= Es la secuencia de datos del archivo a testear.
%
%-----
% Formato de datos ingresados para epsilon debe ser de vector de unos
% y ceros en una sola columna con i filas. Serie binaria.

expected_value = [ 0 0 0 0 0 5.2177052 6.1962507 7.1836656 8.1764248 9.1723243
10.170032 11.168765 12.168070 13.167693 14.167488 15.167379];
variance= [ 0 0 0 0 0 2.954 3.125 3.238 3.311 3.356 3.384 3.401 3.410 3.416 3.419 3.421
];

fileID=fopen(pat);
epsilon = fread(fileID,'int32'); %depende cómo se guarda el archivo a leer, puede ser
'double'
epsilon = epsilon';

nr=-1;
p_value=-1;
evL=-1;
Q=-1;
L=LL;
noaplicable=-1;
n=-1;
c=-1;
K=-1;
sum=-1;
evL=-1;
Q=-1;

n=length(epsilon);

if n>=387840
    L=6;
end

if n>=904960
    L=7;
end

if n>=2068480
    L=8;
end

if n>=4654080
    L=9;
end

if n>=10342400
    L=10;
```

```

end

if n>=22753280
    L=11;
end

if n>=49643520
    L=12;
end

if n>=107560960
    L=13;
end

if n>=231669760
    L=14;
end

if n>=496435200
    L=15;
end

if n>=1059061760
    L=16;
end
Q=10*double(power(2,L));
K=double(floor(n/L)-double(Q));
p=double(power(2,L));
c=0.7-0.8/double(L)+((4+32/double(L))*(K^(-3/double(L))))/15;
sigma=c*sqrt(variance(L)/double(K));

sqrt2=sqrt(2);
sum=0.0;
if (n>=(Q+K)*L)&& (L>=6)&&(L<=16)
    T=zeros(1,p);
    for i=1:1:Q
        decRep=0;
        for j=0:1:L-1
            decRep=decRep+epsilon(((i-1)*L)+1+j)*int32(power(2,L-1-j));
        end
        T(decRep+1)=i;
    end
    for i=Q+1:1:Q+K
        decRep=0;
        for j=0:1:L-1
            decRep=decRep+epsilon((i-1)*L+1+j)*int32(power(2,L-1-j));
        end
        sum=sum+(log(i-T(decRep+1))/log(2));
        T(decRep+1)=i;
    end
end
phi=double(sum/double(K));

```

```

evL=expected_value(L);
arg=abs(phi-evL)/(sqrt2*sigma);
p_value=erfc(arg);
    if p_value<.01
disp('9-Maurer's Universal Statistical Test.. No es random. p_value<0.01' );
p_value
nr=1;
else
disp('9-Maurer's Universal Statistical Test.. Es random. p_value>0.01');
p_value
nr=0;
end
noaplicable=0;
else
disp('9-Maurer's Universal Statistical Test. NO SE PUEDE APLICAR n<(Q+K)*L');
noaplicable=1;
end
fclose(fileID);
end

```

A.10. Código fuente correspondiente al test 10: Linear Complexity Test

La función, [nr,noaplicable, p_value,n] = lc1(pat), permite el cálculo del test 10. Como argumento se le debe ingresar en la variable pat, el path o ruta del archivo en donde se guarda la secuencia binaria a testear. Esta secuencia binaria, se encuentra formada por unos y ceros, guardados en una sola columna y varias filas. La función lc1 calcula la probabilidad Pvalue y devuelve dicho valor en la variable p_value. Además, nos informa el estado de las variables nr y noaplicable. En el caso que la secuencia resulte random nr será igual a 0, en el caso que sea no random nr será igual a 1. En el caso que este test no se pueda aplicar por limitaciones en los requisitos como, por ejemplo, tamaño de la muestra, noaplicable será igual a 1. En caso contrario, es decir, si se puede aplicar, noaplicable será igual a 0. Además, la función retorna el valor de la longitud de la secuencia, n. Todas estas variables, se implementaron para poder realizar un control en el procesamiento de series temporales experimentales.

```

function [nr,noaplicable, p_value,n] = lc1( pat )

% NIST Test Suite en MATLAB
% Traducción del .c del NIST a MATLAB por Myriam Nonaka
% 23-05-2020
%#####
% Descripción:
% Test 10 Linear Complexity Test
%#####
%
```

```

% Variables utilizadas:
%
% n= La longitud del archivo de unos y ceros a testear.
% epsilon= Es la secuencia de datos del archivo a testear.
%
% M= Longitud de bits de un bloque.
% K= Número de grados de libertad.
%-----
% Formato de datos ingresados para epsilon debe ser de vector de unos
% y ceros en una sola columna con i filas. Serie binaria.

nr=-1;
p_value=-1;
M=500;
fileID=fopen(pat);
epsilon = fread(fileID,'int32'); %depende cómo se guarda el archivo a leer, puede ser
'double'
epsilon = epsilon';
n=length(epsilon);
if n>=1000000
K=6;
pi= [ 0.01047 0.03125 0.12500 0.50000 0.25000 0.06250 0.020833 ];
N = floor(n/M);
for i=1:1:K+1
    nu(1,i)=0.00;
end
for ii=1:1:N
for i=1:1:M+1

    B_(1,i)=0.00;
    C(1,i)=0.00;
    T(1,i)=0.00;
    P(1,i)=0.00;
end

    L=0;
    m=-1;
    d=0;
    C(1)=1;
    B_(1)=1;

    N_=0;

while N_ < M
    d= epsilon(1,(ii-1)*M+N_+1);
    for i=2:1:L+1
        d=d+C(1,i)*epsilon(1,(ii-1)*M+N_-(i)+2);
    end
    d=rem(d,2);
    if d==1
        for i=1:1:M

```

```

T(1,i)=C(1,i);
P(1,i)=0;
end
for j=1:1:M
    if B_(j)==1
        P(j+N_-m)=1;
    end
end
for i=1:1:M
    C(1,i)=rem(C(1,i)+P(1,i),2);
end
if L<=N_/2
    L=N_+1-L;
    m=N_;
    for i=1:1:M
        B_(1,i)=T(1,i);
    end
end
end
N_=N_+1;
end
parity=rem(M+1,2);
if parity==0
    sign=-1;
else
    sign=1;
end
mean= M/2.0 + (9.0+sign)/36.0-(1.0/(2^M))*(M/3.0+2.0/9.0);
parity=rem(M,2);
if parity==0
    sign=1;
else
    sign=-1;
end
T_=sign*(L-mean)+2.0/9.0;
if T_<=-2.5
    nu(1,1)=nu(1,1)+1;
else
    if T_>-2.5 && T_<=-1.5
        nu(1,2)=nu(1,2)+1;
    else
        if T_>-1.5 && T_<=-0.5
            nu(1,3)=nu(1,3)+1;
        else
            if T_>-0.5&& T_<0.5
                nu(1,4)=nu(1,4)+1;
            else
                if T_>0.5 && T_<=1.5
                    nu(1,5)=nu(1,5)+1;
                else
                    if T_>1.5&&T_<=2.5

```

```

        nu(1,6)=nu(1,6)+1;
    else
        nu(1,7)=nu(1,7)+1;
    end
end
end
end
end
end
end
end

chi2=0.00;
for i=1:1:K+1
chi2=chi2+power((nu(1,i)-N*pi(1,i)),2)/(N*pi(1,i));
end
p_value=gammainc(chi2/2.0,K/2.0,'upper')
if p_value<.01
disp('10-Linear Complexity test. No es random. p_value<0.01' );
p_value
nr=1;
else
disp('10-Linear Complexity test. Es random. p_value>0.01');
p_value
nr=0;
noaplicable=0;
end
else
disp('10-Linear Complexity Test. NO SE PUEDE APLICAR n<1000000');
noaplicable=1;
end
fclose(fileID);
end

```

A.11. Código fuente correspondiente al test 11: Serial Test

La función, `[nr,noaplicable, p_value1,p_value2,n,psim0,psim1,psim2] = ST1(mm,pat)`, permite el cálculo del test 11. Como argumento se le debe ingresar en la variable `pat`, el path o ruta del archivo en donde se guarda la secuencia binaria a testear y el valor `mm`. Esta secuencia binaria, se encuentra formada por unos y ceros, guardados en una sola columna y varias filas. La función `ST1` calcula la probabilidad `Pvalue1` y `Pvalue2` y devuelve dicho valor en la variable `p_value1` y `p_value2`. Además, nos informa el estado de las variables `nr` y `noaplicable`. En el caso que la secuencia resulte random `nr` será igual a 0, en el caso que sea no random `nr` será igual a 1. En el caso que este test no se pueda aplicar por limitaciones en los requisitos como, por ejemplo, tamaño de la muestra, `noaplicable` será igual a 1. En caso contrario, es decir, si se puede aplicar, `noaplicable`

será igual a 0. Además, la función retorna el valor de la longitud de la secuencia, n; psim0, psim1 y psim2. Todas estas variables, se implementaron para poder realizar un control en el procesamiento de series temporales experimentales. Además, se agrega la función auxiliar del ST1, la función psi2().

Function [nr,noaplicable, p_value1,p_value2,n,psim0,psim1,psim2] = ST1(mm,pat)

```

% NIST Test Suite en MATLAB
% Traducción del .c del NIST a MATLAB por Myriam Nonaka
% 23-05-2020
%#####
% Descripción:
% Test 11 Serial Test
%#####
%
% Variables utilizadas:
%
% n= La longitud del archivo de unos y ceros a testear.
% epsilon= Es la secuencia de datos del archivo a testear.
%
% M= Longitud de bits de un bloque.
% K= Número de grados de libertad.
%-----
% Formato de datos ingresados para epsilon debe ser de vector de unos
% y ceros en una sola columna con i filas.

fileID=fopen(pat);
epsilon = fread(fileID,'int32'); %depende cómo se guarda el archivo a leer, puede ser
'double'
epsilon = epsilon';
n=length(epsilon);
m=mm;
nr=-1;
p_value=-1;
p_value1=-1;
p_value2=-1;
psim0=-1;
psim1=-1;
psim2=-1;
nr=-1;

if m<log2(n)-2
numOfBlocks=n;
psim0=psi2(m,n,epsilon);
psim1=psi2(m-1,n,epsilon);
psim2=psi2(m-2,n,epsilon);
del1=psim0-psim1;

```

```

del2=psim0-2.0*psim1+psim2;
p_value1=gammainc(del1/2.0,(2^(m-1))/2,'upper');
p_value2=gammainc(del2/2.0,(2^(m-2))/2,'upper'); %agarra el valor más grande de
probabilidad
noaplicable=0;
if ( p_value1<.01 && p_value2<.01 )
disp('11-Serial Test. No es random. p_value<0.01' );
p_value1
p_value2
nr=1;
else
disp('11-Serial Test. Es random. ');
p_value1
p_value2
nr=0;
end
else
disp('11-Serial Test.NO SE PUEDE APLICAR m>=log2(n)-2');
noaplicable=1;
end
fclose(fileID);
end

```

```
function [ sum ] = psi2(m,n,epsilon)
```

```

%#####
% Descripción:
% Funcion psi2() para Test 11 Serial Test
%#####

```

```

if m-1==0
%queda igual
else
epsilonp=[epsilon,epsilon(1:m-1)];
epsilon=epsilonp;
end
numOfBlocks=n;
powLen=(2^(m+1))-1;
P=zeros(1,powLen);
for i=1:1:numOfBlocks
k=1;
for j=1:1:m
if epsilon(rem(i+j,n)+1)==0
k=k*2;
else
if epsilon(rem(i+j,n)+1)==1
k=2*k+1;
end
end
end
end

```

```

P(1,k-1)=P(1,k-1)+1;
end
sum=0.0;
for i=2^(m)-1:1:2^(m+1)-1
    sum=sum+(P(1,i)^2);
end
sum=double(sum*(2^m)/double(n))-double(n);
end

```

A.12. Código fuente correspondiente al Test 12: Approximate Entropy Test

La función, [nr, noaplicable, n,p_value,chi_squared,apen] = ae1(mm,pat), permite el cálculo del test 12. Como argumento se le debe ingresar en la variable pat, el path o ruta del archivo en donde se guarda la secuencia binaria a testear y el valor mm, valor inicial del tamaño del bloque m a variar. Esta secuencia binaria, se encuentra formada por unos y ceros, guardados en una sola columna y varias filas. La función ae1 calcula la probabilidad Pvalue y devuelve dicho valor en la variable p_value. Además, nos informa el estado de las variables nr y noaplicable. En el caso que la secuencia resulte random nr será igual a 0, en el caso que sea no random nr será igual a 1. En el caso que este test no se pueda aplicar por limitaciones en los requisitos como, por ejemplo, tamaño de la muestra, noaplicable será igual a 1. En caso contrario, es decir, si se puede aplicar, noaplicable será igual a 0. Además, la función retorna el valor de la longitud de la secuencia, n; chi_squared; apen. Todas estas variables, se implementaron para poder realizar un control en el procesamiento de series temporales experimentales.

```

function [nr, noaplicable, n,p_value,chi_squared,apen ] = ae1( mm,pat )

% NIST Test Suite en MATLAB
% Traducción del .c del NIST a MATLAB por Myriam Nonaka
% 23-05-2020
%#####
% Descripción:
% Test 12 Approximate Entropy Test
%#####
%
% Variables utilizadas:
%
% n= La longitud del archivo de unos y ceros a testear.
% epsilon= Es la secuencia de datos del archivo a testear.
%
%-----
% Formato de datos ingresados para epsilon debe ser de vector de unos
% y ceros en una sola columna con i filas. Serie binaria.

fileID=fopen(pat);

```

```

epsilon = fread(fileID,'int32'); %depende cómo se guarda el archivo a leer, puede ser
'double'
epsilon = epsilon';
fclose(fileID);
n=length(epsilon);
seqLength=n;

m=mm;
r=0;
nr=-1;
p_value=-1;
chi_squared=-1;
apen=-1;

if m <(log2(n)-5)
for blockSize=m : 1:m+1

epsilonp=[epsilon, epsilon(1:blockSize-1)];
epsilon1=epsilonp;

    if blockSize==0

        ApEn(1)=0;
        r=r+1;
    else
        numOfBlocks=double(seqLength);
        powLen=(2^(blockSize+1))-1;

        P=zeros(1,powLen);
        for i=1:1:numOfBlocks
            k=1;
            for j=1:1:blockSize
                %bitwise shift a<<b es a=a<<b
                %Bit shift left arithmetic
                k=bitsll(k,1);
                if (epsilon1(rem((i+j),seqLength)+1))==1
                    k=k+1;
                end
            end
            P(k-1)=P(k-1)+1;
        end

        sum=0.0;
        index=(2^blockSize)-1;
        for i=1:1:(2^blockSize)
            if P(index)>0
                sum=double(sum+P(index)*log(P(index)/numOfBlocks));
            end
            index=index+1;
        end
    end
end

```

```

sum=sum/numOfBlocks;
ApEn(r+1)=sum;
r=r+1;
end
end
apen=double(ApEn(1)-ApEn(2));
chi_squared=double(2*seqLength*(log(2)-apen));
p_value=gammainc(chi_squared/2,(2^(m-1)),'upper');
noaplicable=0;

if p_value<.01
disp('12-Overlapping Test. No es random. p_value<0.01' );
p_value
nr=1;
else
disp('12-Overlapping Test. Es random. p_value>0.01');
p_value
nr=0;
end

else

disp('12-Overlapping Test.NO SE PUEDE APLICAR m>=log2(n)-5');
noaplicable=1;
end
end

```

A.13. Código fuente correspondiente al Test 13: Cumulative Sums Test

La función, `[nrF,nrB, noaplicable, p_valueF,p_valueB,zrev,z,n] = cu1(pat)`, permite el cálculo del Test 13. Como argumento se le debe ingresar en la variable `pat`, el path o ruta del archivo en donde se guarda la secuencia binaria a testear. Esta secuencia binaria, se encuentra formada por unos y ceros, guardados en una sola columna y varias filas. La función `cu1` calcula la probabilidad `Pvalue` para el Forward y el Reverse y devuelve dicho valor en la variable `p_valueF` y el `p_valueB`. Además, nos informa el estado de las variables `nr` y `noaplicable`. En el caso que la secuencia resulte random `nr` será igual a 0, en el caso que sea no random `nr` será igual a 1 (para el caso Forward se utiliza el `nrF` y para el reverse, el `nrB`). En el caso que este test no se pueda aplicar por limitaciones en los requisitos como, por ejemplo, tamaño de la muestra, `noaplicable` será igual a 1. En caso contrario, es decir, si se puede aplicar, `noaplicable` será igual a 0. Además, la función retorna el valor de la longitud de la secuencia, `n`; `zrev` y `z`. Todas estas variables, se implementaron para poder realizar un control en el procesamiento de series temporales experimentales.

```
function [nrF,nrB, noaplicable, p_valueF,p_valueB,zrev,z,n ] = cu1( pat )
```

```
% NIST Test Suite en MATLAB
```

```

% Traducción del .c del NIST a MATLAB por Myriam Nonaka
% 15-04-2019
%#####
% Descripción:
% Test 13 cummulative sums Test
%#####
%
% Variables utilizadas:
%
% n= La longitud del archivo de unos y ceros a testear.
% epsilon= Es la secuencia de datos del archivo a testear.
%
%-----
% Formato de datos ingresados para epsilon debe ser de vector de unos
% y ceros en una sola columna con i filas. Serie binaria

```

```

fileID=fopen(pat);
epsilon = fread(fileID,'int32'); %depende cómo se guarda el archivo a leer, puede ser
'double'

```

```

n=length(epsilon);
nr=-1;
p_value=-1;
p_valueF=-1;
p_valueB=-1;
zrev=-1;
z=-1;
nrF=-1;
nrB=-1;
if n>=100
S=0;
sup=0;
inf=0;
for k=1:1:n
    if epsilon(k)==1
        S=S+1;
    else
        S=S-1;
    end
    if S>sup
        sup=sup+1;
    end
    if S<inf
        inf=inf-1;
    end
    if (sup>-inf)==1
        z=sup;
    else
        z=-inf;
    end
end

```

```

if (sup-S>S-inf)==1
    zrev=sup-S;
else
    zrev=S-inf;
end
%forward
end
sum1=0;
for k=floor(((n/z)+1)/4):1:floor(((n/z)-1)/4)
    sum1=sum1-normcdf(((4*k-1)*z)/sqrt(n))+normcdf(((4*k+1)*z)/sqrt(n));
end
sum2=0;
for k=floor(((n/z)-3)/4):1:floor(((n/z)-1)/4)
    sum2=sum2-normcdf(((4*k+1)*z)/sqrt(n))+normcdf(((4*k+3)*z)/sqrt(n));
end
p_valueF=1-sum1+sum2;
%backwards/reverse
sum1=0;
for k=floor(((n/zrev)+1)/4):1:floor(((n/zrev)-1)/4)
    sum1=sum1+normcdf(((4*k+1)*zrev)/sqrt(n))-normcdf(((4*k-1)*zrev)/sqrt(n));
end
sum2=0;
for k=floor(((n/zrev)-3)/4):1:floor(((n/zrev)-1)/4)
    sum2=sum2+normcdf(((4*k+3)*zrev)/sqrt(n))-normcdf(((4*k+1)*zrev)/sqrt(n));
end
p_valueB = 1- sum1 + sum2;
    if p_valueB<.01
disp('13-Cumulative Sums (Reverse). No es random. p_value<0.01' );
p_valueB
nrB=1;
else
disp('13-Cumulative Sums (Reverse). Es random. p_value>0.01');
p_valueB
nrB=0;
end
if p_valueF<.01
disp('13-Cumulative Sums (Foward). No es random. p_value<0.01' );
p_valueF
nrF=1;
else
disp('13-Cumulative Sums (Foward). Es random. p_value>0.01');
p_valueF
nrF=0;
end
noaplicable=0;
else
disp('13-Cumulative Sums.NO SE PUEDE APLICAR n<100');
noaplicable=1;
end
fclose(fileID);

```

`end`

14. Código fuente correspondiente al Test 14: Random Excursions Test

La función, `[nr,noaplicable, p_valueM,J,valores,n] = RE1(pat)`, permite el cálculo del Test 14. Como argumento se le debe ingresar en la variable `pat`, el path o ruta del archivo en donde se guarda la secuencia binaria a testear. Esta secuencia binaria, se encuentra formada por unos y ceros, guardados en una sola columna y varias filas. La función `RE1` calcula la probabilidad `Pvalue` y devuelve dicho valor en la variable `p_value`. Además, nos informa el estado de las variables `nr` y `noaplicable`. En el caso que la secuencia resulte random `nr` será igual a 0, en el caso que sea no random `nr` será igual a 1. En el caso que este test no se pueda aplicar por limitaciones en los requisitos como, por ejemplo, tamaño de la muestra, `noaplicable` será igual a 1. En caso contrario, es decir, si se puede aplicar, `noaplicable` será igual a 0. Además, la función retorna el valor de la longitud de la secuencia, `n`; `J` y `valores`. Todas estas variables, se implementaron para poder realizar un control en el procesamiento de series temporales experimentales.

```
function [nr,noaplicable, p_valueM,J,valores,n ] = RE1( pat )
% NIST Test Suite en MATLAB
% Traducción del .c del NIST a MATLAB por Myriam Nonaka
% 23-05-2020
%#####
% Descripción:
% Test 14 Random Excursions Test
%#####
%
% Variables utilizadas:
%
% n= La longitud del archivo de unos y ceros a testear.
% epsilon= Es la secuencia de datos del archivo a testear.
%
%-----
% Formato de datos ingresados para epsilon debe ser de vector de unos
% y ceros en una sola columna con i filas. Serie binaria.

fileID=fopen(pat);
epsilon = fread(fileID,'int32'); %depende cómo se guarda el archivo a leer, puede ser
'double'
epsilon = epsilon';
n=length(epsilon);
nr=-1;
p_value=-1;
p_valueM=-1;
J=-1;
valores=-1;
```

```

noaplicable=-1;
if n>=1000000
stateX = [ -4 -3 -2 -1 1 2 3 4 ];
counter = [ 0 0 0 0 0 0 0 0 ];

pi = [0.5000000000 0.2500000000 0.1250000000 0.0625000000 0.0312500000
0.0312500000 ;
      0.7500000000 0.0625000000 0.0468750000 0.0351562500 0.02636718750
0.0791015625 ;
      0.8333333333 0.02777777778 0.02314814815 0.01929012346 0.01607510288
0.0803755143 ;
      0.8750000000 0.0156250000 0.01367187500 0.01196289063 0.01046752930
0.0732727051 ];
S_k=zeros(1,n);
cycle=zeros(1,max(1000,round(n/100)));

J=0;
%Determina ciclos
S_k(1,1)=2*(epsilon(1))-1;
for i=2:1:n
S_k(1,i)=S_k(1,i-1)+2*epsilon(i)-1;
if S_k(1,i)==0
    J=J+1;
    if J>max(1000,n/100)
        disp('ERROR IN FUNCTION randomExcursions: EXCEEDING THE MAX
NUMBER OF CYCLES EXPECTED\n');
        return;
    end
    cycle(J)=i;
end
end
if S_k(1,n)~=0
    J=J+1;
    cycle(J)=n+1;
else
    cycle(J)=n;
end
if S_k(n)==0
S_kp=S_k(1:n);
else

S_kp=[S_k(1:n),0];
end
constraint=max(0.005*(n^0.5),500);
if J<constraint
disp('El test no es aplicable ');
p_valueM=-1;
valores=-1;
else
cycleStart=1;
cycleStop=cycle(1);

```

```

nu=zeros(6,8);
%Inicializar la matriz
for j=1:1:J
counter=zeros(1,8);
for i=cycleStart:1:cycleStop
if (S_kp(i)>=1 && S_kp(i)<=4)||((S_kp(i)>=-4&&S_kp(i)<=-1)
if S_kp(i)<0
    b=4;
else
    b=3;
end
counter(1,S_kp(i)+b+1)=counter(1,S_kp(i)+b+1)+1;
end
end
cycleStart= cycle(j)+1;

if j<=J
    cycleStop=cycle(j+1);
end
for i=1:1:8
if counter(1,i)>=0 && counter(1,i)<=4
    nu(counter(1,i)+1,i)= nu(counter(1,i)+1,i)+1;
else
    if counter(1,i)>=5
        nu(5+1,i)=nu(5+1,i)+1;
    end
end
end
end
valores=zeros(8,2);%donde se guardan
for i=1:1:8
x=stateX(i)
valores(i,1)=x;
sum=0.0;
for k=1:1:6
    sum=double(sum+(nu(k,i)-J*pi((abs(x)),k))^2/(J*pi(abs(x),k)));
end
sum
P_value=gammainc(sum/2,2.5,'upper')
valores(i,2)=P_value;
end
%busca el minimo de los x states y saca si es random o no
[M,I] = min(valores(:,2));
p_valueM=M;
state=I;
if p_valueM<.01
disp('14-Random Excursions Test. No es random. p_value<0.01' );
p_valueM
nr=1;
else
disp('14-Random Excursions Test. Es random. p_value>0.01');

```

```
p_valueM
nr=0;
end
noaplicable=0;
end
else
disp('14-Random Excursions Test.NO SE PUEDE APLICAR n<1000000');
noaplicable=1;
end
fclose(fileID);
end
```

A.15. Código fuente correspondiente al Test 15: Random Excursions Variant Test

La función, [nr,noaplicable, p_valueM,J,n,valores] = REVT1(pat), permite el cálculo del test 15. Como argumento se le debe ingresar en la variable pat, el path o ruta del archivo en donde se guarda la secuencia binaria a testear. Esta secuencia binaria, se encuentra formada por unos y ceros, guardados en una sola columna y varias filas. La función REVT1 calcula la probabilidad Pvalue y devuelve dicho valor en la variable p_value. Además, nos informa el estado de las variables nr y noaplicable. En el caso que la secuencia resulte random nr será igual a 0, en el caso que sea no random nr será igual a 1. En el caso que este test no se pueda aplicar por limitaciones en los requisitos como, por ejemplo, tamaño de la muestra, noaplicable será igual a 1. En caso contrario, es decir, si se puede aplicar, noaplicable será igual a 0. Además, la función retorna el valor de la longitud de la secuencia, n; J y valores. Todas estas variables, se implementaron para poder realizar un control en el procesamiento de series temporales experimentales.

```
function [nr,noaplicable, p_valueM,J,n,valores ] = REVT1( pat )

% NIST Test Suite en MATLAB
% Traducción del .c del NIST a MATLAB por Myriam Nonaka
% 23-05-2020
%#####
% Descripción:
% Test 15 Random Excursions Variant Test
%#####
%
% Variables utilizadas:
%
% n= La longitud del archivo de unos y ceros a testear.
% epsilon= Es la secuencia de datos del archivo a testear.
%
%-----
% Formato de datos ingresados para epsilon debe ser de vector de unos
% y ceros en una sola columna con i filas. Serie binaria.
```

```

fileID=fopen(pat);
epsilon = fread(fileID,'int32'); %depende cómo se guarda el archivo a leer, puede ser
'double'

epsilon = epsilon';
n=length(epsilon);
nr=-1;
p_value=-1;
p_valueM=-1;

J=-1;

valores=-1;
if n>=1000000
stateX=[ -9 -8 -7 -6 -5 -4 -3 -2 -1 1 2 3 4 5 6 7 8 9 ];
S_k=zeros(1,n);
J=0;
S_k(1,1)=2*(epsilon(1))-1;
for i=2:1:n
S_k(1,i)=S_k(1,i-1)+2*epsilon(i)-1;
if S_k(1,i)==0
    J=J+1;
end
end
if S_k(1,n)~=0
J=J+1;
end
if S_k(n)==0
S_kp=S_k(1:n);
else
S_kp=[S_k(1:n),0];
end
constraint=0; %para que funcione el ejemplo del pdf
if J<constraint
%mostrar resultados
disp('La muestra tiene que ser mayor a 500')
else
valores=zeros(18,2);
for p=1:1:18
x=stateX(p)
valores(p,1)=x;
count=0;
for i=1:1:n
if S_kp(1,i)==x
    count=count+1;
end
end
p_value=erfc(abs(count-J)/(sqrt(2*J*(4*abs(x)-2))))
valores(p,2)=p_value;
end
countR=0;

```

```

countNR=0;
for z=1:1:18
    p_value= valores(z,2);
    if p_value<.01
disp('6-Random Excursions Variant Test. No es random. p_value<0.01' );
z
p_value
countNR=countNR+1;
else
disp('6-Random Excursions Variant Test. Es random. p_value>0.01');

z
p_value
countR=countR+1;
    end
end
if countR==18
    disp('Random Excursions Variant Test. Es random para 18 estados. p_value>0.01');
    nr=0;
else
    if countR==0
        disp('Random Excursions Variant Test. Es no random para 18 estados.
p_value<0.01');
        nr=1;
    end
    if countR>0 && countR<0
        disp('Random Excursions Variant Test. no se puede definir si es random/no random
para 18 estados. ');
        nr=2;
    end
end
end
[M,I] = max(valores(:,2)); %el estado desfavorable
p_valueM=M;
state=I;
noaplicable=0;
else
disp('15-Random Excursions Variant Test. .NO SE PUEDE APLICAR n<1000000');
noaplicable=1;
end
fclose(fileID);
end

```

APÉNDICE B

CÓDIGOS DESARROLLADOS EN MATLAB SCRIPT DE LABVIEW PARA LOS PRM1Z8

En este apéndice, se mostrarán las capturas en pantalla de los códigos desarrollados para MATLAB script node de LabVIEW, correspondientes a la programación de las plataformas motorizadas PRM1Z8 correspondientes a las Estaciones A (Alice) y B (Bob), descritos en el capítulo 4.

B.1 Codificación conexión del rotador con la PC: este código permite realizar la conexión de ambas plataformas motorizadas correspondientes a las estaciones A y B. Para ello, recibe los números de serie de ambas plataformas motorizadas mediante la variable "HWSerialNum A" para el caso de la estación A y "HWSerialNum B" para el caso de la estación B. En la figura B.1, se muestra una captura de pantalla del código mencionado, que se aplica al bloque 1 del diagrama de flujo correspondiente a la figura 38 del capítulo 4.

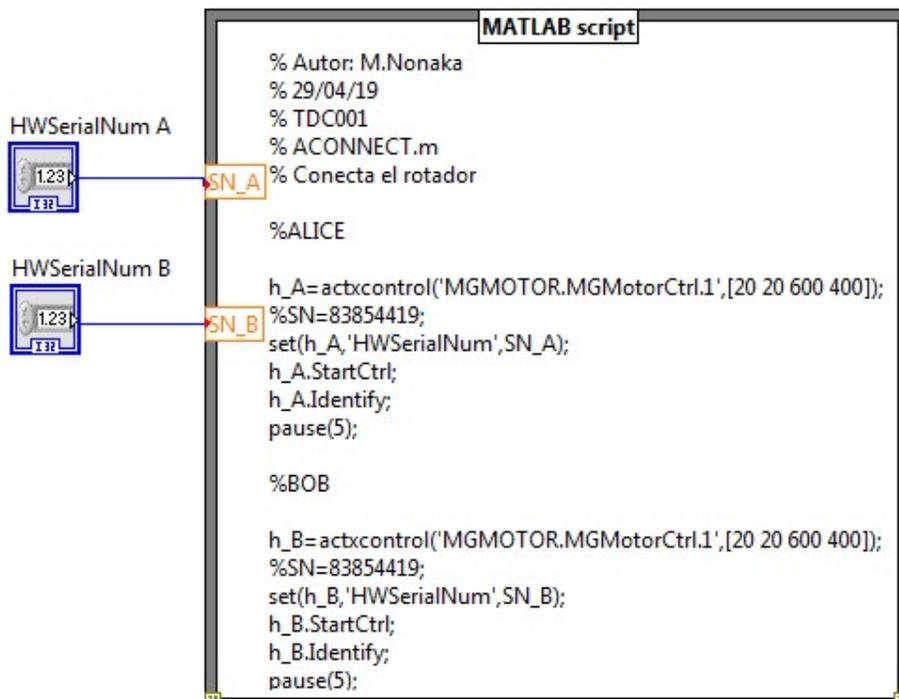


Figura B.1 Código correspondiente al bloque 1 del diagrama de flujos de la figura 37

B.2 Codificación mueve a home: este código permite rotar a la posición HOME (0 grados) a ambos rotadores de las estaciones A y B. En la captura de la figura B.2, se muestra para el caso

del rotador de la estación A. Para poder realizar el movimiento de los rotadores, fue necesario la lectura de los bits de estado que se solicitan al dispositivo. Cabe mencionar aquí, que el estado HOME, es en realidad al revés de lo que se indica en el manual de usuario, esto fue un error con el cual se demoró el control de dicho dispositivo, ya que el dato en el manual de usuario no estaba correcto (el valor del r_A como se indica en el código de abajo, debe ser comparado con el valor 1). La figura B.2, corresponde a la codificación que se aplica al bloque 1 del diagrama de flujo correspondiente a la figura 38 del capítulo 4.

```

MATLAB script
% Autor: M.Nonaka
% 29/04/19
% TDC001
% AHome.m
% Mueve a Home el rotador

%ALICE

h_A.MoveHome(0,0);
pause(2);
statusbits_A=h_A.GetStatusBits_Bits(0);
numero_A=abs(statusbits_A);
B_A=decimalToBinaryVector(numero_A,32,'LSBFirst');
r_A=B_A(11); %Homed?

while(r_A==1) %AL REVES DEL APT USER 0 HOMED 1 HOMING
pause(2);
numero_A=abs(statusbits_A);
B_A=decimalToBinaryVector(numero_A,32,'LSBFirst');
r_A=B_A(11);
statusbits_A=h_A.GetStatusBits_Bits(0);
end
    
```

Figura B.2. Código correspondiente al bloque 2 de la figura 38

B.3 Codificación mueve a home: este código permite rotar a una cierta posición ingresada en la variable "angulo_A", para el caso de la estación A, y "angulo_B", para el caso de la estación B. En la figura B.3, se puede observar la codificación que se implementa al número 3 del diagrama de flujos de la figura 38. Es importante que el rotador llegue a la posición correcta, ya que de ello dependerá que las láminas de media onda montadas en estos rotadores, se encuentren en la posición correcta. Por lo cual, este código, espera y consulta al dispositivo, si ya ha llegado a la posición ingresada.

```
MATLAB script

% Autor: M.Nonaka
% 29/04/19
% TDC001
% AHome.m
% Mueve a cierta POSICION el rotador

%ALICE

h_A.SetAbsMovePos(0,POSICION_A);
h_A.MoveAbsolute(0,1==0);
statusbits_A=h_A.GetStatusBits_Bits(0);

numero_A=abs(statusbits_A);
B_A=decimalToBinaryVector(numero_A,32,'LSBFirst');
r_A=and(B_A(5),B_A(6));

while(r_A==0)
    pause(2);
    numero_A=abs(statusbits_A);
    B_A=decimalToBinaryVector(numero_A,32,'LSBFirst');
    r_A=and(B_A(5),B_A(6));
    statusbits_A=h_A.GetStatusBits_Bits(0);
end
```

↑ angulo_A → POSICION_A

Figura B.3. Código correspondiente al bloque 3 de la figura 38

APÉNDICE C

CAPTURAS EN PANTALLA DE SOFTWARE DESARROLLADO EN LABVIEW

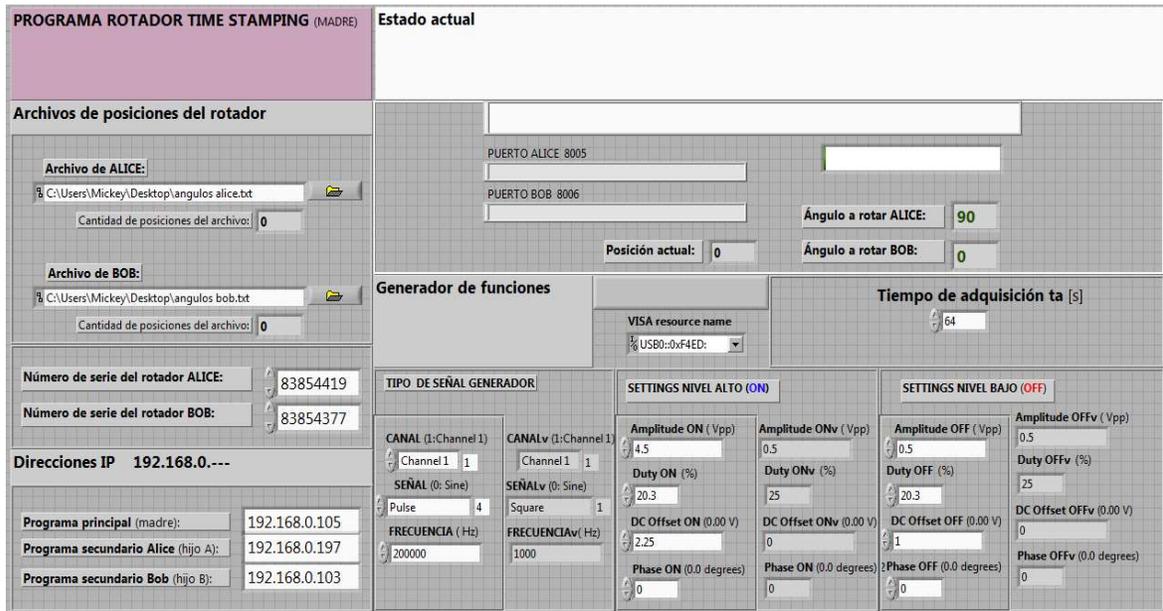


Figura C.1 Captura de pantalla del software madre

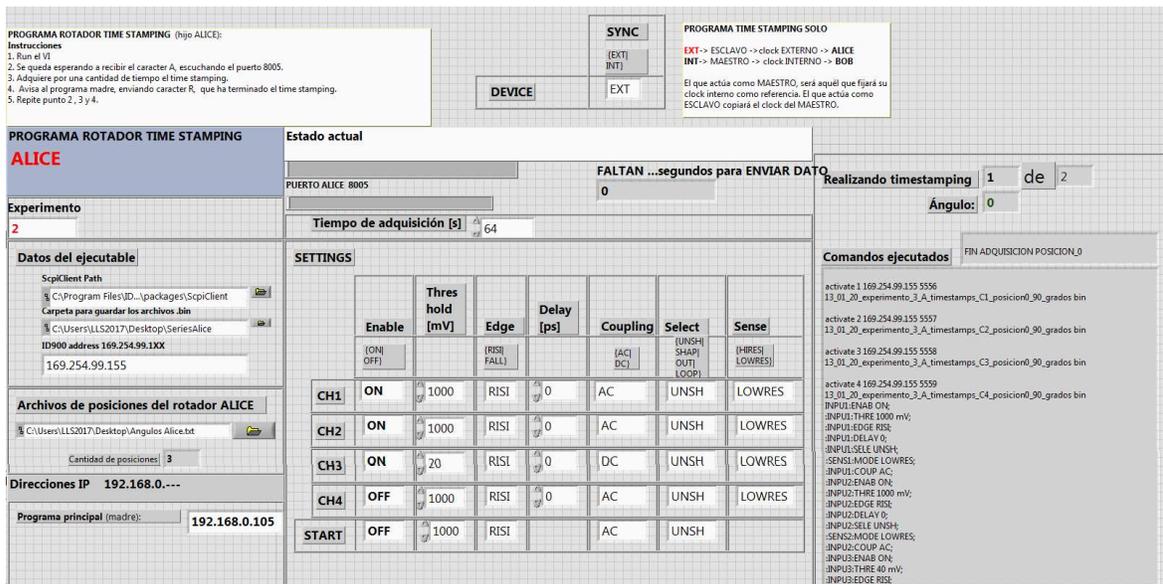


Figura C.2 Captura de pantalla del software *time stamping* Alice

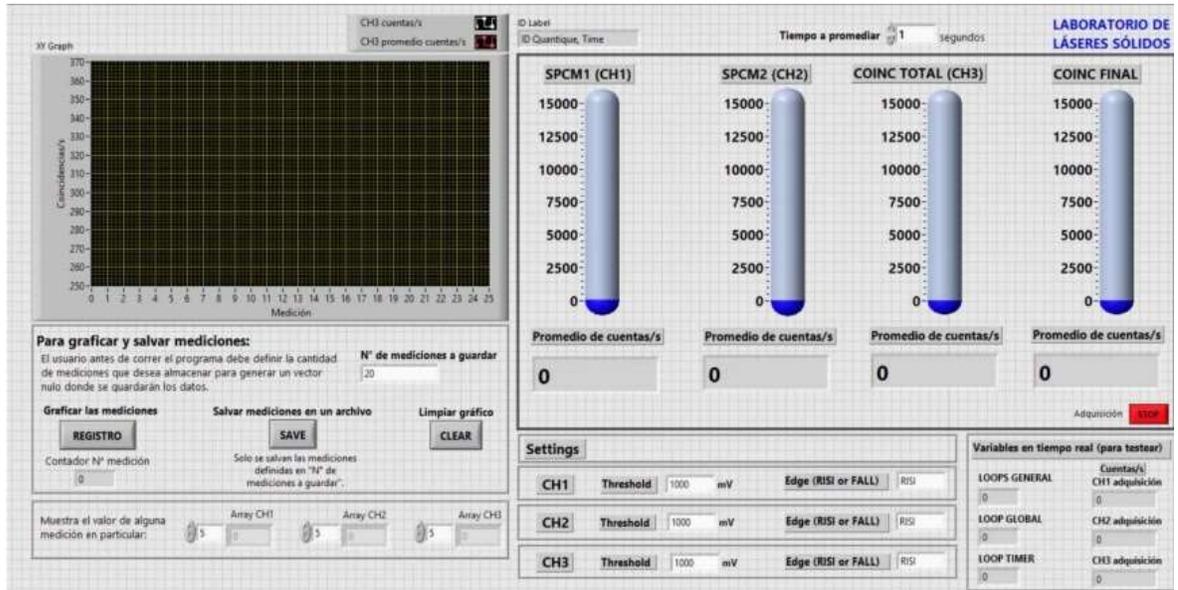


Figura C.3 Captura de pantalla del Software de coincidencias en LabVIEW

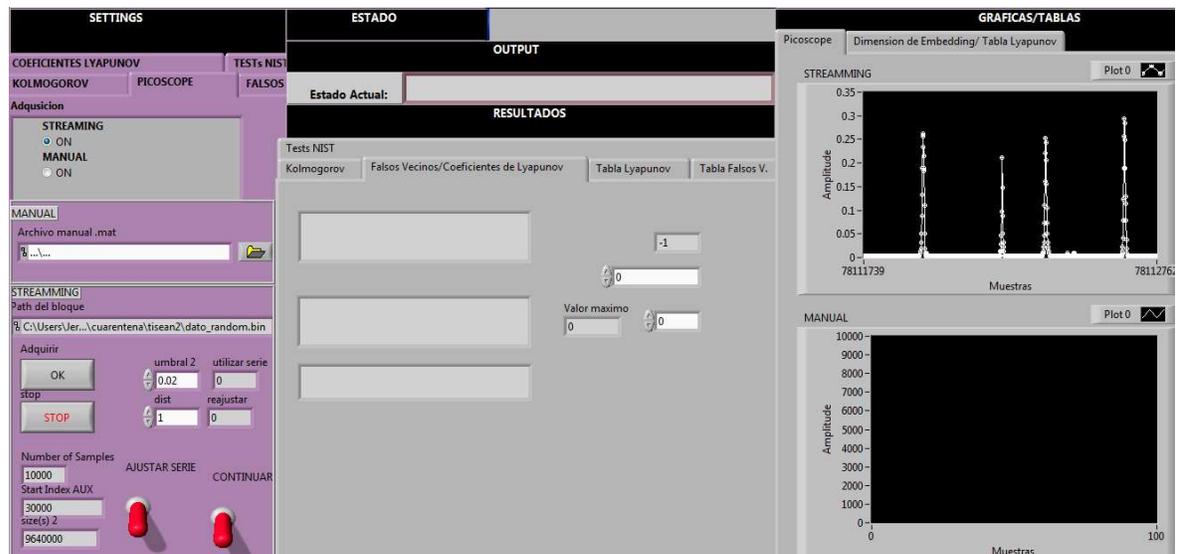


Figura C.4 Captura de pantalla del software de *streaming* para láser con modulación de pérdidas.

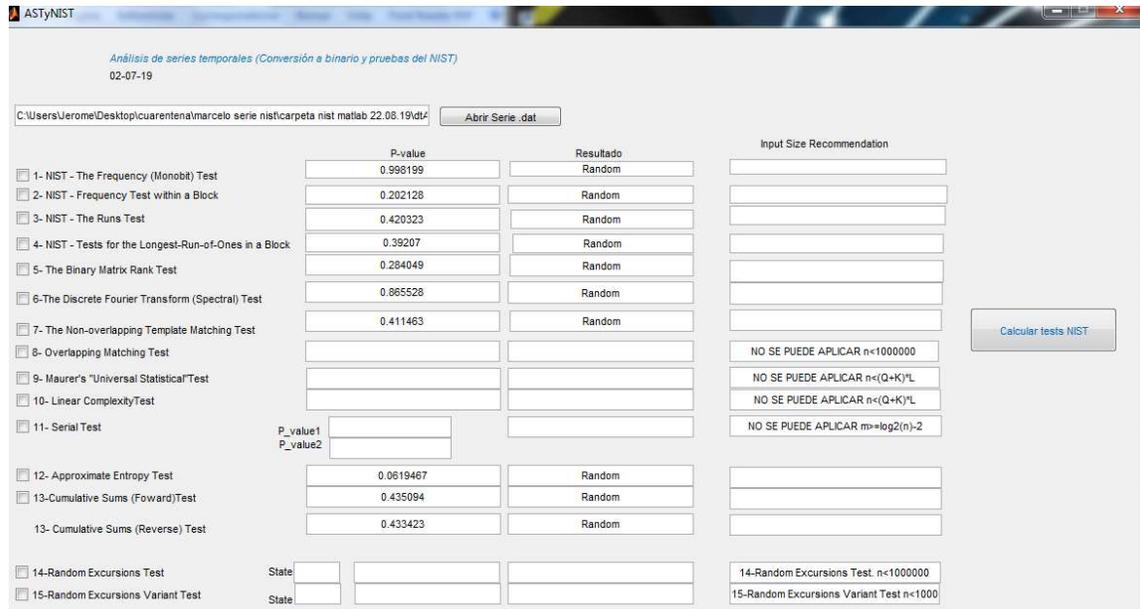


Figura C.5 Captura de pantalla de GUI o app tests del Nist en MATLAB

APÉNDICE D

CÓDIGOS DE ALGORITMOS DE PROCESAMIENTO DESARROLLADOS/UTILIZADOS EN MATLAB

D.1 Código fuente correspondiente a la función [m] = ajustarserie_bin(umbral,dist)

```
function [ m ] = ajustarserie_bin(umbral,dist)
% Funcion ajustarserie_bin
% Fecha 26.05.20
% Autora: M.Nonaka
%-----
% Descripción
%-----
% Lee el archivo .bin que se encuentra en la ruta en
% donde se guarda el bloque de señal del PicoScope.
% Grafica los datos. Filtra los valores que se
% encuentran por debajo del "umbral". Halla los picos
% que se encuentren a una distancia de muestras minima
% de separación "dist". Esta función sirve para ajustar
% los valores de dist y umbral de la señal, como testeo
% previo.
%-----
% Argumentos de entrada:
% umbral: valor que se toma de umbral como referencia
% en Volts
% dist: separación mínima de muestras para hallar picos
%-----
% Salida:
% Retorna valor m, arbitrario, como chequeo que se ha
% realizado la función.

%Ruta del archivo .bin del PicoScope
fileID=fopen('C:\Users\Myriam\Desktop\cuarentena\tisean2\dato_random.bin');
B = fread(fileID,'int32');
figure;
plot(B);
hold on;

%Filtrado de umbral
s1=find(B>umbral);
s2=B(s1);
plot(s1,s2,'*');

%Detección de Picos
[pks,locs] = findpeaks(s2, 'MINPEAKDISTANCE',dist);
%tiempo
tiempo=s1(locs);
%amplitud
```

```
amplitud=pks;
```

```
hold on;
plot(tiempo,amplitud,'go');
fclose(fileID);
m=3;
end
```

D.2 Código fuente correspondiente a la función [m] = guardarserie_bin_dat(umbral,dist)

```
function [ m ] = guardarserie_bin_dat( umbral,dist )
% Funcion guardarserie_bin
% Fecha 26.05.20
% Autora: M.Nonaka
%-----
% Descripción
%-----
% Una vez obtenidos los valores de umbral y dist ejecutando
% ajustarserie_bin.m se procede a obtener una serie temporal binarizada y
% salvaguardar los valores obtenidos.
% Argumentos de entrada:
% umbral: valor que se toma de umbral como referencia
% en Volts
% dist: separación mínima de muestras para hallar picos
%-----
% Salida:
% Retorna valor m, arbitrario, como chequeo que se ha
% realizado la función.

%Ruta del archivo .bin del PicoScope
fileID=fopen('C:\Users\Myriam\Desktop\cuarentena\tisean2\dato_random.bin');
B = fread(fileID,'int32');
figure;
plot(B);
hold on;

%Filtrado de umbral
s1=find(B>umbral);
s2=B(s1);
plot(s1,s2,'*');

%Detección de Picos
[pks,locs] = findpeaks(s2, 'MINPEAKDISTANCE',dist);
%tiempo
tiempo=s1(locs);
%amplitud
amplitud=pks;

hold on;
```

```

plot(tiempo,amplitud,'go');
%vector en donde se guarda
VEC=[amplitud tiempo];

%guarda picos
sorig_t=VEC(1:100001,2);
sorig_a=VEC(1:100001,1);
s=sorig_a;

%Binarización de la serie
m=median(s); %saco la media para el algoritmo de conversión de 1 y 0 (kolbin). median.
N=size(s);
aux1=find(s>=m);
aux2=find(s<m);

s(aux1)=1;
s(aux2)=0;
epsilon=s; %epsilon en serie binarizada

figure;
hold on;
stem(sorig_t,sorig_a);

hold off;
figure;
hold on;
stem(sorig_t,s);
hold off;

%Salvaguada de serie temporal amplitud con valores binarizados .bin
fileID2 =
fopen('C:\Users\Myriam\Desktop\cuarentena\tisean2\serie_random7_binaria_amplitud.bin','w+');
fwrite(fileID2, epsilon, 'int32');

%Salvaguada de serie temporal amplitud con valores sin binarizar .bin
fileID3 =
fopen('C:\Users\Myriam\Desktop\cuarentena\tisean2\serie_random7_original_amplitud.bin','w+');
fwrite(fileID3, sorig_a, 'int32');

%Salvaguada de serie temporal tiempo con valores sin binarizar
fileID4 =
fopen('C:\Users\Myriam\Desktop\cuarentena\tisean2\serie_random7_original_tiempo.bin','w+');
;
fwrite(fileID4, sorig_t, 'int32');

%Salvaguada de serie temporal amplitud sin binarizar en .dat
save('C:\Users\Myriam\Desktop\cuarentena\tisean2\serie_random7_original_amplitud.dat','sorig_a','-ascii');

```

```
fclose(fileID);
```

```
fclose(fileID2);
```

```
fclose(fileID3);
```

```
fclose(fileID4);
```

```
m=3;
```

```
end
```

D.3 Código fuente correspondiente a la función [complejidad] = calculokolmogorov(input_args)

```
function [ complejidad ] = calculokolmogorov( input_args )
```

```
% Funcion guardarserie_bin
```

```
% Fecha 26.05.20
```

```
% Autora: M.Nonaka
```

```
%-----
```

```
% Descripción
```

```
%-----
```

```
% Calculo de la complejidad de Kolmogorov utilizando kolmogorov.m
```

```
% Se utiliza la serie binarizada
```

```
fileID=fopen('C:\Users\Myriam\Desktop\cuarentena\tisean2\serie_random7_binaria_amplitud.  
bin');
```

```
s = fread(fileID,'int32');
```

```
complejidad=kolmogorov(s);
```

```
fclose(fileID);
```

```
end
```

D.4 Código fuente correspondiente al script kolbin.m

```
% FUNCIÓN: kolbin.m
```

```
% Binariza serie de datos respecto un umbral
```

```
v=input('ingrese el archivo','s')
```

```
s=load(strcat(v));
```

```
m=mean(s);
```

```
N=size(s);
```

```
for i=1:N
```

```
    if s(i)>m
```

```
        s(i)=1;
```

```
    else
```

```
        s(i)=0;
```

```
    end
```

```
end
```

 kolmogorov(s)
D.5 Código fuente correspondiente a la función complexity=kolmogorov(s)

```

% FUNCTION: kolmogorov.m
% DATE: 9th Feb 2005
% AUTHOR: Stephen Faul (stephenf@rennes.ucc.ie)
%
% Function for estimating the Kolmogorov Complexity as per:
% "Easily Calculable Measure for the Complexity of Spatiotemporal Patterns"
% by F Kaspar and HG Schuster, Physical Review A, vol 36, num 2 pg 842
%
% Input is a digital string, so conversion from signal to a digital stream
% must be carried out a priori

function complexity=kolmogorov(s);
n=length(s);
c=1;
l=1;

i=0;
k=1;
k_max=1;
stop=0;

while stop==0
  if s(i+k)~=s(l+k)
    if k>k_max
      k_max=k;
    end
    i=i+1;

    if i==l
      c=c+1;
      l=l+k_max;
      if l+1>n
        stop=1;
      else
        i=0;
        k=1;
        k_max=1;
      end
    else
      k=1;
    end
  else
    k=k+1;
    if l+k>n
      c=c+1;
      stop=1;
    end
  end
end

```

```
end  
end  
end
```

```
b=n/log2(n);
```

```
% a la Lempel and Ziv (IEEE trans inf theory it-22, 75 (1976),  
%  $h(n)=c(n)/b(n)$  where  $c(n)$  is the kolmogorov complexity  
% and  $h(n)$  is a normalised measure of complexity.  
complexity=c/b;
```

APÉNDICE E

COMANDOS SPCI UTILIZADOS EN EL ID900 PARA EL DESARROLLO DEL SOFTWARE PROPIO

Comandos que se utilizaron en la implementación de los programas en el área cuántica. Los mismos se ordenan según el orden de uso.

Listado de comandos del ID900		
Comandos para el canal número # (1,2,3,4)	Descripción	
1	INPU#: ENAB ON	Habilita el canal de entrada # para la adquisición de señal. Si fuese el canal de START en vez del INPU# se coloca STAR:ENAB ON.
2	INPU#:THRE D mV	Configura el umbral del nivel de disparo al valor D en unidades de mV para la detección de señal del canal #. Si fuese el canal de START en vez del INPU# se coloca STAR:THRE D mV.
3	INPU1:EDGE L	Configura el flanco de disparo del canal # mediante L que puede ser RISE (<i>rising</i>) o FALL (<i>falling</i>).
4	INPU#:DELAY D	Configura el delay del canal # a un valor D. Por default es 0. Si fuese el canal de START en vez del INPU# se coloca STAR:DELAY D.
5	INPU#:SELE L	Selecciona lo que alimenta el bloque de entrada de cada canal # mediante L que puede ser UNSH (<i>unshaped</i>), SHAP (<i>shaped</i>), OUT (<i>output</i>) o LOOP (<i>loop</i>).
6	SENS#:MODE L	Selecciona el modo de operación del canal # mediante la letra L que puede ser HIRE (<i>hi resolution</i>) o LOWRES (<i>low resolution</i>).
7	INPU#:COUP L	Configura el acople del canal # mediante la letra L que puede ser AC o DC.
8	DEVICE:SYNC T	Selecciona el reloj de referencia. En T, se debe indicar INT o EXT. El primero será para trabajar como clock maestro y el segundo, para trabajar como clock esclavo.
9	TSGE8:ONES:PWID D00000000	Configuración del generador interno del dispositivo en donde D es el valor en milisegundos.
10	TSST#:DATA:RAW:ERRO RS:CLEAR	Limpia el indicador de errores del canal #.
11	TSGE8:ENAB ON	Habilita el generador interno.
12	TSST#: DATA:RAW:SEND ON	Activa la adquisición de <i>time stamping</i> del canal #.
13	TSST#: DATA: RAW:SEND OFF	Desactiva la adquisición de <i>time stamping</i> del canal #.
14	TSGE8:ENAB OFF	Deshabilita el generador interno.
15	TSST#:DATA:RAW:ERRO RS?	Comando para consultar el estado de error en la transferencia de datos del canal #.

16	INPU#: ENAB OFF	Habilita el canal de entrada # para la adquisición de señal. Si fuese el canal de START en vez del INPU# se coloca STAR:ENAB OFF.
-----------	------------------------	--

APÉNDICE F

IMPLEMENTACIÓN EN LABVIEW PARA EL CÁLCULO DE LA DIMENSIÓN DE *EMBEDDING* Y DE LOS COEFICIENTES DE LYAPUNOV CON PAQUETES DE TISEAN

El paquete de software TISEAN [31], por "*Time Series Analysis*", es un programa para el análisis de series temporales con métodos basados en la teoría de la dinámica no lineal.

Este paquete se incorporó a nuestro programa de adquisición y procesamiento mencionados en la sección 4.3.4, del capítulo 4, de la siguiente forma:

Implementación de cálculo de dimensión de *embedding* en LabVIEW con `false_nearest` de TISEAN:

Se utilizó el programa "`false_nearest`" de TISEAN [38], que busca todos los vecinos más cercanos de todos los puntos en las m dimensiones e itera estos vecinos un paso adelante en el futuro. Si la relación entre la distancia de la iteración y la del vecino más cercano supera un umbral determinado, el punto se marca como vecino incorrecto. El programa da como resultado una fracción de los falsos vecinos para una dimensión de *embedding* específica [33].

El comando para ejecutar dicho programa en MATLAB es el siguiente:

```
!false_nearest "C:\Users\...\serie_temporal.dat" -oresultado.txt -M# -d#
```

Como primer argumento de la línea de comando, se inserta el llamado del programa `false_nearest` con `!false_nearest` seguido entre comillas con la ruta en donde se encuentra el `.dat`, es decir, la serie temporal de picos de amplitud (esta serie debe ser la original, es decir, sin binarizar). Como tercer argumento, se debe ingresar `-oseguido` sin espacios del archivo en donde se guardarán los datos en formato `.txt`, en este caso lo guardaremos en un archivo llamado "resultado.txt", o sea, se debería escribir `-oresultado.txt`. Finalmente, se deberá ingresar `-M10 -d1`, esto significa máxima dimensión de los vectores de retardo $\#$, por lo general se ingresa 10, y retardo de los vectores a $\#$, por lo general se ingresa a 1.

En LabVIEW se implementó dicho comando con MATLAB Script node de la siguiente forma realizando los pasos descritos en la tabla F.1:

Paso Nro	Descripción
1	Se ingresa como input (entrada) del MATLAB script, el path (ruta) del archivo, el nombre del archivo, el # de M y el # de d, como se muestra en la figura F.1.
2	Se ejecuta el comando en MATLAB script figura F.1.
3	Se procede a abrir el archivo de salida (resultado.txt). Como en la figura F.2. Y se guarda el primer coeficiente de cada línea en vectores (figura F.3).
4	Se comparan los tres últimos valores, si son decrecientes (figura F.4). En caso afirmativo, se chequea que el último valor sea menor o igual al umbral. En caso afirmativo, se busca el valor de dimensión correspondiente a dicho coeficiente (figura F.5). Este valor, DIMENSION UMBRAL, será el valor de la dimensión de <i>embedding</i> , ver figura F.6. En el caso de no hallar dimensión, dicho valor toma el valor -1.

Tabla F.1. Pasos en LabVIEW para el cálculo de la dimensión de *embedding* con programa false_nearest de TISEAN.

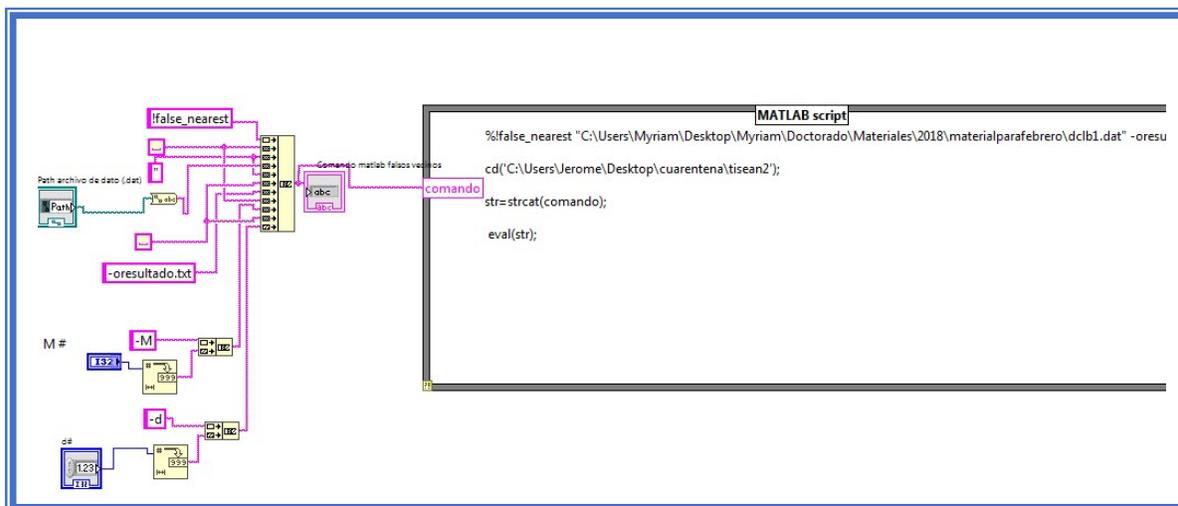


Figura F.1. Input del comando para ejecución del false_nearest en MATLAB script Node de LabVIEW.

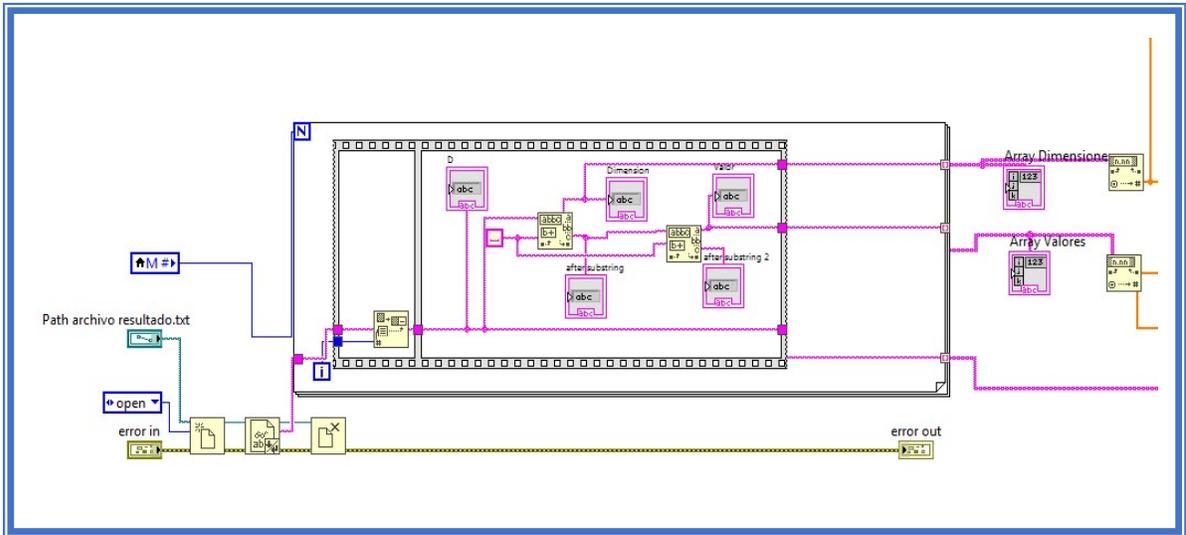


Figura F.2. Captura de pantalla de apertura de archivo resultado.txt de los *outputs* (salidas) del programa false_nearest de TISEAN en LabVIEW y salvaguarda en arreglos.

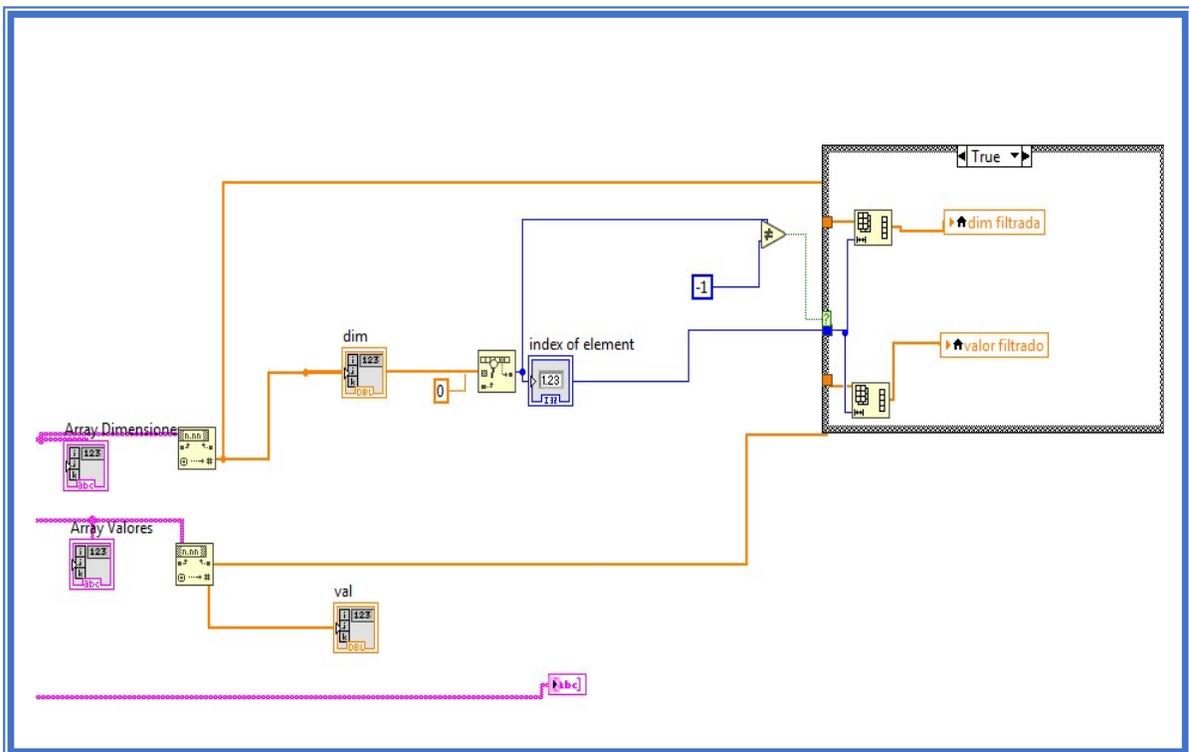


Figura F.3. Captura de pantalla de salvaguarda de primeros coeficientes de cada dimensión del archivo resultado.txt en LabVIEW.

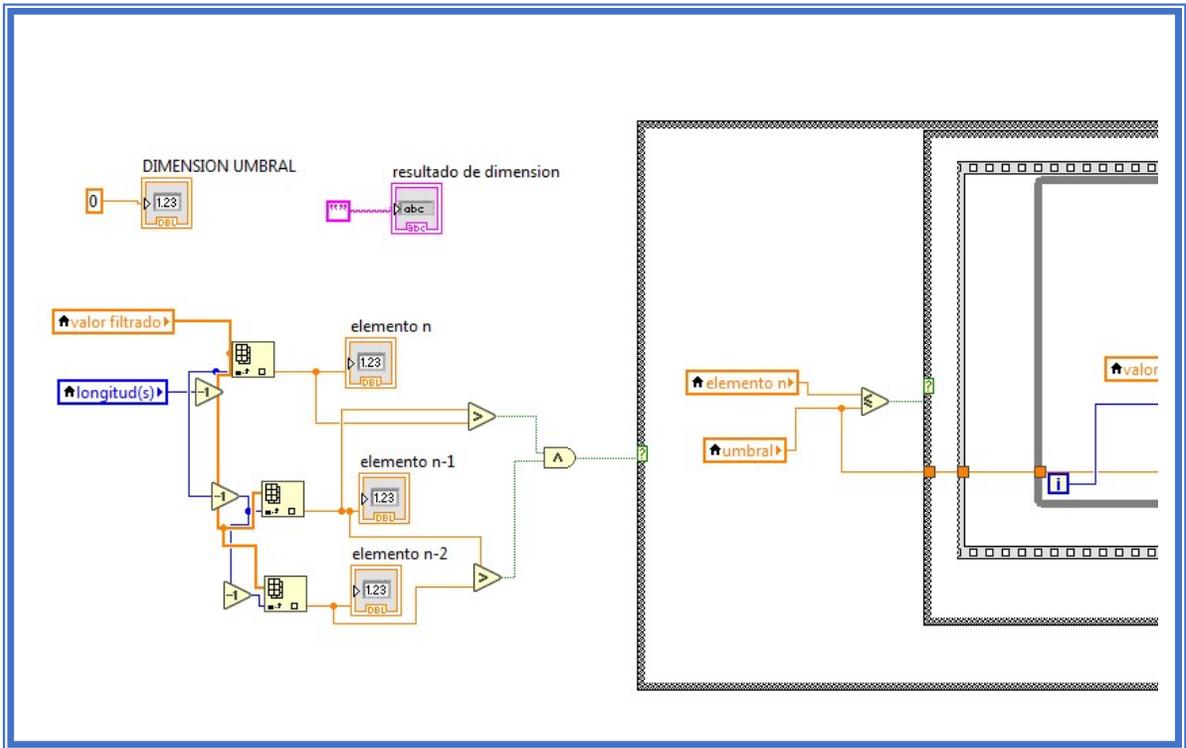


Figura F.4. Captura de pantalla de comparación (decreciente) de últimos valores de tres elementos en los arreglos de la figura F.2 en LabVIEW.

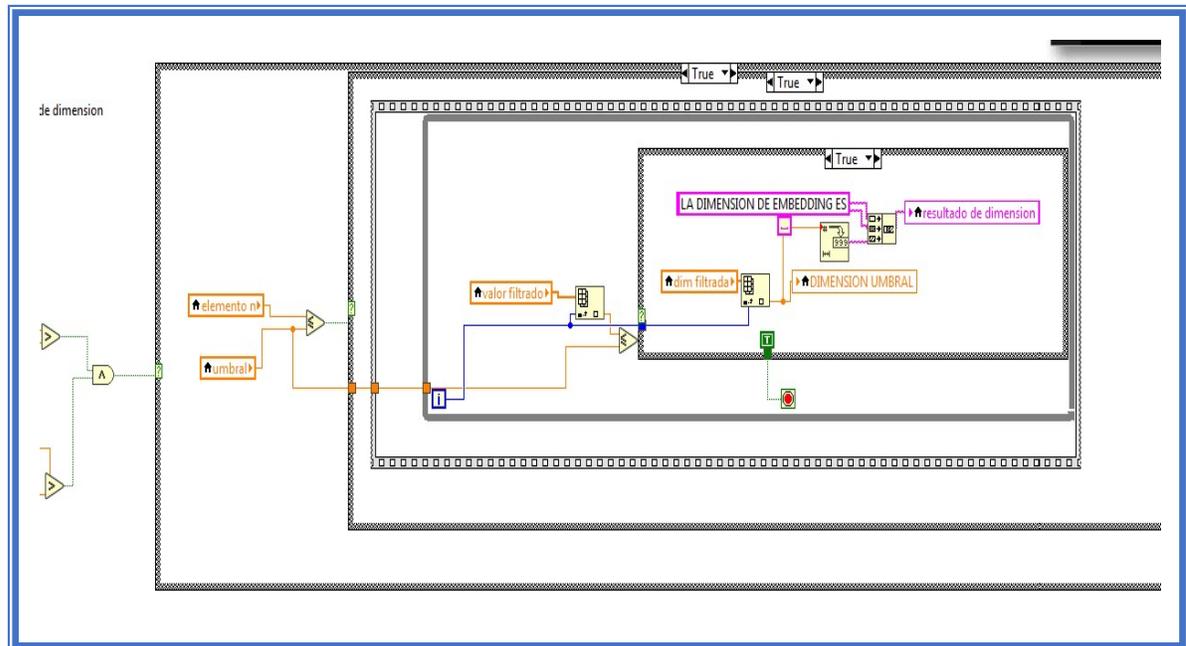


Figura F.5. Captura de pantalla de comparación de último elemento decreciente del arreglo respecto del valor umbral configurado en LabVIEW.

Implementación de cálculo de los exponentes de Lyapunov en LabVIEW con `lyap_spec` de TISEAN:

Se utilizó el programa "`lyap_spec`" de TISEAN [39] https://www.pks.mpg.de/~tisean/TISEAN_2.1/docs/docs_c/lyap_spec.html, que estima todo el espectro de exponentes de Lyapunov para una serie temporal dada, posiblemente multivariante. Todo el espectro significa: si se dan d componentes y la dimensión de *embedding* es m , entonces se determinarán $m*d$ exponentes. Este método se basa en el trabajo de Sano y Sawada [40].

El comando para ejecutar dicho programa en MATLAB es el siguiente:

```
!lyap_spec "C:\Users\...\serie_temporal.dat" -oresultado_lyap.txt -m1,# -d#
```

Como primer argumento de la línea de comando, se inserta el llamado del programa `lyap_spec` con `!lyap_spec` seguido entre comillas con la ruta en donde se encuentra el `.dat`, es decir, la serie temporal de picos de amplitud (esta serie debe ser la original, es decir, sin binarizar). Como tercer argumento, se debe ingresar `-oseguido` sin espacios del archivo en donde se guardarán los datos en formato `.txt`, en este caso lo guardaremos en un archivo llamado "`resultado_lyap.txt`", o sea, se debería escribir `-oresultado_lyap.txt`. Finalmente, se deberá ingresar `-m1,# -d#`, esto significa $m1$,# el número de componentes (1) y la dimensión de *embedding* calculada (#), y $d#$, corresponde dimensión de los vectores de retardo #, se ingresa 1.

En LabVIEW se implementa dicho comando con MATLAB Script node:

Paso Nro	Descripción
1	Como primer paso, antes de realizar el cálculo de los exponentes de Lyapunov, se debe constatar que exista dimensión de <i>embedding</i> . Si el valor de DIMENSION UMBRAL es igual a -1, no existe dimensión de <i>embedding</i> y no se pueden calcular los exponentes de Lyapunov (figura F.6).
2	En el caso de ser posible el cálculo de exponentes de Lyapunov, se ingresa como input (entrada) del MATLAB script, el path (ruta) del archivo, el nombre del archivo, el # de m 1,# con el # correspondiente a la DIMENSION UMBRAL obtenida con <code>false_nearest</code> y el # de d , como se muestra en la figura F.7. Se ejecuta el comando en MATLAB script.
3	Se procede a abrir el archivo de salida (<code>resultado_lyap.txt</code>). Como en la figura F.8. Y se guardan los coeficientes del último dato antes del numeral.
4	Con los datos anteriores, se verifica que al menos un coeficiente sea positivo y que la sumatoria de coeficientes sea negativa (figura F.9). Si cumple dicha condición entonces es una señal caótica, caso contrario no lo es.

Tabla F.2. Pasos en LabVIEW para el cálculo de los exponentes de Lyapunov con programa `lyap_spec` de TISEAN.

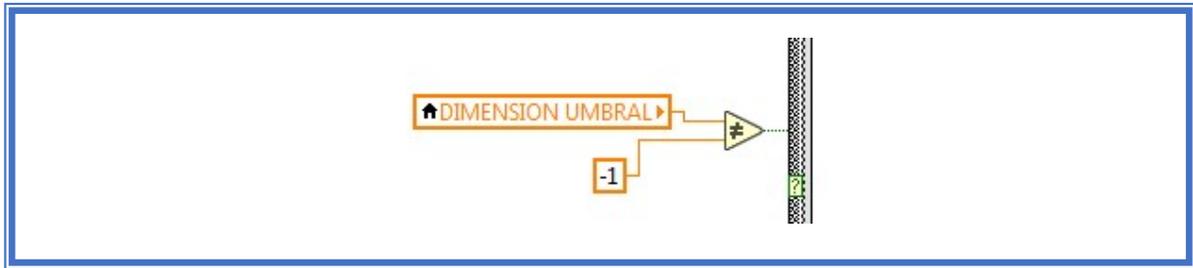


Figura F.6. Captura de pantalla si se ha detectado dimensión de *embedding* en LabVIEW.

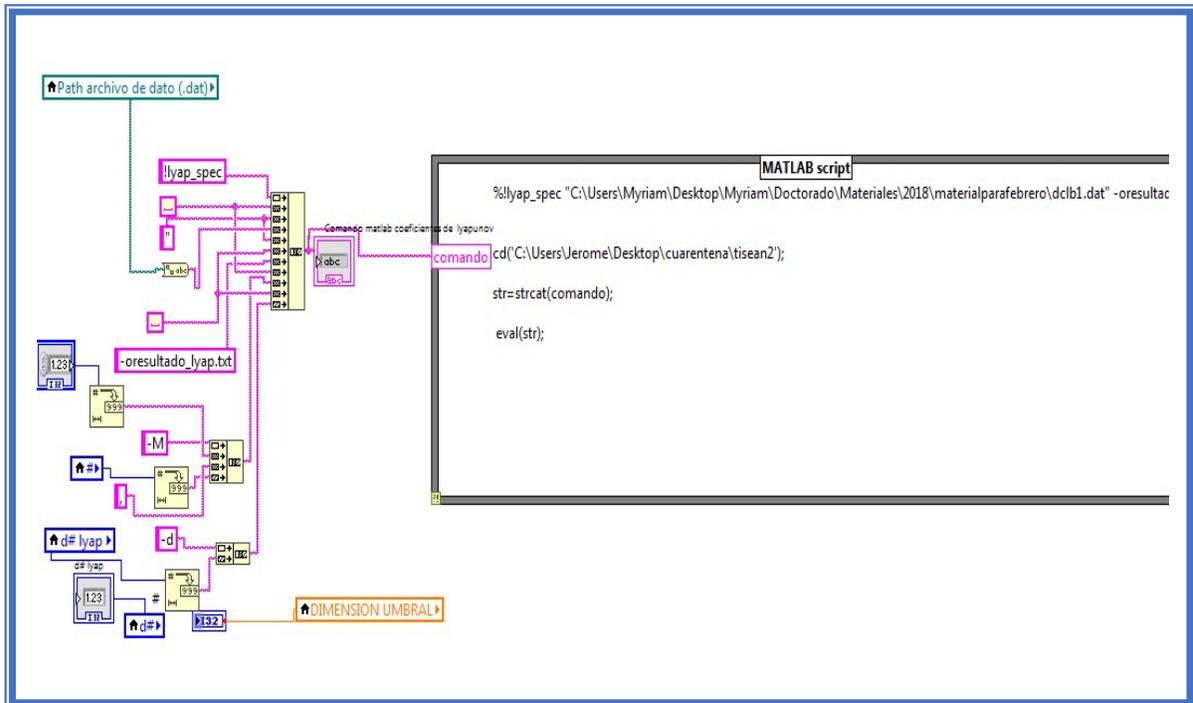


Figura F.7. Captura de pantalla de apertura de archivo resultado_lyap.txt de los *outputs* (salidas) del programa lyap_spec de TISEAN en LabVIEW.

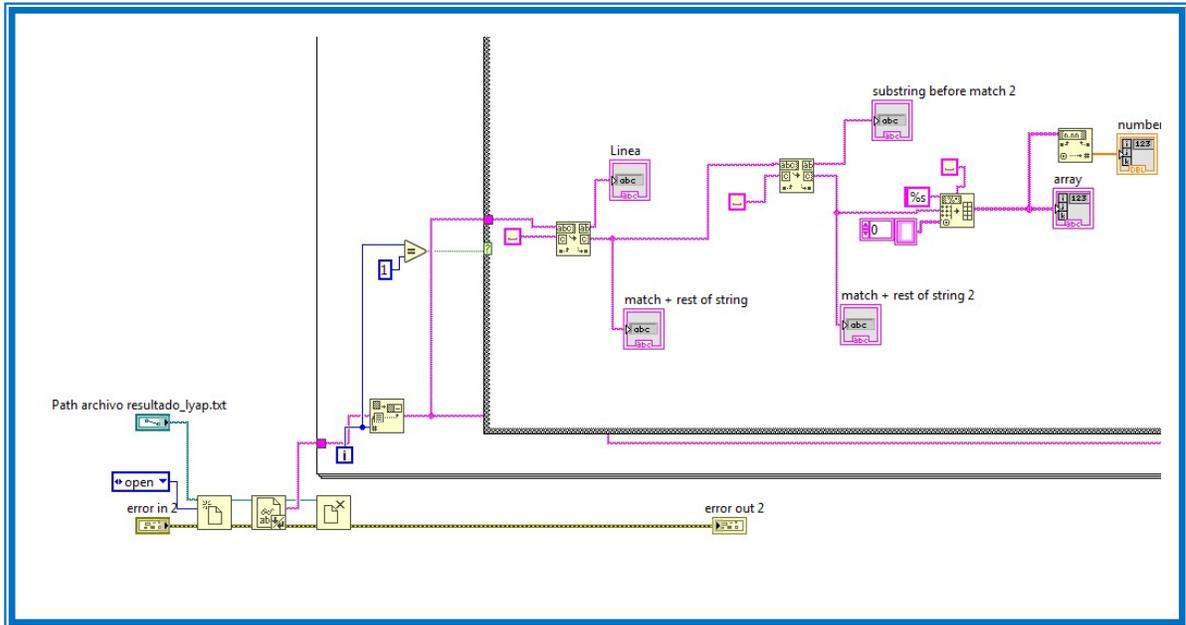


Figura F.8. Captura de pantalla de apertura de archivo resultado_lyap.txt de los *outputs* (salidas) del programa lyap_spec de TISEAN en LabVIEW y salvaguarda en arreglos.

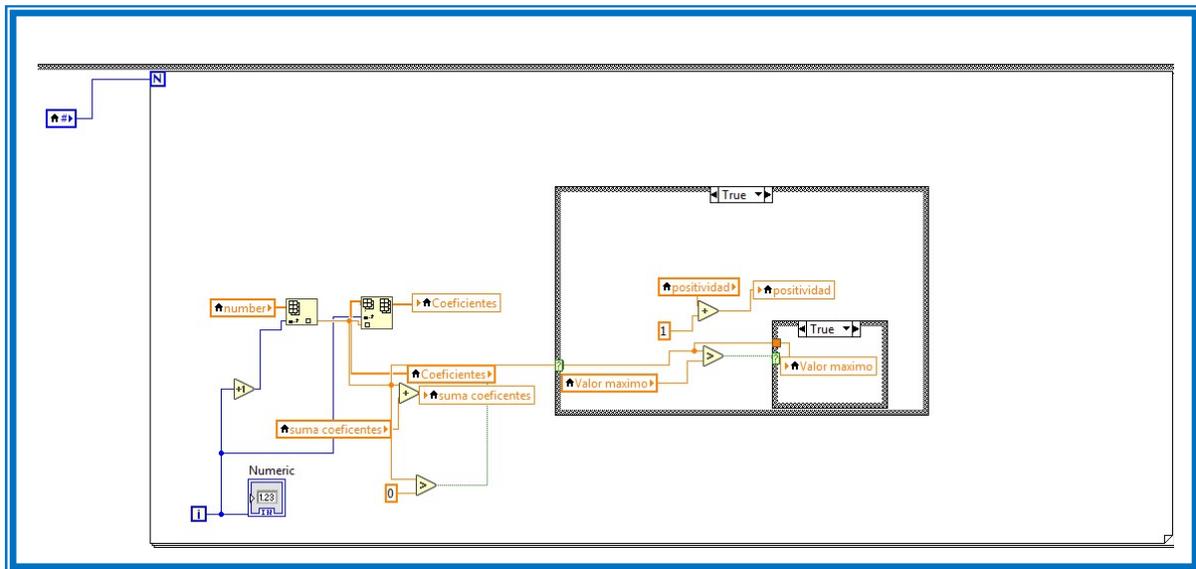


Figura F.9. Captura de pantalla de comparación de al menos un coeficiente positivo y sumatoria de coeficientes total negativa en LabVIEW.

APÉNDICE G

ECUACIONES Y REFERENCIAS DE NOMENCLATURAS PARA EL CÁLCULO DEL PARÁMETRO DE BELL S_{CHSH}

El cálculo del parámetro de Bell S_{CHSH}, surge de realizar el siguiente cálculo de la ecuación:

$$S = \sum |E_{\alpha\beta}| = S_0 + S_1 + S_2 + S_3 \quad \text{Ecuación G.1}$$

De donde $E_{\alpha\beta}$, para el caso de los valores utilizados en nuestro trabajo de tesis:

$$E_{\alpha=0 \text{ rad } \beta=\frac{\pi}{16}} = \frac{A_0B_0+A_2B_2-[A_0B_2+A_2B_0]}{A_0B_0+A_2B_2+[A_0B_2+A_2B_0]} \quad \text{Ecuación G.2}$$

$$E_{\alpha=0 \text{ rad } \beta=\frac{3\pi}{16}} = \frac{A_0B_1+A_2B_3-[A_0B_3+A_2B_1]}{A_0B_1+A_2B_3+[A_0B_3+A_2B_1]} \quad \text{Ecuación G.3}$$

$$E_{\alpha=\frac{\pi}{8} \text{ rad } \beta=\frac{\pi}{16}} = \frac{A_1B_0+A_3B_2-[A_1B_2+A_3B_0]}{A_1B_0+A_3B_2+[A_1B_2+A_3B_0]} \quad \text{Ecuación G.4}$$

$$E_{\alpha=\frac{\pi}{8} \text{ rad } \beta=\frac{3\pi}{16}} = \frac{A_1B_1+A_3B_3-[A_1B_3+A_3B_1]}{A_1B_1+A_3B_3+[A_1B_3+A_3B_1]} \quad \text{Ecuación G.5}$$

Reemplazando la ecuación G.1 por las ecuaciones G.2, G.3, G.4 y G.5:

$$S = \left| \frac{A_0B_0+A_2B_2-[A_0B_2+A_2B_0]}{A_0B_0+A_2B_2+[A_0B_2+A_2B_0]} \right| + \left| \frac{A_0B_1+A_2B_3-[A_0B_3+A_2B_1]}{A_0B_1+A_2B_3+[A_0B_3+A_2B_1]} \right| + \left| \frac{A_1B_0+A_3B_2-[A_1B_2+A_3B_0]}{A_1B_0+A_3B_2+[A_1B_2+A_3B_0]} \right| + \left| \frac{A_1B_1+A_3B_3-[A_1B_3+A_3B_1]}{A_1B_1+A_3B_3+[A_1B_3+A_3B_1]} \right| \quad \text{Ecuación G.6}$$

Cada parámetro S de la ecuación G.1 , corresponde a las coincidencias detectadas C entre los ángulos α (de la estación A) y β (de la estación B) , de los SPCM correspondientes a entrada roja + o blanca -, ver nomenclatura simplificada en tablas G.1,G.2,G.3,G.4.

Para el S_0 se tiene el juego de valores angulares utilizados:

S_0	$\alpha = 0 \text{ rad} 0^\circ$	β		Coincidencias	Nomenclatura simplificada
		$\frac{\pi}{16} \text{ rad} 11.25^\circ$	→	$C_{\alpha=0 \text{ rad } \beta=\frac{\pi}{16} \text{ rad}}^{++}$	A0B0
		$\frac{3 \cdot \pi}{16} \text{ rad} 33.75^\circ$	→	$C_{\alpha=0 \text{ rad } \beta=\frac{3 \cdot \pi}{16} \text{ rad}}^{++}$	A0B1
		$\frac{5 \cdot \pi}{16} \text{ rad} 56.25^\circ$	→	$C_{\alpha=0 \text{ rad } \beta=\frac{5 \cdot \pi}{16} \text{ rad}}^{+-}$	A0B2
		$\frac{7 \cdot \pi}{16} \text{ rad} 78.75^\circ$	→	$C_{\alpha=0 \text{ rad } \beta=\frac{7 \cdot \pi}{16} \text{ rad}}^{+-}$	A0B3

Tabla G.1. Coincidencias correspondientes al parámetro S_0

Para el S_1 se tiene el juego de valores angulares utilizados:

S_1	$\alpha = \frac{\pi}{8} \text{ rad} 22,5^\circ$	β		Coincidencias	Nomenclatura simplificada
		$\frac{\pi}{16} \text{ rad} 11.25^\circ$	→	$C_{\alpha=\frac{\pi}{8} \text{ rad } \beta=\frac{\pi}{16} \text{ rad}}^{++}$	A1B0
		$\frac{3 \cdot \pi}{16} \text{ rad} 33.75^\circ$	→	$C_{\alpha=\frac{\pi}{8} \text{ rad } \beta=\frac{3 \cdot \pi}{16} \text{ rad}}^{++}$	A1B1
		$\frac{5 \cdot \pi}{16} \text{ rad} 56.25^\circ$	→	$C_{\alpha=\frac{\pi}{8} \text{ rad } \beta=\frac{5 \cdot \pi}{16} \text{ rad}}^{+-}$	A1B2
		$\frac{7 \cdot \pi}{16} \text{ rad} 78.75^\circ$	→	$C_{\alpha=\frac{\pi}{8} \text{ rad } \beta=\frac{7 \cdot \pi}{16} \text{ rad}}^{+-}$	A1B3

Tabla G.2. Coincidencias correspondientes al parámetro S_1

Para el S_2 se tiene el juego de valores angulares utilizados:

S_2	$\alpha = \frac{\pi}{4} \text{ rad} 45^\circ$	β		Coincidencias	Nomenclatura simplificada
		$\frac{\pi}{16} \text{ rad} 11.25^\circ$	→	$C_{\alpha=\frac{\pi}{4} \text{ rad } \beta=\frac{\pi}{16} \text{ rad}}^{-+}$	A2B0
		$\frac{3 \cdot \pi}{16} \text{ rad} 33.75^\circ$	→	$C_{\alpha=\frac{\pi}{4} \text{ rad } \beta=\frac{3 \cdot \pi}{16} \text{ rad}}^{-+}$	A2B1
		$\frac{5 \cdot \pi}{16} \text{ rad} 56.25^\circ$	→	$C_{\alpha=\frac{\pi}{4} \text{ rad } \beta=\frac{5 \cdot \pi}{16} \text{ rad}}^{--}$	A2B2
		$\frac{7 \cdot \pi}{16} \text{ rad} 78.75^\circ$	→	$C_{\alpha=\frac{\pi}{4} \text{ rad } \beta=\frac{7 \cdot \pi}{16} \text{ rad}}^{--}$	A2B3

Tabla G.3. Coincidencias correspondientes al parámetro S_2

Para el S_3 se tiene el juego de valores angulares utilizados:

S_3	$\alpha = \frac{3\pi}{8} \text{ rad} \mid 67,5^\circ$	β		Coincidencias	Nomenclatura simplificada
		$\frac{\pi}{16} \text{ rad} \mid 11.25^\circ$	→	$C_{\alpha=\frac{3\pi}{8} \text{ rad } \beta=\frac{\pi}{16} \text{ rad}}^{-+}$	A3B0
		$\frac{3\pi}{16} \text{ rad} \mid 33.75^\circ$	→	$C_{\alpha=\frac{3\pi}{8} \text{ rad } \beta=\frac{3\pi}{16} \text{ rad}}^{-+}$	A3B1
		$\frac{5\pi}{16} \text{ rad} \mid 56.25^\circ$	→	$C_{\alpha=\frac{3\pi}{8} \text{ rad } \beta=\frac{5\pi}{16} \text{ rad}}^{--}$	A3B2
		$\frac{7\pi}{16} \text{ rad} \mid 78.75^\circ$	→	$C_{\alpha=\frac{3\pi}{8} \text{ rad } \beta=\frac{7\pi}{16} \text{ rad}}^{--}$	A3B3

Tabla G.4. Coincidencias correspondientes al parámetro S_3