



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
PROGRAMA DE MAESTRÍA Y DOCTORADO EN INGENIERÍA
ELÉCTRICA – INSTRUMENTACIÓN

Desarrollo de un sistema instrumental para el estudio de plasmas inducidos con láser y de los pulsos que lo producen.

TESIS
QUE PARA OPTAR POR EL GRADO DE:
MAESTRO EN INGENIERÍA

PRESENTA:
Ulises Vargas Nolasco

TUTORA PRINCIPAL
Dra. Citlali Sánchez Aké
Instituto de Ciencias Aplicadas y Tecnología

CIUDAD DE MÉXICO, SEPTIEMBRE, 2022



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

JURADO ASIGNADO:

Presidente: Dr. Garduño Mejía Jesús

Secretario: Dr. Guadarrama Santana Asur

1 er. Vocal: Dra. Sánchez Aké Citlali

2 do. Vocal: Dr. Pérez Alcázar Pablo Roberto

3 er. Vocal: Dr. Qureshi Naser

Esta tesis se realizó en el Instituto de Ciencias Aplicadas y Tecnología.
Ciudad Universitaria, UNAM.

TUTOR DE TESIS:
Dra. Citlali Sánchez Aké

NOMBRE



FIRMA

(Segunda hoja)

Agradecimientos

“La felicidad real siempre aparece escuálida por comparación con las compensaciones que ofrece la desdicha. Y, naturalmente, la estabilidad no es, ni con mucho, tan espectacular como la inestabilidad. Estar satisfecho de todo no posee el encanto que supone mantener una lucha justa contra la infelicidad, ni el pintorequismo del combate contra la tentación o contra una pasión fatal o una duda. La felicidad nunca tiene grandeza”

-Aldous Huxley, Un mundo feliz.

Muchas personas cercanas a mi han escuchado de mi voz citar la frase del libro de Aldous Huxley, “La felicidad nunca tiene grandeza”. Algunos al oír esta frase piensan en lo trágico que se escucha. Sin embargo, tiene un trasfondo mayor y es el de las adversidades que se atraviesan para llegar al objetivo que uno visualiza en la lejanía y el momento cumbre de lograrlo dichos objetivos. Esta frase recobra fuerza en mi mente al concluir estos dos años de la maestría y terminar mi tesis. De manera simultánea, me trae recuerdos de los sucesos por los que pase y las personas que estuvieron allí apoyándome, por consiguiente quiero agradecer con estos pequeños párrafos el que hayan formado parte de esta aventura.

En primera instancia quiero agradecer a mis padres y hermano, que durante la nueva etapa que inicie estuvieron siempre brindándome su apoyo, además de siempre brindarme su amor incondicional.

A la Dra. Citlali Sanchez Aké por aceptarme como su estudiante, darme orientación, por compartir sus conocimientos, por confiar en mis habilidades y permitirme desarrollar mi curiosidad.

Al Dr. Tupak García Fernández por tener la paciencia de enseñarme, apoyarme en varios aspectos de la realización de este trabajo y por fungir como mi segundo tutor.

A los miembros de mí jurado al Dr. Garduño Mejía Jesús, Dr. Guadarrama Santana Asur, Dr. Pérez Alcázar Pablo Roberto y al Dr. Qureshi Naser, por su atención, sus sugerencias y comentarios para mejorar este trabajo.

Al Consejo Nacional de Ciencia y Tecnología (CONACyT) por la beca otorgada para la realización de mis estudios de maestría, al Proyecto CONACYT-CB-A1-S-23112 y al Proyecto PAPIIT IN103621 financiado por DGAPA-UNAM.

A mis AMIGOS que me quieren y que quiero, que me brindan su apoyo y su hombro cuando lo necesito.

Por último y no menos importante, a Rosario, que llego en un momento importante de mi vida a compartirme su filosofía del mundo, cambiar paradigmas en mi mente y compartirme su esperanza con una sonrisa y muchas risas.

A mis padres y hermano.

Resumen

En este trabajo se desarrolla un sistema instrumental que permite el estudio de plasmas inducidos por pulsos de láser de nanosegundos, así como una técnica que permite medir parámetros del láser.

El sistema instrumental realiza dos tipos de mediciones que dependen de las interacciones de los pulsos de láser con el aire y la generación de plasma. Las interacciones en cuestión suceden en una posición ubicada entre dos placas de un capacitor que forma parte de un circuito eléctrico del sistema instrumental. Los plasmas formados son consecuencia de la absorción de parte de la energía de láser por átomos y moléculas del aire. Estos plasmas provocan perturbaciones que ocasionan un flujo de corriente a través de una resistencia del circuito, midiendo así caídas de voltaje se obtiene una señal eléctrica. Esto abarca el primer tipo de medición. Por otro parte, el segundo tipo de medida consistió en detectar simultáneamente la energía transmitida de los pulsos del láser, es decir, la energía que no fue absorbida, ni esparcida por el plasma. Este sistema instrumental se configuró de manera que realizara las mediciones de manera automatizada, con la finalidad de poder abarcar un mayor rango en cada uno de los parámetros involucrados en los experimentos. Los parámetros que se variaron fueron: energía de los pulsos del láser, voltaje aplicado a las placas y distancia de enfoque de los pulsos de láser (distancia focal de la lente convergente utilizada para producir plasma).

El sistema desarrollado en este trabajo permite medir la energía de los pulsos de láser mediante el análisis de la amplitud del primer pico de la señal eléctrica en función de la energía. El análisis se hizo en tres rangos de energía, [50,160] mJ, [180,340] mJ y [360,660] mJ, mostrando que el rango [50,160] mJ tiene mejores características para ser utilizado como medidor de energía, debido a que se tiene una relación monótona creciente y biunívoca. Además, del análisis de la amplitud de la señal eléctrica en función del voltaje aplicado a las placas se observó que la resolución de la medición mejora al aumentar el voltaje aplicado. Por otra parte, del análisis de la energía transmitida en función de la distancia focal de las lentes, se obtuvo que la transmitancia tiene un comportamiento creciente y monótono con respecto al cambio de la distancia focal. Así conociendo la energía transmitida se puede conocer la distancia focal de un lente. De igual forma, se obtuvo que sin importar los parámetros experimentales utilizados (energía de los pulsos del láser, voltaje aplicado a las placas y distancia focal) con la señal eléctrica se puede medir la duración de los pulsos de láser. Los resultados anteriores abren la posibilidad de desarrollar un sistema que permita medir la densidad de potencia en el punto de formación del plasma.

Por otro lado, se desarrolló una ecuación que relaciona la energía de los pulsos del láser con el voltaje. Esta ecuación depende del rango de energía en el que se esté trabajando y de la distancia focal utilizada. Sin embargo, es una herramienta útil y confiable para medir la energía de pulsos en rangos acotados.

Índice general

AGRADECIMIENTOS	3
RESUMEN	5
ÍNDICE GENERAL	6
INTRODUCCIÓN	9
OBJETIVOS	10
1 ESTADO DEL ARTE	11
1.1 LÁSER	11
1.1.1 ELEMENTOS DE UN LÁSER Y TIPOS DE LÁSERES	11
1.1.2 LÁSER Nd:YAG	12
1.1.3 PULSOS DE LÁSER Y LENTES	13
1.2 PLASMA	14
1.2.1 PLASMA INDUCIDO POR PULSOS DE LÁSER EN AIRE	14
1.2.2 MÉTODOS DE ESTUDIO DEL PLASMA	15
1.2.3 MÉTODOS ELÉCTRICOS	15
2 DESARROLLO EXPERIMENTAL	20
2.1 SISTEMA EXPERIMENTAL	20
2.1.1 SISTEMA DE GENERACIÓN DE PULSOS Y VARIACIÓN DE ENERGÍA	20
2.1.2 SISTEMA DE MEDICIÓN	22
2.2 PROGRAMACIÓN PARA LA AUTOMATIZACIÓN DEL SISTEMA EXPERIMENTAL	24
2.2.1 SISTEMA DE CALIBRACIÓN DE LA ENERGÍA DE LOS PULSOS	24
2.2.2 PROGRAMA GENERAL DE CONTROL DE LOS EXPERIMENTOS	26
2.2.2.1 Parámetros de los experimentos	26
2.2.2.2 Experimento	27
2.2.2.3 Programación	27
2.2.3 PROCESAMIENTO DE DATOS	29
2.2.3.1 Ruido	29
2.2.3.2 Procesamiento de señales y datos	30

3	MEDICIONES Y RESULTADOS	33
3.1	CALIBRACIÓN DE LA ENERGÍA DE LOS PULSOS DEL LÁSER	33
3.2	SEÑAL ELÉCTRICA Y SU RUIDO	35
3.3	VARIACIÓN DE ENERGÍA Y VOLTAJE	36
3.3.1	RANGO 50 MJ A 160 MJ	36
3.3.2	RANGO DE 180 MJ A 340 MJ	37
3.3.3	RANGO 360 MJ A 660 MJ	38
3.3.4	ECUACIÓN AMPLITUD (ENERGÍA, VOLTAJE)	39
3.4	DURACIÓN DEL PULSO	41
3.5	ÁREA DE ENFOQUE	43
4	CONCLUSIONES	46
5	TRABAJOS FUTUROS	47
6	REFERENCIAS	48
7	APÉNDICES	52
7.1	AUTOMATIZACIÓN	52
7.1.1	BUSES DE COMUNICACIÓN	53
7.1.1.1	GPIB	53
7.1.1.1.1	Tipos de dispositivos y topología de conexión	53
7.1.1.1.2	Mensajes y protocolo de transferencia	55
7.1.1.1.3	Comandos	57
7.1.1.1.4	Controlador NI – VISA y NI – 488.2	59
7.1.1.2	Bus Serial Universal (USB, Universal Serial Bus)	59
7.1.1.2.1	Definición y descripción general del <i>USB</i>	59
7.1.1.2.2	<i>Host – USB</i>	60
7.1.1.2.3	Controlador del host	61
7.1.1.2.4	<i>Hub - USB</i> y topología del <i>USB</i>	61
7.1.1.2.5	Arquitectura general del <i>USB</i> y modelo lógico funcional	62
7.2	MEDIDORES DE ENERGÍA	63
7.2.1	SENSORES PIROELÉCTRICOS	63
7.2.2	SENSORES CALORIMÉTRICOS	64
7.2.3	SENSORES ALTERNATIVOS	65
7.3	SISTEMA DE GENERACIÓN DE LOS PULSOS Y VARIACIÓN DE ENERGÍA	65
7.3.1	PROGRAMA DE CONTROL DEL LÁSER	66
7.3.2	PROGRAMA DE CONTROL DEL ATENUADOR DE ENERGÍA	67
7.4	SISTEMA DE MEDICIÓN: FUENTE DE ALTO VOLTAJE, MEDIDOR DE ENERGÍA Y OSCILOSCOPIO	67
7.4.1	PROGRAMA DE CONTROL DE LA FUENTE DE ALTO VOLTAJE	68
7.4.2	PROGRAMA DE CONTROL DEL MEDIDOR DE ENERGÍA	68
7.4.3	PROGRAMA DE CONTROL DEL OSCILOSCOPIO	68
7.5	COMANDOS FUNDAMENTALES UTILIZADOS EN LOS PROGRAMAS	69
7.5.1	COMANDOS DE MATLAB	70
7.5.2	COMANDOS DEL GENERADOR DE PULSOS	71

7.5.3	COMANDOS DEL ATENUADOR DE ENERGÍA	71
7.5.4	COMANDOS DE LA FUENTE DE ALTO VOLTAJE	71
7.5.5	COMANDOS DEL MEDIDOR DE ENERGÍA	72
7.5.6	COMANDOS DEL OSCILOSCOPIO	72
7.6	PROGRAMA DE CONTROL DEL LÁSER	74
7.7	PROGRAMA DE CONTROL DEL ATENUADOR WATT PILOT	75
7.8	PROGRAMA DE CONTROL DE LA FUENTE DE ALTO VOLTAJE	76
7.9	PROGRAMA DE CONTROL DEL MEDIDOR DE ENERGÍA	77
7.10	PROGRAMA DE CONTROL DEL OSCILOSCOPIO	78
7.11	PROGRAMA CALIBRACIÓN ENERGÍA	80
7.12	PROGRAMA DEL EXPERIMENTO	84
7.13	PROGRAMA ANÁLISIS DE SEÑALES Y ENERGÍA TRANSMITIDA	95
7.14	PROGRAMA DE INTEGRACIÓN	101
7.15	ANÁLISIS AMPLITUD(ENERGÍA,VOLTAJE)	102

Introducción

La formación de plasmas inducidos por pulsos de láser, "LIP" (por sus siglas en inglés "laser induced plasmas") ha tenido un impacto en la investigación y el desarrollo tecnológico. De manera paralela al desarrollo de láseres, y debido a las diversas investigaciones y aplicaciones, se hizo importante conocer la energía, la potencia y la densidad de potencia de la radiación láser. Esto impulsó el desarrollo de medidores de potencia y energía.

En 1960, se reportó la operación del primer láser en un cristal de rubí.¹ Tres años después, se observó que enfocando el láser de rubí, se produce una chispa.^[1-3] En ese mismo año nace una de las técnicas espectroscópicas más poderosas: la técnica de espectroscopia de rompimiento inducida por láser, "LIBS" (por sus siglas en inglés "*Laser-induced breakdown spectroscopy*"). LIBS permite la detección y medición de especies elementales a través del análisis espectral de la luz emitida por el LIP.^[2,4] Para el estudio del LIP también se utilizan otras técnicas que involucran fenómenos ópticos, como el cambio en el índice de refracción del medio donde se produce el plasma, que se estudia a través de la técnica de sombras ("*Shadowgraphy*").^[5,6] También se utiliza el cambio de fase que permite medir la densidad de electrones y la temperatura del gas, a través de interferometría.^[5,6] Otra técnica que permite estudiar los componentes de la muestra, los valores cuantitativos de la temperatura y la densidad electrónica es la espectroscopia de emisión resuelta en tiempo.^[5,6]

En este trabajo se empleó un método alternativo para caracterizar LIPs, un diagnóstico eléctrico. El procedimiento consistió en utilizar un circuito RC y un medidor de energía piroeléctrico para estudiar plasmas inducidos por pulsos de láser. El plasma es producido por un pulso de láser enfocado en el aire entre un par de placas paralelas (capacitor), donde estas producen un campo eléctrico uniforme. El campo eléctrico se ve perturbado por la generación del plasma y produce un flujo de cargas que pasan por la resistencia y se mide como una caída de voltaje. Por otro lado, parte de la energía del láser que no es utilizada para la formación del plasma es captada por el medidor de energía colocado en la dirección de propagación del haz (energía transmitida).

La intención de este trabajo es ampliar el estudio de LIPs haciendo uso de un sistema automatizado que nos permita realizar mayor número de medidas para diferentes variables experimentales. Los parámetros variados en este trabajo serán: la energía de los pulsos láser, el voltaje aplicado a las placas y la distancia de enfoque de los pulsos de láser.

Objetivos

Objetivo general:

Desarrollar un sistema instrumental que permita estudiar plasmas producidos por pulsos de láser y que además sirva como medidor de parámetros de los pulsos del láser, energía y su duración temporal.

Objetivos particulares:

1. Desarrollar el sistema instrumental que realiza de dos tipos de análisis. El primero realiza la medición del voltaje (“señal eléctrica”) inducido a través de una resistencia de un circuito eléctrico, sensor RC, que es consecuencia de la formación del plasma en el aire entre el par de placas paralelas. El segundo realiza la medición de la energía de los pulsos del láser que es transmitida y detectada por un sensor optoelectrónico.
2. Automatizar este sistema instrumental, haciendo uso del sistema de cómputo numérico MATLAB, de manera que sea posible controlar sus diferentes partes, así como para la adquisición, transferencia y procesamiento de los datos obtenidos de las señales estudiadas.
3. Estudiar el efecto de la energía de los pulsos de láser en el comportamiento de la señal eléctrica en el sensor RC, de manera que posteriormente sirva para el proceso inverso, es decir, para utilizar el sistema instrumental como medidor de energía de los pulsos de láser.
4. Estudiar el efecto del voltaje entre el par de placas del capacitor del sensor RC en el comportamiento de la señal eléctrica.
5. Estudiar el efecto que provoca, en el comportamiento de la señal eléctrica y la energía transmitida, la variación de la distancia focal de las lentes utilizadas para producir el plasma en el aire.
6. Mostrar que la evolución temporal de la integral de la señal eléctrica puede servir para describir la duración del pulsos del láser, bajo las condiciones experimentales aquí utilizadas.
7. Encontrar relaciones matemáticas que describan el comportamiento de algunas de las mencionadas dependencias.

1 Estado del arte

1.1 Láser

El término láser es un acrónimo que proviene del inglés *Light Amplification by Stimulated Emission of Radiation* (amplificación de luz por emisión estimulada de radiación).⁷ El láser es un dispositivo que, como bien lo especifica su nombre, utiliza la emisión estimulada de la radiación en un medio para generar un haz de luz (cuantos de energía o fotones) cuyas características especiales se encuentran controladas. Algunas de estas características de la radiación del láser son: la monocromaticidad, la coherencia (espacial y temporal), la polarización y la direccionalidad. Si tenemos cuantos de luz, la monocromaticidad se refiere a que todos estos cuantos tienen la misma energía, que equivale a tener la misma frecuencia de oscilación de los campos eléctricos y magnéticos. La coherencia espacial y la coherencia temporal indican cuanto se pueden mantener en fase entre sí los cuantos de luz tanto en el espacio como en el tiempo. La direccionalidad se refiere a la dirección en la que se propagan los cuantos de luz, que en muchos casos significa una propagación de la luz que no diverge de forma importante.^[7,8]

1.1.1 Elementos de un láser y tipos de láseres

Para la construcción de los láseres se consideran tres elementos básicos: el medio activo, el sistema de bombeo y la cavidad óptica. El medio activo es el material donde se produce la emisión estimulada. El sistema de bombeo consiste en suministrar de energía al medio activo para producir la emisión estimulada. Por último, la cavidad óptica es un dispositivo que mantiene la luz circulando a través del medio activo para poder amplificarla, permitiendo que una parte de esta luz salga de la misma. Usualmente, los sistemas de bombeo más comunes son: el bombeo óptico y el bombeo por descarga eléctrica.^[7,8]

Los láseres pueden clasificarse en función de varias de sus características. Una forma de clasificación es por su medio activo, pudiendo diferenciarse en 4 grupos principales:⁸

- Láser de estado sólido: utilizan típicamente cristales o vidrios como medios activos, usualmente están dopados con tierras raras. El láser que se utilizó en este trabajo es de esta categoría.
- Láser de semiconductor: El medio activo consiste en una unión semiconductor tipo diodo emisor de luz (*LED* por sus siglas en inglés *light-emitting diode*) alojado en una cavidad reflectora que permite la oscilación de la luz.
- Láser de gas: Estos láseres utilizan un gas o una mezcla de gases como medio activo, usualmente se encuentran confinados en un tubo de vidrio.
- Láser de colorante: el medio activo utilizado en estos láseres es una disolución líquida de compuestos orgánicos colorantes en algún líquido.

Otra forma de clasificación es según la longitud de onda de emisión, existiendo láseres con emisión en el infrarrojo, en el visible y en el ultravioleta. Además, se pueden clasificar según

la duración de la emisión, existiendo láseres con emisión continua y pulsada. Incluso, se pueden diferenciar según la duración de los pulsos, por ejemplo, láseres de nanosegundos y femtosegundos. De igual manera, se suelen clasificar según la potencia de la emisión, por ejemplo, de alta y baja potencia. También, se pueden diferenciar ya sea por potencia media o por potencia pico, por ejemplo, un láser pulsado con 100 mJ de energía por pulso y 10 ns de duración, trabajando, con un repetición de disparo de 10 Hz tendría una potencia media de 1 W y un potencia pico de 10^7 W. Existen otros tipos de posibles clasificaciones. Conviene recordar que la potencia se define como la energía entregada (**Energía**) dividida entre el tiempo de entrega (**tiempo**). En cambio, la densidad de potencia (**DP**) se define como la potencia entregada entre el área o volumen de incidencia del haz del láser.

$$\text{Potencia} \quad P = \frac{\text{Energía}}{\text{tiempo}} \quad (1-1)$$

$$\text{Densidad de potencia} \quad DP = \frac{P}{\text{área}} \quad \text{ó} \quad DP = \frac{P}{\text{volumen}} \quad (1-2)$$

1.1.2 Láser Nd:YAG

El tipo de láser Nd:YAG, como el utilizado en este trabajo, es un láser de estado sólido cuyo medio activo consiste en una barra de granate óxido de itrio y aluminio (YAG) dopado con neodimio (Nd). Esta es una barra que puede absorber la emisión de una o más lámparas flash o diodos láser que cumple la función del sistema de bombeo. El bombeo excita al medio activo para la generación de la radiación láser. La cavidad está compuesta por dos espejos, uno parcialmente reflejante y otro altamente reflejante, que hacen la función de un resonador. Entre estos espejos se encuentra un celda de Pockels (sistema Q-switch), la cual contiene un cristal líquido, que al aplicar un voltaje cambia el índice de refracción, lo que permite junto a un sistema de polarización modular el paso de la luz en el resonador (ver Figura 1.1). Este sistema láser funciona de manera pulsada emitiendo a una longitud de onda de 1064 nm y con una duración de los pulsos en el rango de los nanosegundos. Este tipo de equipos suelen emitir pulsos con energías que pueden oscilar entre unos cuantos mJ y varios J.^[7,9]

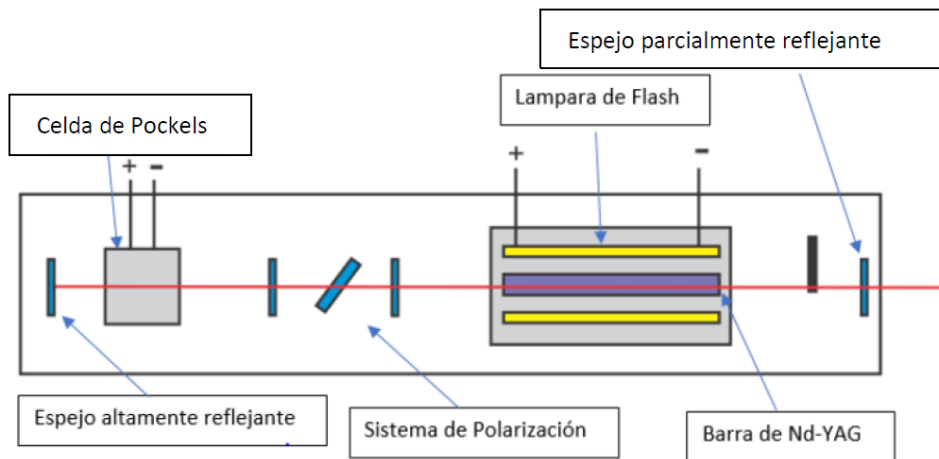


Figura 1.1. Esquema de un láser Nd:YAG. Figura tomada de un trabajo de Rodriguez.⁹

1.1.3 Pulsos de láser y lentes

En este trabajo una de las variables principales utilizadas es la distancia focal de las lentes que permiten la focalización de los pulsos de láser. La Figura 1.2 muestra una representación de este proceso. En nuestro caso se utiliza un láser con haz gaussiano ⁷, de modo que para calcular las características de focalización se puede utilizar las ecuaciones de lentes delgadas¹⁰ para este tipo de haz. Esta versión de la ecuación de la lente delgada que tenía en cuenta la propagación gaussiana fue desarrollada por Sidney Self en 1983.¹⁰ Para ello se realizan ciertas consideraciones, la cintura del haz de entrada (w_0) se considera como el objeto y la cintura del haz de salida (w'_0) como la imagen. Entre estas ecuaciones se tiene la que describe la cintura del haz (cintura de salida) en el punto de focalización w'_0 dependiendo de la cintura de entrada w_0 , de la distancia desde esta posición hasta la lente s , la distancia focal de la lente f y el rango de Rayleigh z_R de la cintura de entrada que es un parámetro clave para describir la propagación de los haces gaussianos cerca de los puntos focales y viene definido para este caso como:

$$z_R = \frac{\pi w_0^2}{\lambda}, \quad (1-3)$$

donde λ la longitud de onda del haz gaussiano.

Así, la ecuación de la cintura en la posición de focalización es:

$$w'_0 = \frac{f}{\sqrt{(|s| - f)^2 + z_R^2}} w_0, \quad (1-4)$$

En este trabajo $w_0 = 3.5$ mm, $\lambda = 1064$ nm y las distancias focales están en el rango de $f = [75,500]$ mm. Ahora considerando que $|s| \sim f$, entonces $|s| - f \sim 0$ por lo que la ecuación (1-4) queda:

$$w'_0 = \frac{f}{z_R} w_0 = \frac{\lambda f}{\pi w_0} \quad (1-5)$$

Se observa que $w'_0 \propto f$ y dado que el área de enfoque del haz puede ser considerado como el área de una circunferencia con radio w'_0 , $A = \pi w_0'^2$, entonces se puede decir que $A \propto f^2$.

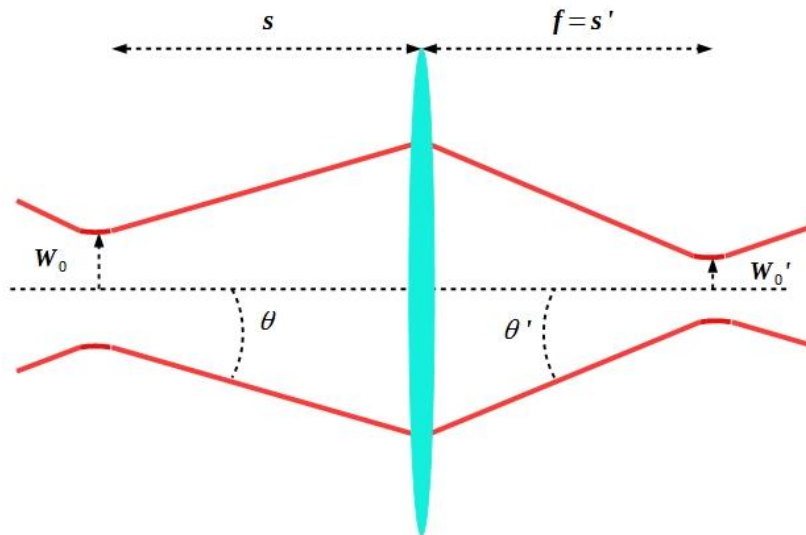


Figura 1.2. El "objeto" al reenfocar un haz gaussiano es la cintura de entrada y la "imagen" es la cintura de salida.

Por otra parte, como se presentó en la ecuación (1-2) la densidad de potencia es inversamente proporcional al área, $DP \propto \frac{1}{A}$, lo que implica que $DP \propto \frac{1}{f^2}$. De esta manera, se puede considerar que para la lente con menor distancia focal utilizada en nuestro trabajo $f_1 = 75$ mm se tiene un densidad de potencia de aproximadamente 44 veces mayor que para la lente con mayor distancia focal $f_2 = 500$. Debe considerarse que la densidad de potencia es posiblemente el parámetro fundamental que define la formación del plasma.

1.2 Plasma

En la actualidad se cree que la composición del universo está conformada por 69% energía oscura, 27% materia oscura y 4% de materia normal. De la materia normal el 99% está en estado de plasma, que es como se le conoce al cuarto estado de la materia. El 1% se encuentra en estado sólido, líquido o gas.² Se dice de forma coloquial que un plasma es un gas ionizado, pero en general los gases presentan cierto grado de ionización. Una definición común es:

*Un plasma es un gas cuasi-neutral de partículas neutras y cargadas, que exhibe un comportamiento colectivo.*²

Consideremos las moléculas del aire y las fuerzas que actúan sobre dichas moléculas. Dado que la molécula es neutra, no hay una fuerza electromagnética neta y se puede despreciar la fuerza que produce el campo gravitacional. Las moléculas de aire se mueven sin perturbaciones hasta que chocan con otra molécula. Ahora, si a un gas neutro se le aplica una fuerza macroscópica, por ejemplo, ondas sonoras por medio de un altavoz, esta fuerza se transmitirá por medio de colisiones. Sin embargo, en un plasma es distinto, las partículas cargadas al moverse pueden generar concentraciones de cargas eléctricas positivas y negativas, que interactúan mediante la fuerza coulombiana con otras concentraciones de cargas ubicadas en otra región del plasma. Esto resulta en la generación de campos eléctricos y magnéticos. Estos campos afectan el movimiento de otras partículas cargadas lejanas en otra región del plasma.² Dado lo anterior, se refiere por comportamiento colectivo, a los movimientos de partículas que dependen no solo de las condiciones locales (interacción de dos partículas de los alrededores) sino también al movimiento de cada una de estas partículas del plasma en regiones remotas.

Al extraer o adicionar uno o más electrones de un átomo, resulta en una escasez o exceso de cargas negativas, a un átomo con estas características se le denomina ion. Entonces, un gas ionizado que pierde electrones y que los mantiene en sus cercanías puede ser un plasma. La neutralidad del plasma indica que $n_i \simeq n_e \simeq n$, donde n_i la densidad iónica, n_e la densidad electrónica y n una densidad llamada densidad del plasma, pero no tan neutro para que las interacciones electromagnéticas pierdan importancia.²

Este concepto contemporáneo de plasma data de inicios de los años cincuenta.

1.2.1 Plasma inducido por pulsos de láser en aire

Si en un volumen milimétrico de un medio gaseoso se enfoca un pulso láser, el gas puede perder su propiedad aislante y puede pasar a ser conductor. Esto debido a que los campos eléctricos del láser pueden ser muy intensos (por ejemplo, del orden de 10^6 a 10^7 V/cm⁻¹)¹ y rebasar el valor límite de ruptura dieléctrica del aire. Como resultado ocurre lo que llamamos rompimiento del gas. Este rompimiento produce una pequeña chispa de temperatura elevada y densidad alta de partículas ($T > 10^4$ K $n > 10^{20}$ partículas/cm³).^[11-13]

Para producir un plasma en aire a una presión de 580 torr (presión similar a la de la CD. MX.) se necesita que la intensidad del láser sea del orden de 10^9 W/cm² o superiores.¹

La ruptura tiene una dependencia con los siguientes parámetros experimentales: duración del pulso láser, longitud de onda, distancia de enfoque del pulso (distancia focal de la lente), entre otros.¹⁴

Para el proceso de ruptura del gas tienen que suceder dos etapas:

- En las inmediaciones del volumen focal del pulso deben encontrarse unos cuantos electrones (por lo menos uno).
- Crear una avalancha de ionización en la región focal.

Para crear la avalancha de ionización existen dos procesos diferentes que dependen esencialmente de las densidades de energía del láser que se utilicen: el bremsstrahlung inverso y la ionización multifotónica. Aunque, para la ionización multifotónica este último no es necesario que haya un electrón en las inmediaciones de la región focal.¹⁴

1.2.2 Métodos de estudio del plasma

La generación de los plasmas inducidos por pulsos laser, llevó consigo el interés de conocer los procesos y fenómenos que de él se desprenden. Esto condujo al desarrollo de nuevos láseres y tecnologías para la observación y detección de los fenómenos producidos por los plasmas.

La técnica de espectroscopia de rompimiento inducida por láser, “LIBS” (por sus siglas en inglés “*Laser-induced breakdown spectroscopy*”), es considerada una de las técnicas más importantes. Esto debido a que ha logrado hitos de gran importancia en diferentes campos de investigación. En el año de 1966, se analizó metal fundido directamente con el chispazo láser; posteriormente en 1980, LIBS es utilizado en la industria de la energía nuclear; en 1992, se utiliza para detectar metales en suelos; el método es implementado en el rover marciano Curiosity.¹ LIBS permite la detección y medición de especies elementales a través del análisis espectral de la luz emitida por el plasma.^[2,4] La técnica, espectroscopia de emisión resuelta en tiempo permite estudiar los componentes de la muestra, los valores cuantitativos de la temperatura y la densidad electrónica.⁵ Para realizar el estudio del fenómeno de cambio de índice de refracción en el plasma, se usa la técnica de sombras (“Shadowgraphy”).⁵ Por otro lado, utilizando el cambio de fase se mide la densidad de electrones y la temperatura del gas, a través de la interferometría. El efecto fotoacústico es otro fenómeno inmerso en la generación de los plasmas inducidos por láser. Este efecto se produce cuando el pulso de láser interactúa con la materia, produciendo ondas ultrasónicas. Esta técnica basada en el efecto fotoacústico, permite obtener información sobre propiedades termoelásticas de los materiales.^[5,15]

Por otro lado, durante la formación del plasma se observan otros fenómenos particulares, como: la variación espacio-temporal del potencial del campo eléctrico y los campos magnéticos autogenerados.¹³ Para realizar el estudio de estos fenómenos se utilizan sondas de cable coaxial y sondas B[·] (bobina inductiva de 4 vueltas).^[14,16,17]

1.2.3 Métodos eléctricos

En numerosos trabajos se ha reportado el estudio de plasmas inducidos por láser bajo la presencia de campos eléctricos.^{[18]–[25]} En 2002, Bredice et. al.²⁶ desarrolló un método similar a la espectroscopia optogalvánica. El método consistía en generar plasma en el aire por la acción de pulsos de láser que incidían entre un par de placas paralelas de un condensador C1 (ver Figura 1.3), cuya diferencia de potencial (V) producía un campo eléctrico. El chispazo entre las placas produce una perturbación en este campo eléctrico. Esta perturbación se puede medir como una señal eléctrica que refleja la diferencia potencial que se induce en una resistencia R conectada en paralelo con un condensador C2 a la placa de tierra. La amplitud máxima de esta señal eléctrica (**h**) guarda una estrecha

relación con los siguientes parámetros: la energía de ruptura (E_b), la tensión utilizada entre las placas (V), la distancia entre las placas (d) y una constante de proporcionalidad (k_c) que depende de los parámetros del circuito ²⁶

$$h = k_c E_b \frac{V}{d^{2.8}} \quad (1-6)$$

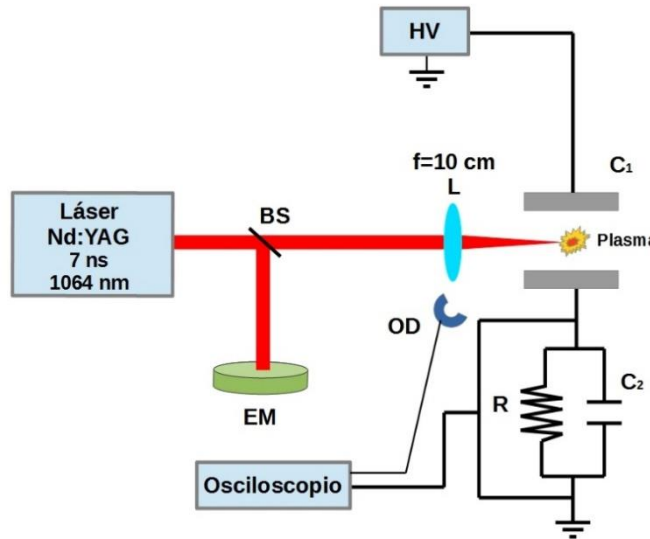


Figura 1.3. Arreglo experimental utilizado por Bredice et al. ²⁶, para analizar la perturbación generada por el chispazo producido por los pulsos de láser, bajo la presencia de un campo eléctrico dc. EM: medidor de energía, HV: fuente de alto voltaje, OD: fotodiodo, BS: divisor de haz, R: resistencia, C₁ y C₂ capacitores. Imagen tomada de mi tesis de licenciatura ⁶.

Este método ha sido utilizado por el grupo del Laboratorio de Fotofísica y Películas Delgadas del Instituto de Ciencias Aplicadas y Tecnología (ICAT) de la Universidad Nacional Autónoma de México (UNAM), para realizar diversos trabajos. En el 2005, con este método se estudió la ablación láser de metales. Se enfocó un pulso de láser sobre la superficie de un blanco metálico en aire. Para el análisis del plasma se usó el método eléctrico con alguna variante (ver Figura 1.4).

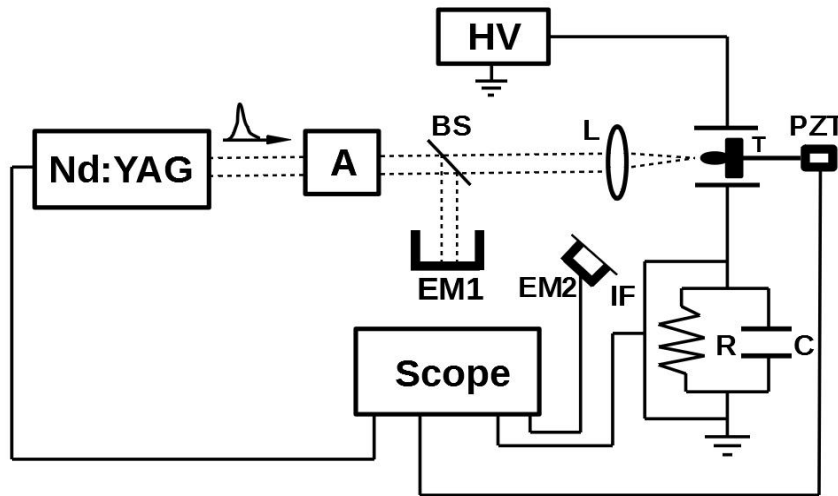


Figura 1.4. Arreglo experimental usado para analizar la energía utilizada en la ablación láser en blancos sólidos. A: atenuador, BS: divisor de haz, EM: medidor de energía, IF: Filtro interferencial, T: blanco, L: lente, PZT: sensor piezoeléctrico, HV: fuente de alto voltaje, R: resistencia, C: capacitor. Imagen tomada del trabajo de Sobral et. al.²⁷

Detrás del blanco metálico se colocó un sensor piezoeléctrico, utilizado para registrar una señal fotoacústica. Para medir la energía esparcida se utilizó un filtro interferencial conectado a un medidor de energía. Se obtuvo un balance de la energía utilizada en la generación del plasma, a partir del análisis de las señales. Por otro lado, la señal eléctrica se utilizó para optimizar el punto de ablación en los blancos metálicos.²⁷

En 2007²⁸, se enfocó un pulso láser en aire y en blancos de cobre, se realizó el análisis de la perturbación del plasma mediante el método desarrollado por Bredice et. al. De manera paralela, utilizando un sistema experimental como el que se muestra en la Figura 1.5 se midió el ensanchamiento Stark de las líneas H_{α} . Se encontró una relación unívoca de la evolución temporal del ensanchamiento Stark de la línea H_{α} y la amplitud de la señal eléctrica. Por cuanto se conoce la relación entre el ensanchamiento Stark de las líneas H_{α} y la densidad de electrones del plasma, este método se puede usar para predecir la evolución temporal de esta última.

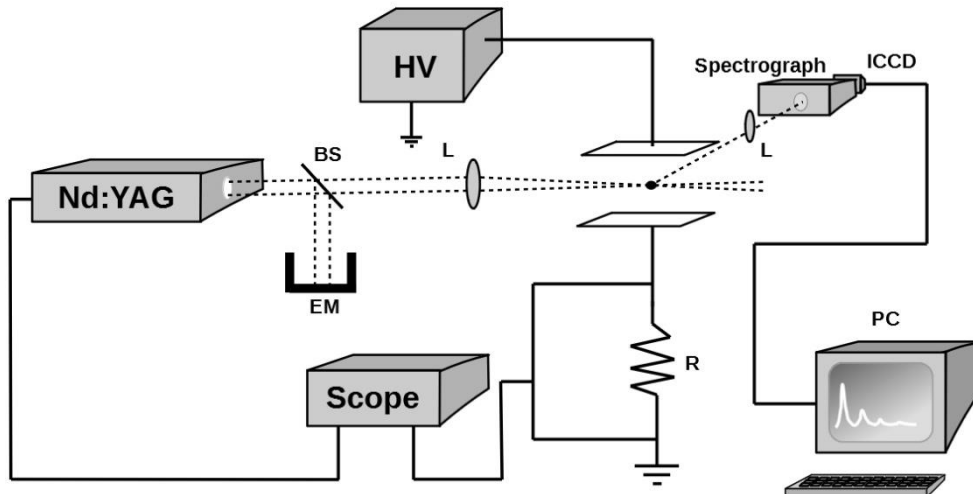


Figura 1.5. Arreglo experimental para medir las líneas H_{α} y la señal eléctrica. BS: divisor de haz, HV: fuente de alto voltaje, EM: medidor de energía, L: lente, R: resistencia. Imagen tomada de un trabajo de Bredice et. al.²⁸

En el 2018, Pacheco et al.²⁹, llevaron a cabo experimentos analizando el fenómeno de ablación láser en metales mediante la técnica antes mencionada. Se utilizó un arreglo

Se logró obtener la interpretación física de las señales estudiadas. Está es la parte más importante del trabajo realizado. Se encontró que el voltaje medido a través del resistor mide el cambio del flujo de campo eléctrico. La siguiente ecuación muestra lo anterior:

$$V_{R2} = \epsilon_0 \frac{d\phi_{ep}}{dt} R2 , \quad (1-8)$$

donde V_{R2} es el voltaje medido en la resistencia $R2$, ϵ_0 es la permitividad eléctrica del vacío y ϕ_{ep} es el flujo del campo eléctrico. Al integrar la ecuación anterior se obtuvo lo siguiente:

$$\Delta\phi_{ep} = \frac{1}{\epsilon_0 R2} \int_{-\infty}^y V_{R2} dt . \quad (1-9)$$

Se mostró que el primer pico de la integral de la señal eléctrica en el tiempo, que corresponde al cambio del flujo del campo eléctrico, coincide con la evolución temporal del pulso de láser (ver Figura 1.7). Dado el resultado anterior se llegó a la conclusión, que se puede utilizar la integral de la señal eléctrica para conocer la duración de los pulsos de láser.

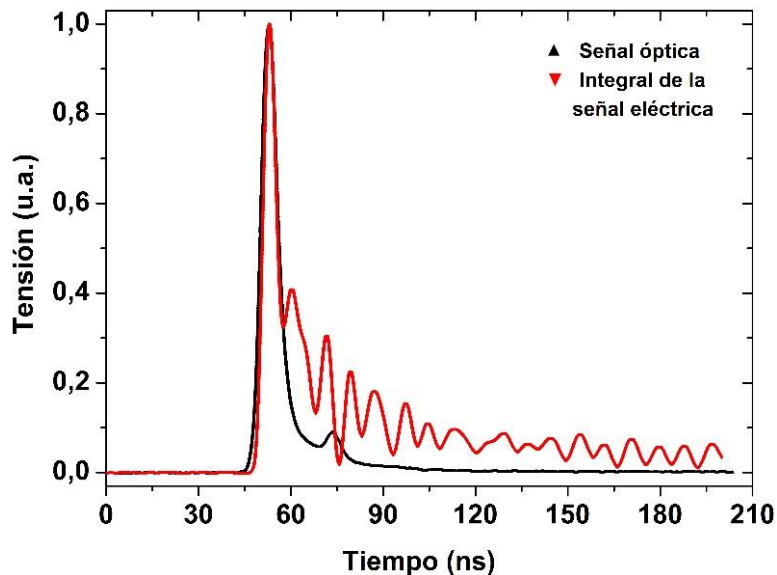


Figura 1.7. Señal óptica (curva negra) e integral de la señal eléctrica (curva roja). Las señales se encuentran normalizadas. Energía de los pulsos: 120 mJ, longitud de onda: 1064 nm, distancia entre placas: 35 mm y tensión entre placas: 1000 V. Imagen tomada de mi tesis de licenciatura ⁶.

2 Desarrollo experimental

Los experimentos involucrados en este trabajo consistieron en estudiar plasmas producidos en aire por la acción de pulsos de láser. Cambios en los procesos de formación y evolución de estos plasmas se lograron variando la energía y el enfoque de los pulsos del láser. Para el estudio de estos procesos se desarrolló un sistema instrumental que además sirve como medidor de parámetros de los pulsos del láser. Este sistema instrumental se basa en la realización de dos tipos de mediciones, que dependen de las características de las interacciones de los pulsos del láser con el aire. Estas interacciones se localizan en una posición entre dos placas de un capacitor que forma parte de un circuito eléctrico del sistema instrumental. La formación de los plasmas es consecuencia de la absorción por el aire de parte de la energía del láser. De esta manera, el primer tipo de medición consiste en realizar mediciones de caídas de voltaje en una resistencia del circuito (señales eléctricas), consecuencia de las perturbaciones que provocan en el capacitor la generación del plasma. Por otra parte, el segundo tipo de medición consiste en medir la energía transmitida de los pulsos del láser.

A continuación, se presenta la descripción de los experimentos dividiendo el capítulo en dos secciones: sistema experimental y la programación del sistema experimental.

2.1 Sistema experimental

Para lograr los fines de este trabajo se desarrolló un sistema experimental como el que se muestra en la Figura 2.1, el cual se adaptó para su utilización en tres rangos de energía, [360,660] mJ **(a)**, [180,340] mJ **(b)** y [50,160] mJ **(c)**. Fue necesario realizar esta división debido a que el atenuador de energía no permite reducir todo el rango de energía utilizado en este trabajo, que fue de hasta 660 mJ, sin que el haz saliente del atenuador sea modificado.

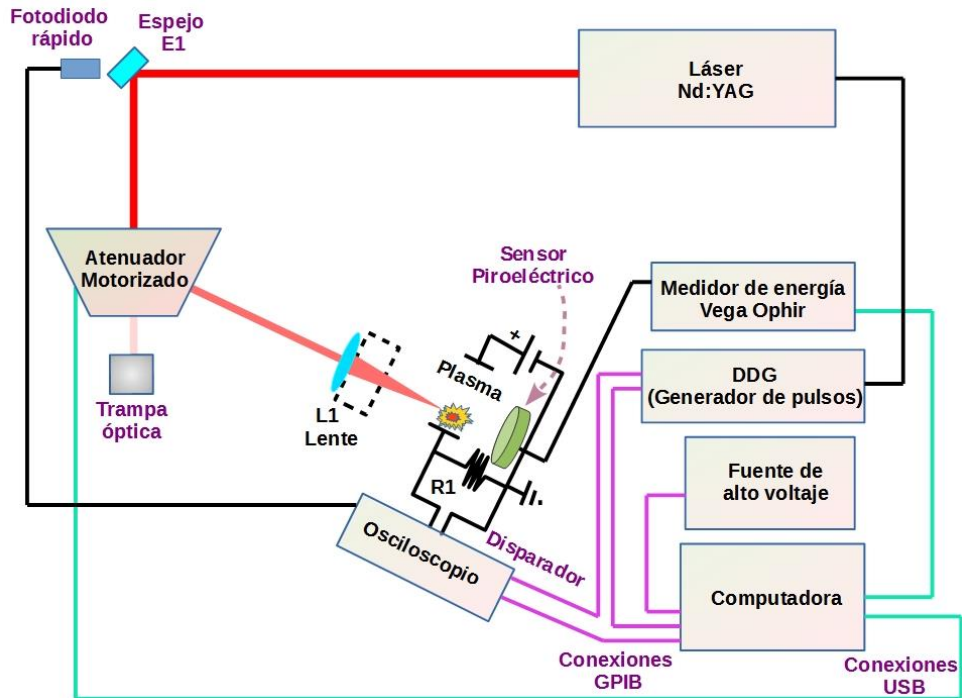
Para facilitar la descripción del arreglo experimental, éste se desglosó en dos partes: el sistema de generación de pulsos y variación de energía y el sistema de medición. Cada uno de estos sistemas fue automatizado y controlado por una PC.

2.1.1 Sistema de generación de pulsos y variación de energía

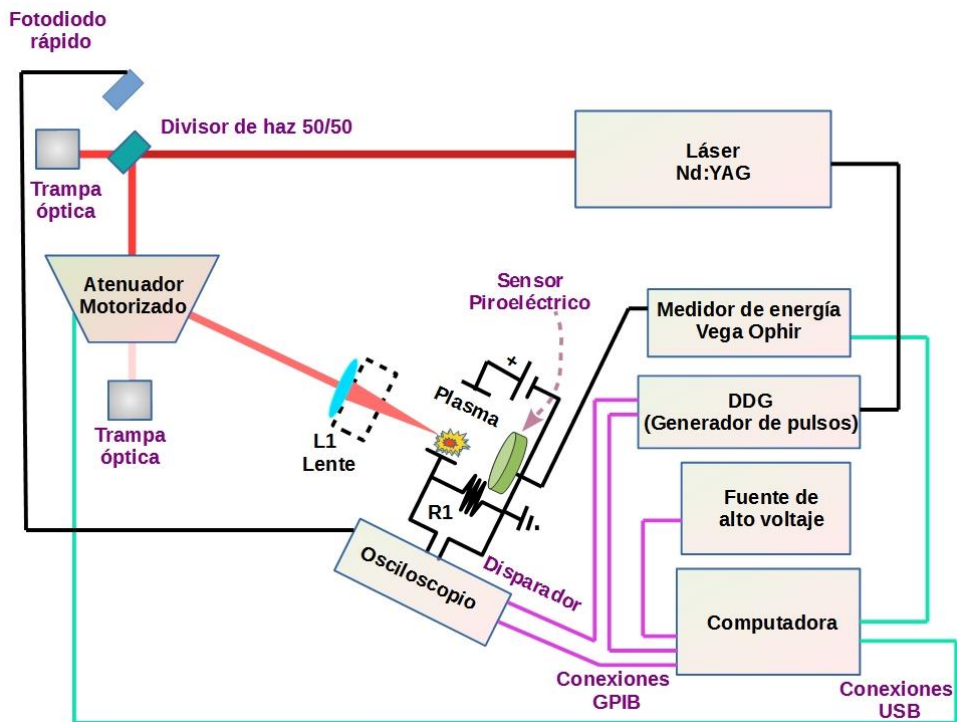
La función de este sistema como su nombre lo indica, es la de generar los pulsos láser así como variar su energía, que es una de las variables de este trabajo. Como fuente de emisión de este sistema se utilizó un láser Nd:YAG, modelo Surelite III-10 de la firma Continuum. Los parámetros usuales del láser y con los que se trabajó fueron: $\lambda = 1064$ nm, V-lamp (voltaje de la lámpara flash) = 1.47 kV, Q-switch = 230 μ s y frecuencia de los pulsos = 10 Hz. Dando las siguientes características a los pulsos del láser: duración de pulso de 6-8 ns y energías de los pulsos de hasta 800 mJ. Los disparos del láser fueron controlados mediante un generador de pulsos, modelo DG645, marca Stanford Research Systems. El generador de pulsos se conectó al láser Nd:YAG mediante un par de cables BNC hembra-

macho. Para el rango de energía de [360,660] mJ, se colocó un espejo E1 que permitió cambiar la dirección del haz hacia el atenuador motorizado Watt Pilot de la marca Altechna.

a)



b)



c)

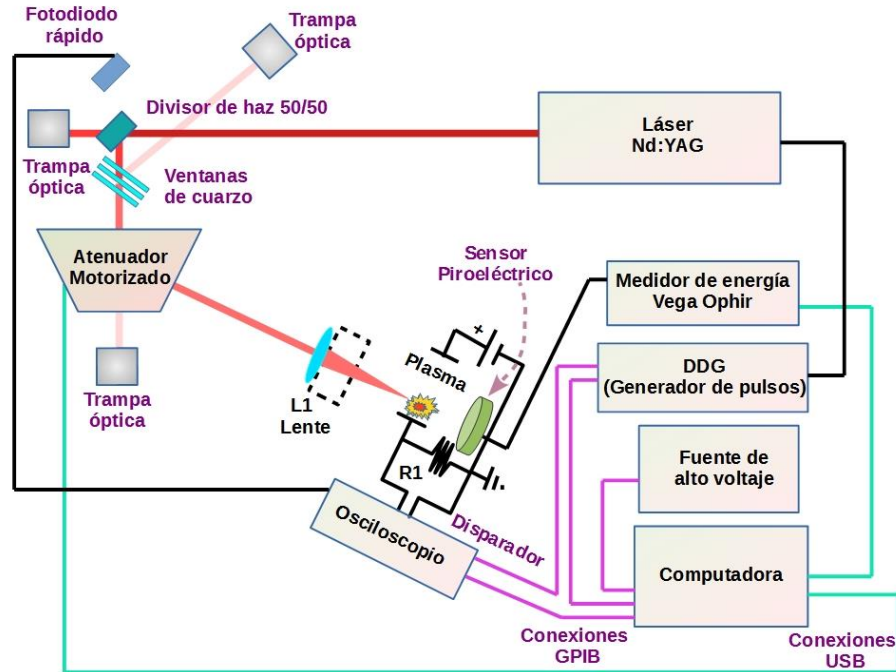


Figura 2.1. Arreglo experimental para los rangos de energía de los pulsos láser: a) [360,660] mJ, b) [180,340] mJ y c) [50,160] mJ.

Para los rangos de [50,160] mJ y [180,340] mJ, en lugar del espejo se colocó un divisor de haz 50/50 (ver Figura 2.1 b) y c)), de manera que no solo cambia la dirección de los pulsos del láser sino que también reduce a la mitad la energía. Dado que el divisor parte en dos el haz de luz, uno en dirección hacia el atenuador y el otro sigue de largo, se colocó una trampa óptica detrás de dicho divisor. Por otro lado, para el rango de energía de [50,160] mJ, después del divisor se colocó un conjunto de ventanas de cuarzo para reducir aún más la energía que llega al atenuador (ver Figura 2.1 c)). Así, el atenuador permitió controlar el rango deseado de energía de los pulsos. El atenuador de energía watt Pilot tiene un funcionamiento particular, utiliza una placa de onda de fase lambda medios de cuarzo giratorio y un polarizador, separando el haz en dos. De este modo al rotar la placa se incrementa o disminuye la intensidad del haz. Uno de los haces que sale del atenuador se dirigió a una lente de cuarzo L1 convergente, planoconvexo de una pulgada de diámetro. Para reducir la posible formación de plasma debido a la reflexión de la radiación en la lente, la parte convexa de la lente se colocó en dirección hacia el atenuador. Para el desarrollo del trabajo se utilizaron nueve lentes con diferentes distancias focales (f : 75 mm, 100 mm, 125 mm, 150 mm, 200 mm, 250 mm, 300 mm, 400 mm y 500 mm). El cambio de la distancia focal también permite modificar la densidad de potencia de la radiación incidente, de la cual depende la formación y evolución del plasma en el aire.

2.1.2 Sistema de medición

La función de este sistema es el de realizar las diferentes mediciones que se llevaron a cabo en este trabajo, que consistieron en la medición de la señal eléctrica; de la energía incidente y la transmitida del pulso láser; y la evolución temporal del mismo.

Para llevar a cabo la medición de la evolución temporal se utilizó un fotodiodo de respuesta rápida (tiempo de subida de 1 ns), modelo DET10A marca Thorlabs. Para el rango de

energías de [360, 660] mJ el fotodiodo se colocó detrás del espejo E1, en cambio para los rangos de [50,160] mJ y [180,340] mJ el fotodiodo se colocó a un lado apuntando a la trampa óptica como se observa en la Figura 2.1 **c)** y **b)** respectivamente. El fotodiodo se conectó mediante un cable BNC – BNC a un osciloscopio de la marca Tektronix, modelo DPO 7054, con una impedancia de entrada de 50 Ω .

Para realizar la medición de la señal eléctrica se utilizó un circuito que aquí llamamos “sensor eléctrico” que consistió en una fuente de alto voltaje marca Stanford Research Systems, modelo PS350 con un rango de voltaje de [35,5000] V; una resistencia R1 de 1 (0.05) Ω (un alambre de aluminio, con un grosor de 0.7 (0.05) mm y una longitud de 89 (0.05) mm); y un par de placas paralelas de aluminio, con un diámetro cada una de 5 (0.005) cm y una distancia entre ellas de 35 (0.05) mm que hacen la función de capacitor (la forma en que se reporta las incertidumbres se justifica en un trabajo de Miranda³⁰ y en un trabajo de la Universidad de Pensilvania³¹). Todos estos elementos estuvieron conectados en serie. La señal eléctrica se obtuvo conectando el osciloscopio a través de un cable BNC a las dos terminales de la resistencia R1, incluyendo la que va a tierra (ver Figura 2.1). Debido a que resultó más fácil mover este sistema que realinear los elementos del sistema de generación de pulsos y variación de energía, el par de placas se colocaron de manera tal que la posición de enfoque del haz del láser se encontraba entre ellas. La distancia del plasma a la placa de tierra (A) se fijó en 7 (0.05) mm (ver Figura 2.2). Se tomó de esta manera tomando en cuenta los resultados de mi tesis de licenciatura⁶. El tamaño de las placas se mantuvo constante, de manera que no afecta a la amplitud del primer pico de la señal eléctrica.³²

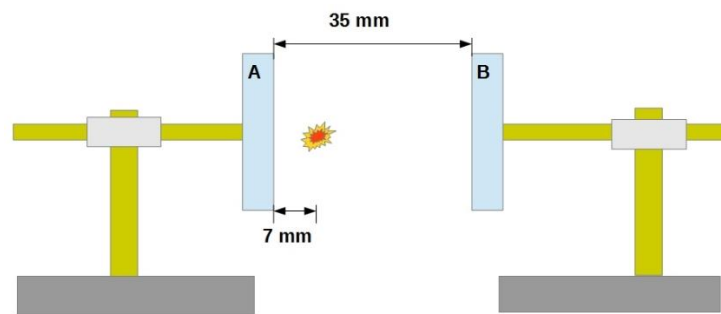


Figura 2.2. Posición fija de las placas. La posición óptima entre el plasma y la placa de tierra es 7 (0.05) mm. Figura tomada de mi tesis de licenciatura.⁶

Por último, para la medición de la energía incidente y la energía transmitida se utilizó un medidor de energía y de potencia modelo VEGA y un cabezal piroeléctrico de alta energía modelo PE50BF-DIFC/PE50BF-DIFH-C, ambos de la marca OPHIR. Para determinar la energía incidente el cabezal del piroeléctrico se colocó después de la lente L1, en la posición del cuadro punteado (ver Figura 2.1). El cabezal fue colocado de una manera muy próxima a esta lente para evitar la formación del plasma en la superficie del cabezal piroeléctrico y así evitar su daño. Después de calibrar el atenuador con las energías requeridas, lo que se describe en la sección 2.2.1, se colocó el cabezal del piroeléctrico detrás del punto de formación del plasma, para medir la energía transmitida.

El osciloscopio, la fuente de alto voltaje y el generador de pulsos se controlaron mediante un dispositivo controlador *GPB – USB – HS* de la marca National Instrument. El atenuador y el medidor de energía fueron conectados mediante un bus *USB*. Se utilizó un programa escrito en Matlab, versión 2019b, para controlar el generador de pulsos, atenuador de energía, osciloscopio, fuente de alto voltaje y el medidor de energía. En los subcapítulos siguientes se describirá de forma más detallada la automatización de cada aparato y cada sistema.

2.2 Programación para la automatización del sistema experimental

En esta sección se explica desde el punto de vista de la programación el funcionamiento del sistema experimental, y más específicamente de su automatización. Se dividió la sección en 3 apartados distintos. El primer apartado consiste en el sistema de calibración de energía de los pulsos. El segundo apartado es el programa de control general del experimento. El tercer apartado es el de procesamiento de datos, donde se explica el tratamiento que se le dio a los datos obtenidos.

En cada uno de los apartados se muestra una estructura general de los programas. En los apéndices 7.3 y 7.4 se muestran los programas de cada uno de los dispositivos utilizados en este trabajo. Los comandos utilizados se encuentran explicados de forma explícita en el apéndice 7.5. Los programas completos utilizados se encuentran apartir del apéndice 7.6.

2.2.1 Sistema de calibración de la energía de los pulsos

El programa completo del sistema de calibración de la energía de los pulsos se encuentra en el apéndice 7.11. No obstante, a continuación se da una descripción de su estructura.

Anteriormente se mostró el funcionamiento del atenuador, que consistió en girar una placa de onda en pequeños pasos y de esta manera se disminuyó o aumento la energía de los pulsos de láser. Sin embargo, esto todavía no permite conocer los valores de la energía que salió del atenuador. Para ello se realizó la calibración del atenuador, que significa conocer la relación entre la energía que sale y la que entra, para cada paso del mismo. Para esta calibración se utilizó el medidor de energía, el generador de pulsos, el láser y el atenuador en una configuración como la de la Figura 2.3. Cabe recordar que la configuración de la óptica antes del atenuador dependió del rango de energía a utilizar. En la Figura 2.3 se muestra la configuración para el rango de energía de [180,360] mJ.

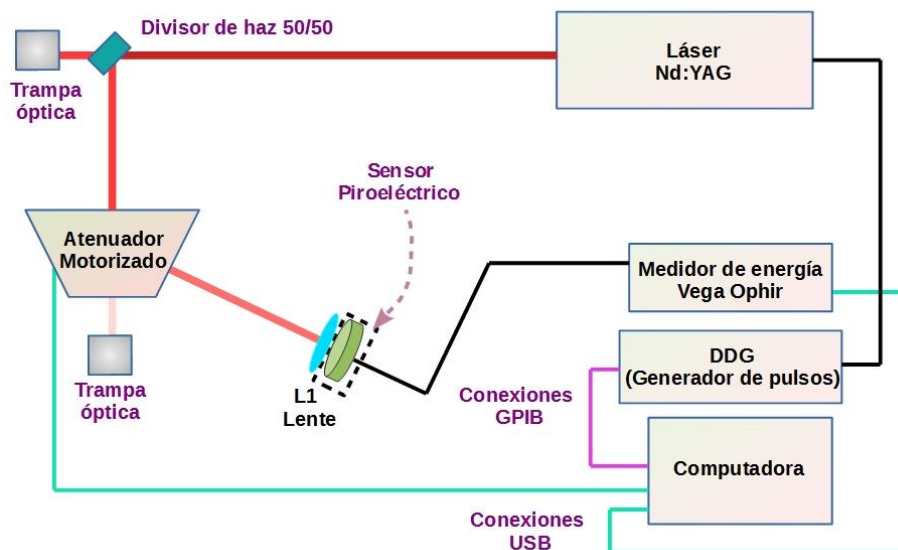


Figura 2.3. Configuración del sistema de calibración correspondiente al rango de energía [180,360] mJ.

Por experimentos realizados en este trabajo se pudo observar que el haz del láser sufre deformaciones al atenuar la energía por debajo del 25% de la energía total de entrada. Ésto puede implicar que la formación del plasma se vea afectada o modificada. Para asegurar que no pase esto se acotó el mínimo del rango de trabajo del atenuador a un 30% de la

energía total. Por otro lado, utilizando el láser de forma manual y el programa del atenuador, se encontró que el rango de pasos en el que es óptimo a operar va de $m = [-550,2500]$ pasos. Varios aspectos de la programación de los aparatos involucrados en el sistema de calibración ya se pueden revisar en los apéndices 7.4 y 7.5. Se puede observar en la Figura 2.4 el diagrama de flujo del proceso de calibración.

El primer paso consistió en definir los parámetros utilizados en cada dispositivo. Seguido, se hizo la vinculación de los dispositivos con la computadora. Se abrió la carpeta donde se guardaron los archivos (Figura 2.4 - 3). Después, se colocó el motor en la posición inicial que da el atenuador por defecto, se movió el motor a la posición cero que fue en el paso $m = -550$. Se utilizó un ciclo *for* para variar los pasos m , de -550 a 2500, con pasos de 50. En cada ciclo se dispararon 175 pulsos, esto debido a que en trabajos anteriores ⁶ se mostró que después de 150 pulsos la energía de los pulsos de láser se hace estable. Posteriormente, el medidor tomó los datos de la energía de cada pulso que envió al Matlab, donde se hizo el cálculo de la promediación y la desviación estándar de los últimos 25 pulsos. Estos datos, en conjunto con el paso m se fueron guardando en una lista.

Para observar la evolución de la calibración se graficaron los datos: promedio de energía vs pasos y energía de cada pulso vs número del pulso de láser. El primero de estos gráficos es la calibración de la energía de los pulsos de láser. El segundo se mostraba en la pantalla en tiempo real de manera que se podía observar si algún valor de energía quedaba fuera de rango. Después, concluido el ciclo *for* en un archivo .txt se guardaron los datos de la energía promedio y su desviación estándar, para cada paso. Es pertinente mencionar que el número de pasos se eligió tomando en cuenta los siguientes dos factores: tiempo de calibración (se refiere al tiempo real que toma hacer la calibración) y resolución de la calibración (número de saltos entre pasos del rango [-550,2500]). En este caso se tomaron

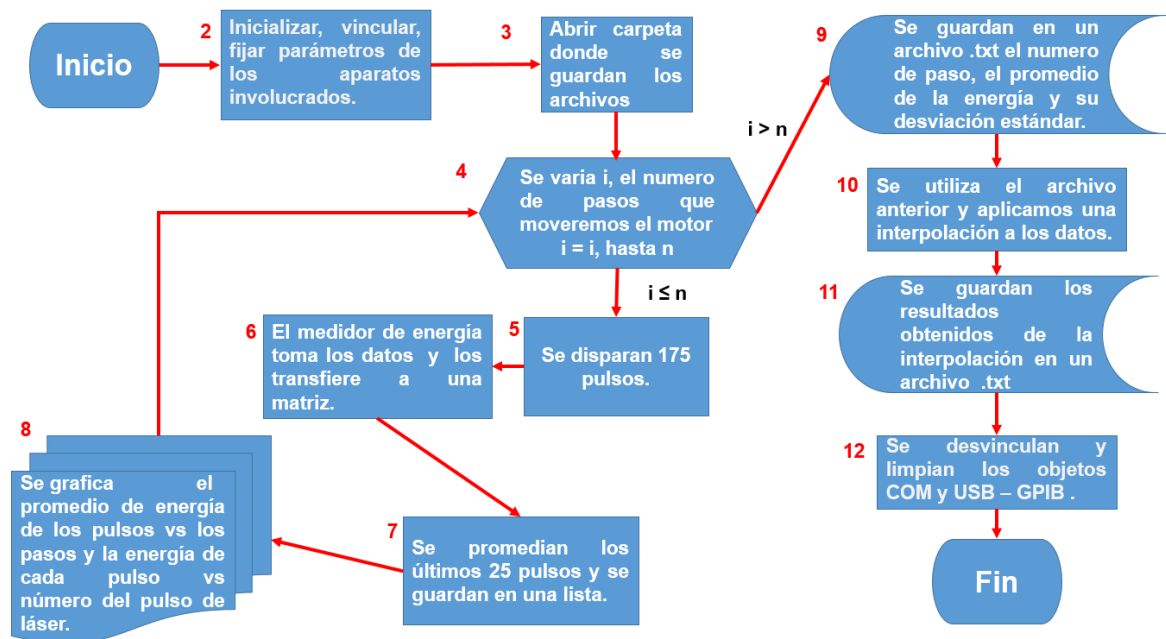


Figura 2.4. Diagrama de flujo del sistema de calibración de la energía.

61 puntos y tomando en cuenta que 175 pulsos de láser equivalen a 17.5 segundos, el tiempo por experimento fue de 18 minutos. Se realizó la calibración por cada rango de energía y por cada lente utilizada. Teniendo en cuenta estos factores, se hizo una interpolación de los datos de la energía de los pulsos de láser guardados, para poder obtener las energías correspondientes a los pasos intermedios (Figura 2.4 - 10). Finalmente

se guardaron los datos en un archivo .txt (Figura 2.4 - 11), se desvincularon y limpiaron los objetos COM y *USB – GPIB*.

2.2.2 Programa general de control de los experimentos

En esta sección se describe el programa general que se utilizó para realizar los experimentos de esta tesis. Se dividió esta sección en tres subsecciones para facilitar la explicación. La primera subsección explica los parámetros del experimento y se muestran los rangos en los que se variaron dichos parámetros. La segunda subsección del experimento describe en que consistió el experimento. Finalmente, la subsección de la programación describe el algoritmo utilizado para llevar a cabo el experimento, el programa a detalle lo podemos ver en el apéndice 7.12.

2.2.2.1 Parámetros de los experimentos

En los experimentos realizados en esta tesis se variaron tres parámetros: la energía de los pulsos de láser, el voltaje aplicado al capacitor y la distancia focal de las lentes.

En el uso de las diferentes distancias focales mencionadas en la sección 2.1.1 se tuvo en cuenta ciertos detalles experimentales. Es conocido que para un mismo pulso de láser una reducción de la distancia focal de la lente significa un incremento de la densidad de potencia incidente, lo cual favorece la formación de plasma en el aire. Esto significa que para algunos pulsos de láser con algunas distancias focales se pueda generar plasma y con otras no. Para cada lente, existe un rango donde la generación del plasma no es estable. Por ejemplo, para la lente de f : 75 mm los pulsos de menos de 40 mJ no resultaron estables, es decir, a veces se generaba plasma y a veces no. Teniendo esto en cuenta, los rangos de energía para cada lente fueron seleccionados de manera que estos estuvieran por encima de los valores umbrales de energía para la formación de plasma (ver Tabla 2.1). Considerando los tres rangos de energía utilizados en los experimentos se variaron los saltos de valores de energía para cada rango, de manera que para el rango de energía 1 se realizó con saltos de 30 mJ, para el rango 2 con saltos de 20 mJ y para el rango 3 con saltos de 10 mJ.

f (mm)	No hay plasma por debajo(mJ)	Plasma estable por encima de (mJ)	Rango de energía 1 (mJ)	Rango de energía 2 (mJ)	Rango de energía 3 (mJ)
75	28	40	360 – 660	180 – 340	50 – 160
100	28	55	360 – 660	180 – 340	60 – 160
125	30	74	360 – 660	180 – 340	80 – 160
150	34	93	360 – 660	180 – 340	100 – 160
200	50	180	360 – 660	180 – 340	
250	38	217	360 – 660	220 – 340	
300	45	307	360 – 660		
400	38	480	480 – 660		
500	55	570	570 – 660		

Tabla 2.1. Rangos de energías utilizados para cada lente.

Para el voltaje aplicado al capacitor se tomaron dos rangos: [400,3000] V y [50,200] V. El primer rango con saltos de 200 V y el segundo con saltos de 30 V.

2.2.2.2 Experimento

El experimento realizado en este trabajo consistió en generar un plasma enfocando pulsos de láser de nanosegundos con una lente convergente. El plasma provocó una perturbación en un campo eléctrico formado entre un par de placas de un capacitor, de modo que se midieron caídas de voltaje en una resistencia conectada a la placa de tierra como consecuencia de dichas perturbaciones. Entonces, para realizar el monitoreo de este fenómeno se usó lo siguiente: un fotodiodo que monitorea los pulsos de láser, un osciloscopio que muestra las señales producidas consecuencia de la perturbación del plasma y las señales producidas debidas al los pulsos de láser y un medidor de energía que determinó la energía incidente (energía utilizada para generar el plasma) y la energía transmitida (energía que pasa de largo, que no es ni absorbida ni esparcida por el plasma).

2.2.2.3 Programación

Para la realización del programa general se usó la configuración como se muestra en la Figura 2.1. Se utilizó el programa de calibración de energía para definir los pasos m y las energías de los pulsos de láser correspondientes. La calibración se realizó antes de cada experimento.

En la Figura 2.5 se muestra el diagrama de flujo del experimento. Igual que en el programa de calibración solo se abordara los puntos clave que no se abordaron en los subsecciones anteriores.

Se definieron los parámetros iniciales, se inicializó y vincularon los dispositivos involucrados. Después se creó una lista mediante un ciclo *for* con variable H donde introducimos de manera ordenada los valores de los pasos m y sus correspondientes valores de energías (ver Figura 2.5 inciso a)), que previamente se obtuvieron en el proceso de calibración. Los valores de las energías se acomodaron de forma ascendente. Después, se abrió la carpeta donde se almacenaban los archivos de las señales (eléctrica y óptica) y los pulsos de energía. Seguido, mediante un ciclo *for* j se varió la energía moviendo el motor del atenuador. Esto utilizando la lista del párrafo anterior. Aquí se configuró la escala vertical del osciloscopio para el canal 3 que corresponde a la señal eléctrica. El valor de la escala vertical se eligió tomando en cuenta que la señal eléctrica con la máxima energía y el máximo voltaje estuvieran dentro de la ventana del osciloscopio y que la imagen no se saturara. Realizado esto se introdujo otro ciclo *for* con variable i pero ahora para variar el voltaje entre las placas. Los valores de los voltajes se aplicaron de forma ascendente. La forma de aplicar el voltaje y la energía de manera ascendente, se debe a que en mi tesis de licenciatura⁶ se observó que al incrementar estos dos parámetros anteriores la amplitud de la señal eléctrica crece de manera monótona. Seguido, se aplicó el primer voltaje y se puso al osciloscopio en espera para la toma de datos de las señales. Después, se activa el modo burst, de forma que se dispararon 600 pulsos de manera continua, en el tiempo que ocurren los 600 pulsos se toman las señales y los datos de las energías de los pulsos de láser. A continuación se desglosara como se ocuparon cada uno de los pulsos. Antes de continuar, es pertinente recordar que la frecuencia de disparo del láser es de 10 Hz, por lo que 10 pulsos equivalen en tiempo a 1 s. Teniendo esto en cuenta, se dividen los 600 pulsos en cinco pausas. La primera pausa fue de 12 s (120 pulsos), este paso es esencial debido a la necesidad de la estabilidad de la energía del láser. Antes de la segunda pausa de 18 s, se activó el osciloscopio en su forma secuencial. En este modo solo se toma en cuenta los primeros 100 pulsos adquiridos por el osciloscopio que son los que se promediarán, sin embargo, se disparan 180 debido a que la frecuencia de adquisición de datos del osciloscopio es menor a la frecuencia del láser para las características de adquisición de este experimento. La frecuencia de adquisición del osciloscopio depende, por ejemplo, del número de puntos de la señal adquirida, así mientras más puntos menor es la

frecuencia de adquisición. Seguido, el osciloscopio analizó la señal obtenida de la segunda pausa. De la medición del osciloscopio se obtuvo el valor máximo de la señal. Este valor ayudó al reescalamiento del osciloscopio en la parte vertical. Antes de continuar, es importante aclarar la forma y el porque del reescalamiento de las señales.

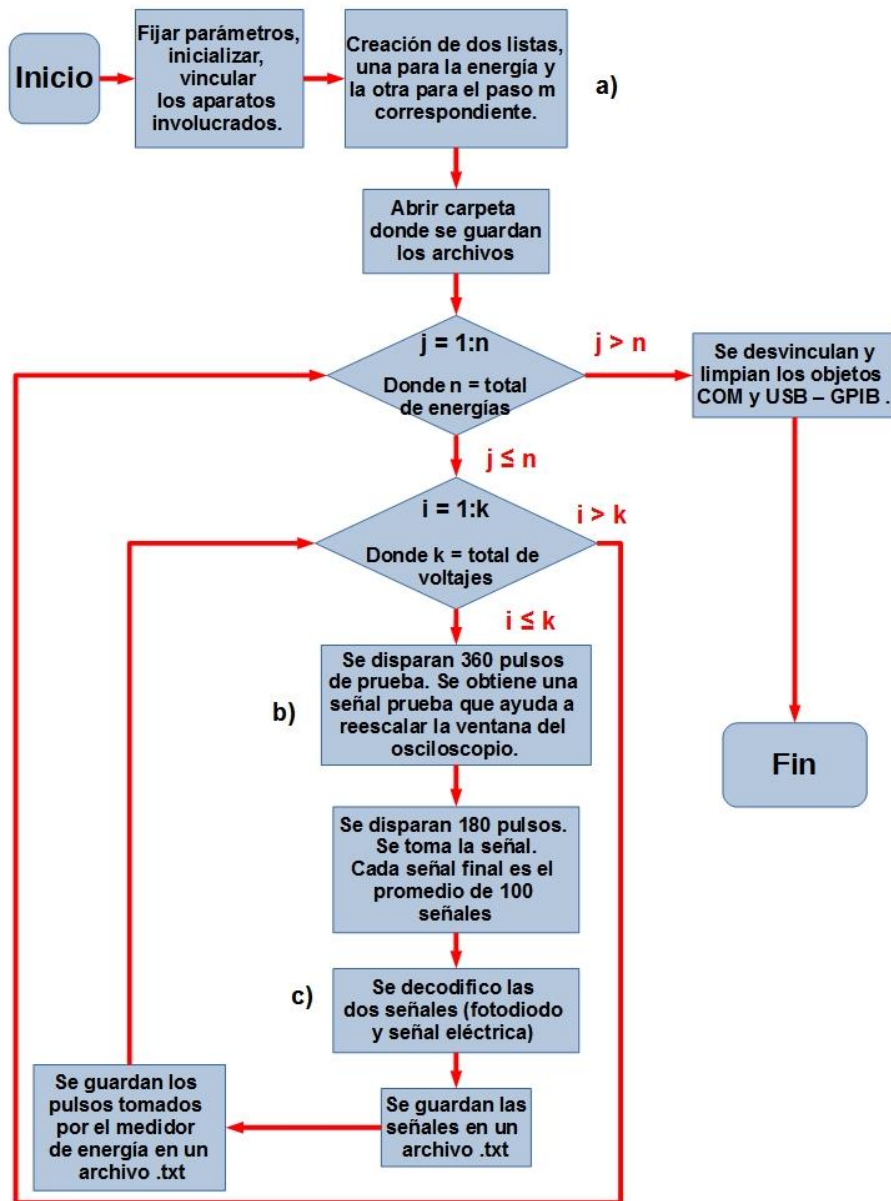


Figura 2.5. Diagrama de flujo del experimento.

El reescalamiento consiste, en un inicio tomar una escala donde la señal eléctrica con máximo voltaje y máxima energía no sature la ventana del osciloscopio y después al iniciar con la primera energía y el primer voltaje ir hacia valores inferiores de escala buscando la escala adecuada. Los reescalamientos se realizan teniendo en cuenta la resolución del osciloscopio y la señal captada. El osciloscopio trabaja en 8 bits, es decir, que la escala vertical de la pantalla de 10 V se dividirá en 256, lo que da una resolución de 39 mV. Debido a esto, por ejemplo, al tener señales con amplitudes de 100 mV el número de puntos en la señal sería de dos, por lo que se tendrá que reescalar la parte vertical de la pantalla. Después de enviar el código de reescalamiento se realizó la tercera pausa de 6 s, para

darle tiempo al osciloscopio de realizar este proceso. Posteriormente, se prepara el osciloscopio y se toma la señal definitiva, para lo cual se realiza la cuarta pausa de 20 s. Por ultimo, se hizo la decodificación de los datos de las señales obtenidas por el osciloscopio y se guardaron en un archivo .txt. (ver Figura 2.5 c)). Seguido, se tomó los datos del medidor de energía y se guardaron en un archivo .txt, para lo cual se realizó una pausa de 6 s para hacer estos dos últimos procesos. Terminado los dos ciclos *for j e i*, se desvincula y limpia la memoria de los objetos COM y *USB – GPIB*.

Lo descrito anteriormente se realizó para cada rango de energía y voltaje, además de cada una de las lentes.

2.2.3 Procesamiento de datos

En esta sección se explicará el tratamiento que se le dio a los datos obtenidos en los experimentos realizados en este trabajo. Es pertinente mencionar que todas las señales obtenidas tuvieron las siguientes características:

- Número de puntos adquiridos de la señal fue de 10000 puntos.
- La resolución o el tiempo entre dos puntos de la señal fue de 0.05 ns.

2.2.3.1 Ruido

Durante la preparación del experimento, se realizaron pruebas para medir el nivel de ruido de la instrumentación utilizada y analizar su influencia sobre las señales obtenidas. Las pruebas realizadas para observar el ruido fueron las siguientes:

- Prueba 1. Se midió el nivel de ruido del osciloscopio sin conectar cable alguno.
- Prueba 2. Se midió el nivel de ruido del osciloscopio conectando los cables bnc – bnc.
- Prueba 3. Se midió el nivel de ruido de todo el sistema experimental conectado y encendido.
- Prueba 4. Esta prueba incluye todos los ruidos anteriores, además de la medición de ruido provocado por la generación de los pulsos de láser, que fueron bloqueados con una trampa óptica en su recorrido hacia la lente. Este proceso se llevó a cabo sin que se creara plasma.

Se observó que de las dos primeras pruebas, el ruido estaba entre ± 1 mV y no seguía algún patrón determinado, es decir, se trató de ruido blanco. Sin embargo, para la prueba 3 se observó una señal de ruido con una frecuencia de 100 MHz y una amplitud entre ± 10 mV. La señal es continua en el tiempo y es independiente del sistema experimental. Para la prueba 4 se observó un ruido mayor (ver Figura 2.6) que se encontró en el rango de los $\pm [15,25]$ mV, que no dependió de los parámetros del láser que se mantuvieron fijos. Este ruido es la suma de tres señales: la suma del ruido blanco, el ruido con la frecuencia de 100 MHz y el ruido generado por las descargas eléctricas del *Q-switch* del láser, que tiene una frecuencia de 250 MHz. El ruido de la prueba 4 es pulsado en el tiempo y con la misma frecuencia del láser, esto debido a que se produce de las descargas del *Q-switch* y tiene una duración aproximada de 500 ns.

Antes de iniciar los experimentos se tomó una señal de ruido (ruido de la prueba 4), con las mismas características que se tomaron las señales eléctricas. A estas últimas se les resto la señal de ruido. En la Figura 2.6 se observa el ruido perteneciente a dos diferentes experimentos.

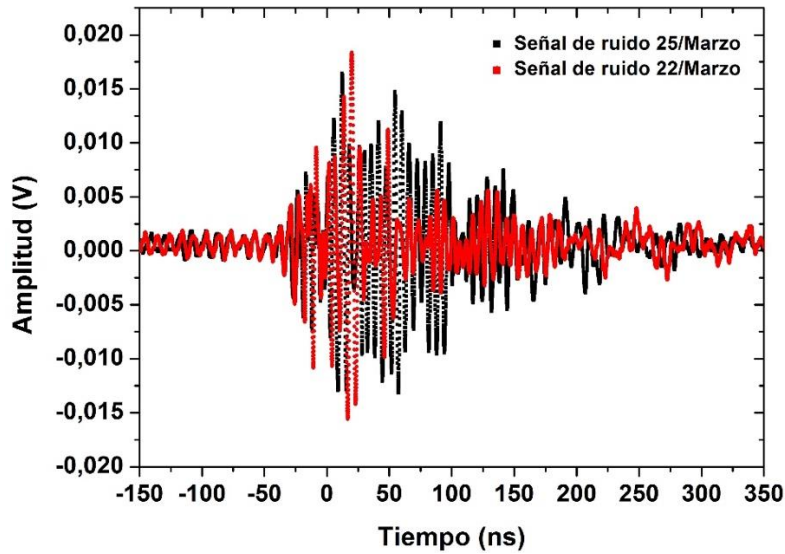


Figura 2.6. Señales de ruido.

2.2.3.2 Procesamiento de señales y datos

La información recabada en este trabajo fue de dos tipos: las señales captadas por el osciloscopio y los datos recabados a través del medidor de energía.

Las señales captadas por el osciloscopio fueron de dos tipos: las señales provenientes del fotodiodo que corresponden a la evolución temporal de los pulsos de láser y las señales eléctricas provenientes del sensor eléctrico que ofrecen información sobre la creación y evolución del plasma. Las señales finales de ambos tipos resultaron de la promediación de 100 señales captadas por el osciloscopio (ver Figura 2.7).

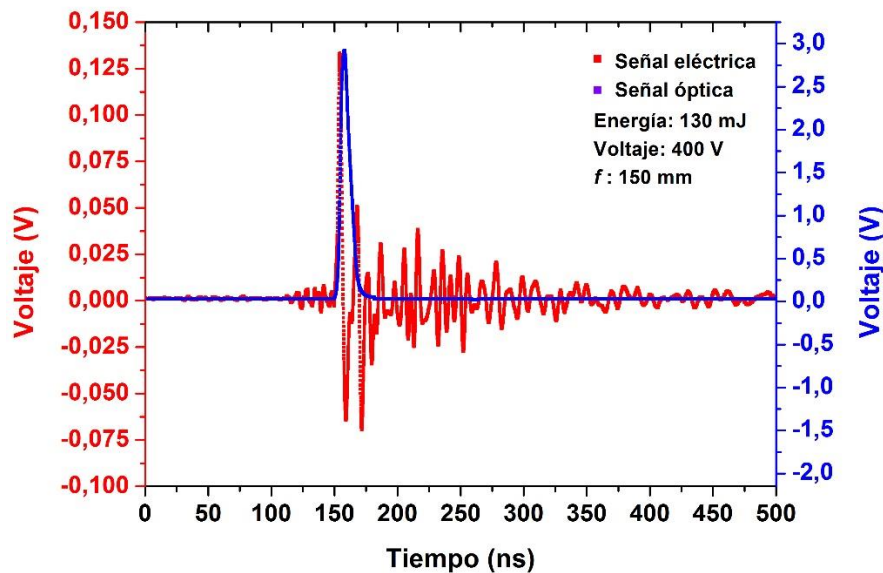


Figura 2.7. Señal óptica y eléctrica.

Las señales que se obtuvieron del experimento se guardaron con una nomenclatura específica como se muestra a continuación: *CH3_300mm_360mJ_1800V*. El primer término corresponde al canal del osciloscopio teniendo dos posibilidades de la nomenclatura, CH3 para el canal 3 (señal eléctrica) y CH1 para el canal 1 (señal óptica). El segundo término

corresponde a la distancia focal utilizada. El tercer término corresponde a la energía de los pulsos de láser. Finalmente, el cuarto término corresponde al voltaje aplicado. El guardado de las señales de esta manera permitió que el análisis de las señales también fuera de manera automatizada. La automatización del análisis se debe a que el número de señales captadas en los experimentos son muchas, por ejemplo, para el experimento donde se utilizó una lente con distancia focal de 75 mm, el rango de energía de [50,160] mJ y el rango de voltaje de [400, 3000] V se tuvo un total de 336 señales teniendo en cuenta los dos canales. Entonces, analizar las señales de forma manual resulta ineficiente.

Del trabajo realizado en mi tesis de licenciatura⁶ se conoce que la forma de la señal eléctrica es independiente de los rangos estudiados, el único cambio es en la amplitud de los picos. Para el análisis realizado a las señales se utilizó la amplitud del primer pico (y_{max}). Antes de realizar el análisis de la amplitud se hizo un tratamiento a las señales, el tratamiento consistió en lo siguiente: quitar el *offset* en las señales (eléctrica, óptica y de ruido) y hacer coincidir los inicios de las señales, esto último debido a como se decodificaron las señales (ver Figura 2.8). Después, a la señal eléctrica se le restó la señal de ruido y se procedió a obtener la amplitud del primer pico. Finalmente se guardó la señal sin *offset* y la señal sin *offset* y ruido.

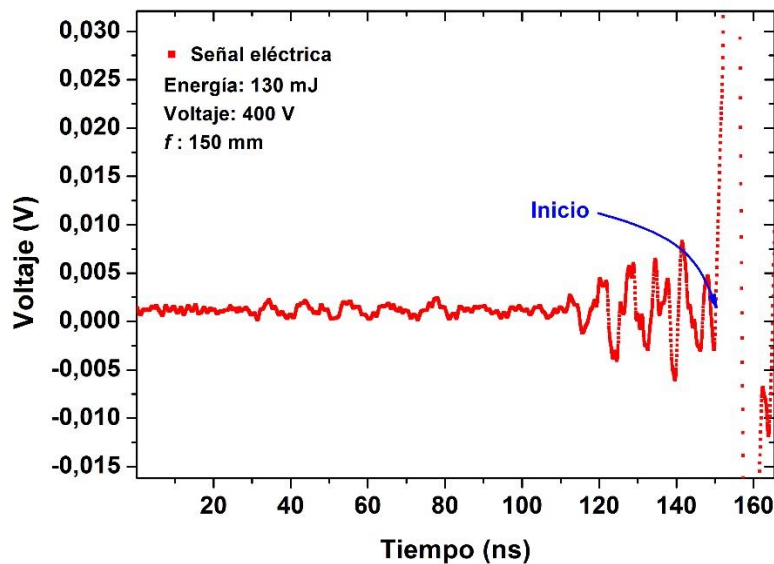


Figura 2.8. Zoom de la señal eléctrica de la Figura 2.7 . Se muestra el inicio de la señal y el *offset*.

Por otro lado, se analizó los datos obtenidos mediante el medidor de energía. Recordemos que el experimento constó de 600 pulsos cada ciclo. El medidor de energía tomó lectura de cada pulso y los guardó en un archivo. A diferencia del osciloscopio el medidor de energía tiene una frecuencia mayor de adquisición que la de los pulsos de láser. Recordemos que del total de pulsos solo una parte es utilizada para la promediación de las señales eléctricas, la porción utilizada parte desde el pulso 361 al 460. Teniendo esto en cuenta, se tomó el archivo donde se guardaron los datos de las energías de los pulsos y se calculó el promedio de la energía transmitida y su error, utilizando las ecuaciones (2-1) a (2-3).

Promedio (media)
$$\bar{y} = \frac{1}{N} \sum_{i=1}^n y_i, \quad (2-1)$$

Desviación estándar

$$\Delta y = \sigma = \sqrt{\frac{\sum_{i=1}^N (y_i - \bar{y})^2}{N}}, \quad (2-2)$$

Error

$$\Delta y_{\text{avg}} = \frac{\sigma}{\sqrt{N}}, \quad (2-3)$$

A partir del promedio de la energía transmitida se calculó la transmitancia y se graficó con respecto al parámetro de interés (voltaje aplicado a las placas, energía de los pulsos del láser o distancia focal). La transmitancia viene dada por:⁶

$$T = \frac{E_t}{E_i}, \quad (2-4)$$

donde E_t es el promedio de la energía transmitida y E_i es el promedio de la energía incidente. La energía incidente se obtuvo de la calibración de la energía, que se midió después de la lente y antes del punto de enfoque.

Los datos de las señales eléctricas y de las energías de los pulsos se analizaron en un solo programa (ver apéndice 7.13).

En otro programa se llevo a cabo el análisis de integral de la señal eléctrica en el tiempo haciendo uso de una función propia de Matlab para estos fines. Posteriormente, se realizó una normalización a esta señal y se comparó con la señal óptica, que también esta normalizada (ver apéndice 7.14). Finalmente, se guardaron las señales de las integrales en un archivo .txt.

3 Mediciones y resultados

A lo largo de este capítulo se revisarán los datos obtenidos de la automatización del sistema experimental, incluyendo la calibración de la energía de los pulsos del láser.

Se hará una revisión de las señales de ruido, se centrará los estudios en las señales tomadas (eléctricas y óptica) y los datos recabados por el medidor de energía (energía incidente y transmitida), con respecto a los parámetros que se variaron en los experimentos.

3.1 Calibración de la energía de los pulsos del láser

Se obtuvo un sistema que nos permite automatizar un atenuador de energía motorizado, un generador de pulsos y un medidor de energía. El sistema funcionó para dos cosas: calibrar la energía de un láser pulsado respecto a la rotación del motor (por pasos m) y conociendo la relación rotación/energía se utilizó en un programa general para atenuar la energía del láser.

La calibración de la energía se realizó dando pasos de 50, iniciando en la posición de -550 hasta la posición de 1850 y por cada paso se promedió 25 pulsos. La curva de calibración se muestra en la Figura 3.1 (los puntos tienen sus barras de error). En los experimentos realizados en este trabajo se utilizaron energías específicas, por ejemplo, en el rango de [180,340] mJ se usaron las energías de 180 mJ, 200 mJ, etc. Sin embargo, la forma de calibrar por pasos no tiene que dar necesariamente las energías de los pulsos que se utilizaron. Entonces, para obtener las energías intermedias entre los pasos se aplicó una interpolación. La curva de calibración se muestra en la Figura 3.1.

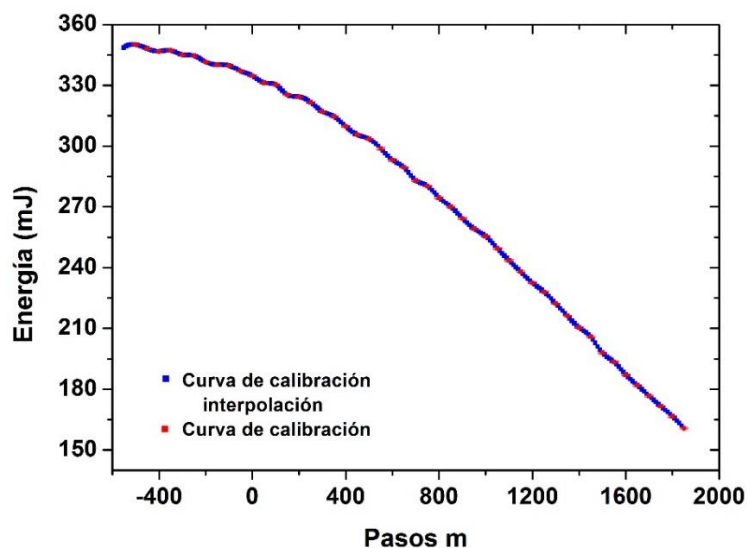
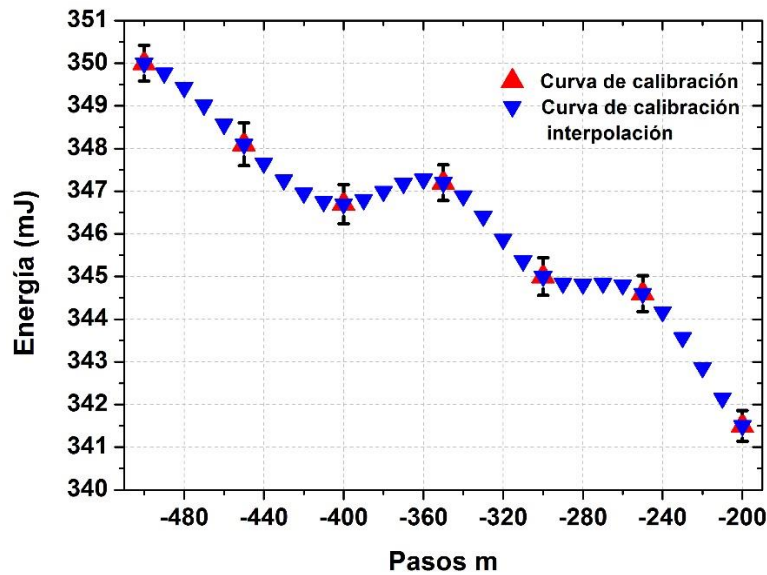


Figura 3.1. Curvas de calibración de la energía. Rango de energía [180,340] mJ.

De las curvas de calibración se observó que las energías de los pulsos cercanos al máximo de energía, ubicados próximo al paso $m=-550$, pueden ser la misma para diferentes pasos m , por ejemplo, el rango de $[-440,-350]$ pasos corresponde la energía de 347 mJ (ver Figura 3.2 a)). Por lo que las mediciones de la energía cercanos al máximo deben tener saltos de pasos mayores a los 100 pasos. Por otro lado, en el rango $[900,1200]$ pasos las mediciones de la energía tiene una relación biunívoca. Pudiendo tener una resolución de 10 pasos (ver Figura 3.2 b)). Entonces, si se requiere tener una resolución fina de la energía, se debe acudir a la parte donde la curva de calibración tiene un comportamiento lineal.

a)



b)

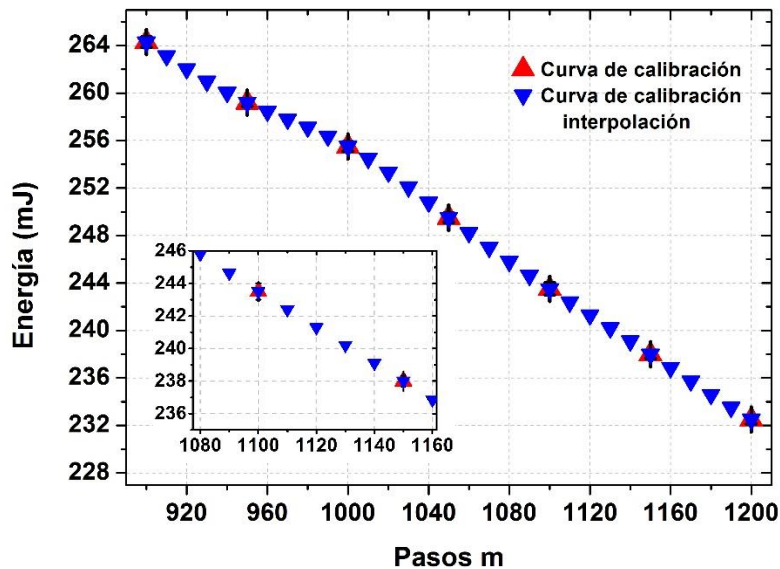


Figura 3.2. Curva de calibración. a) Rango $[-500,-200]$ pasos. b) Rango $[900,1200]$ pasos, se realiza un acercamiento al intervalo de $[1080,1160]$ pasos donde se aprecia las barras de error de los datos (en negro).

3.2 Señal eléctrica y su ruido

Anteriormente se mencionó que se trabajó con dos tipos de señales: la eléctrica y la óptica. Las señales eléctricas se componen de las propias señales eléctricas y sus ruidos. El comportamiento oscilatorio de las señales eléctricas es una característica que resalta a primera vista (ver Figura 3.3 a)). Este comportamiento oscilatorio debe tener su origen en la instrumentación (circuito y cables), debido a que se trabajó con altas frecuencias (mayores a 100 MHz) y los fenómenos inductivos de los elementos antes mencionados se hicieron presentes. En el subsección 2.2.3.1, se mencionó que la señal de ruido constó de la suma de tres señales: la primera señal, perteneciente al ruido blanco con una amplitud entre ± 1 mV; la segunda señal, perteneciente a la frecuencia de 100 MHz, continua y con una amplitud entre los ± 10 mV; y la tercera señal, con una frecuencia de 250 MHz, pulsada y con una amplitud en el rango de $\pm [15,25]$ mV. La primera señal de ruido es la propia del osciloscopio, sin embargo su contribución es pequeña en la señal de ruido debido a la baja amplitud y su naturaleza aleatoria. La segunda señal de ruido tiene su origen en señales provenientes del exterior (señales de radio), ya que el sistema al conectarse y específicamente el par de placas actúan como una antena. Esta señal tiene una pequeña contribución, esto debido a que su amplitud es menor a los 10 mV y de igual forma es aleatoria, lo cual hace que su aporte sea menor cuando se realiza la promediación de la señal. Por otro lado, la tercera señal tiene su origen en la descarga del Q-switch y tiene una naturaleza pulsada, debido a que solo se genera cuando es disparado el láser. Esta señal domina debido a que su amplitud se encuentra en el rango de los $\pm [15,25]$ mV.

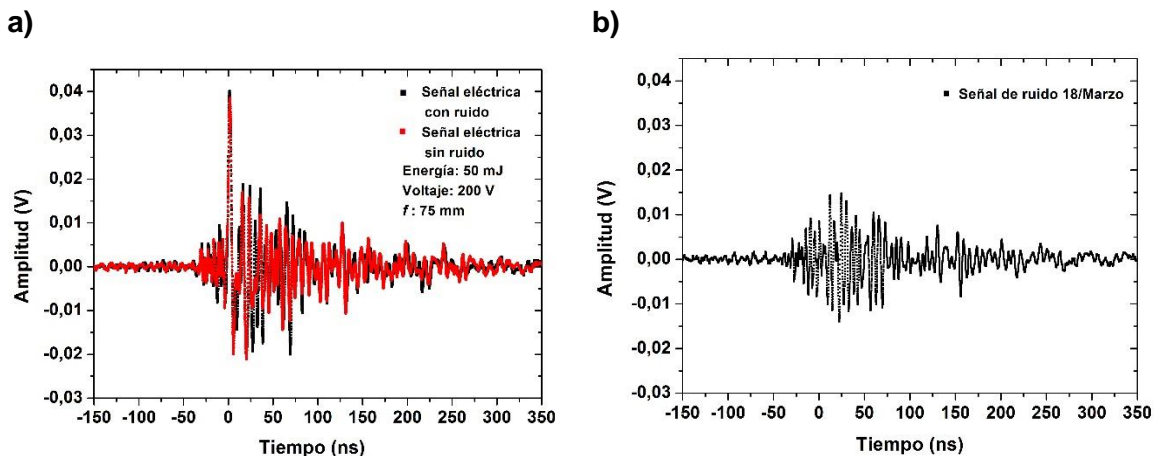


Figura 3.3. a) Señal eléctrica con y sin ruido. b) Señal de ruido.

La señal de ruido mantuvo su forma, solo cambió en amplitud, lo que permitió realizar un procesamiento sencillo a las señales eléctricas. El procesamiento consistió en restar la señal de ruido perteneciente al día de la medición a las señales eléctricas del mismo día.

La señal eléctrica como se mencionó anteriormente tiene un comportamiento oscilatorio debido a la instrumentación, no obstante no se realizó un análisis profundo debido a la complejidad de los fenómenos involucrados. Sin embargo, se pudo trabajar con esta señal tomando el primer pico, que es el de mayor amplitud. En el trabajo de Bredice²⁵ se encuentra una relación donde el pico máximo es directamente proporcional a la energía de rompimiento y a la tensión entre las placas, e inversamente proporcional al cubo de la distancia (ecu. (1-6)). Entonces, a lo largo del trabajo y tomando en cuenta lo explicado anteriormente se optó con solo utilizar la amplitud del primer pico de la señal eléctrica.

Se observó que para los rangos de voltaje de [50,200] V y todos los valores de energía utilizados las señales eléctricas no tienen una distinción clara con la señal de ruido. En la Figura 3.3 se muestra una señal con los siguientes parámetros f : 75 mm, energía: 50 mJ y voltaje: 200 V y la señal de ruido perteneciente al mismo día de las mediciones. Se observa que la amplitud del primer pico de la señal eléctrica comparada con la amplitud del máximo pico de ruido es del doble. Entonces, para tener una relación señal a ruido de nuestro parámetro de interés de por lo menos el triple, descartamos la señales del rango [50,200] V. Cabe, aclarar que la relación señal a ruido, no es la usual, ya que en este trabajo nos centramos en la amplitud del primer pico de la señal eléctrica y no en toda la señal.

En las siguientes secciones se relacionará la amplitud del primer pico de la señal eléctrica con los parámetros que se variaron en este trabajo: distancia focal, energía y voltaje.

3.3 Variación de energía y voltaje

En esta sección se muestra los resultados obtenidos de la amplitud del primer pico de la señal eléctrica con respecto a la variación de la energía (para sus tres rangos) y el voltaje. Debido a la cantidad de datos obtenidos y para no saturar los gráficos solo se eligieron los correspondientes a ciertos valores de energías y de voltajes.

3.3.1 Rango 50 mJ a 160 mJ

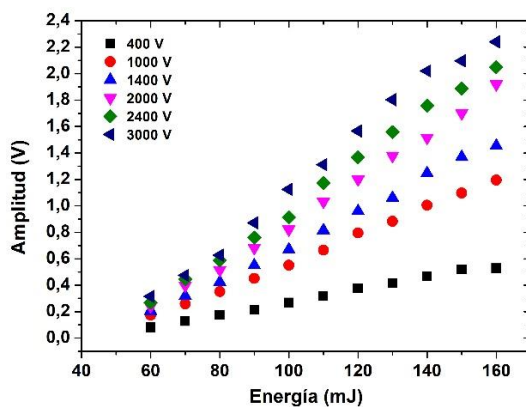
Para este rango, como se muestra en Tabla 2.1 se pudieron utilizar cuatro distintas lentes. Sin embargo, para no saturar la imagen se muestran solamente los resultados correspondientes a dos lentes: la lente de 100 mm y 150 mm (ver Figura 3.4).

Para f : 100 mm, se observa del gráfico de la amplitud de la señal eléctrica contra energía (ver Figura 3.4 a)), que al variar el voltaje se tiene una amplificación de la amplitud del primer pico de la señal eléctrica. De igual forma, en el gráfico de amplitud contra voltaje (ver Figura 3.4 c)), se observa que al variar la energía hay una amplificación.

Para f : 150 mm, ocurre el mismo comportamiento para los dos gráficos (ver Figura 3.4 b) y d)). Sin embargo, se observa que las amplitudes del primer pico de la señal eléctrica para la lente de 150 mm son menores. Esto sucede debido a que se tiene una mayor densidad de potencia en 100 mm, que en 150 mm, tal como se muestra en la subsección 1.1.3.

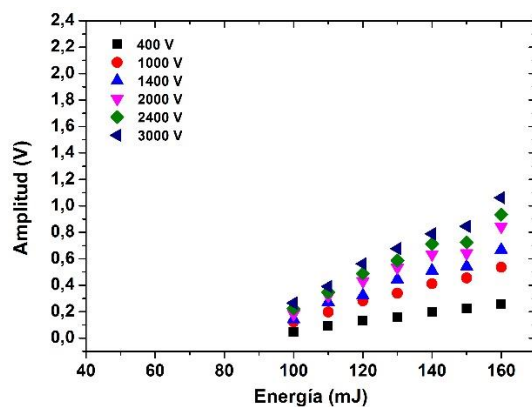
100 mm

a)



150 mm

b)



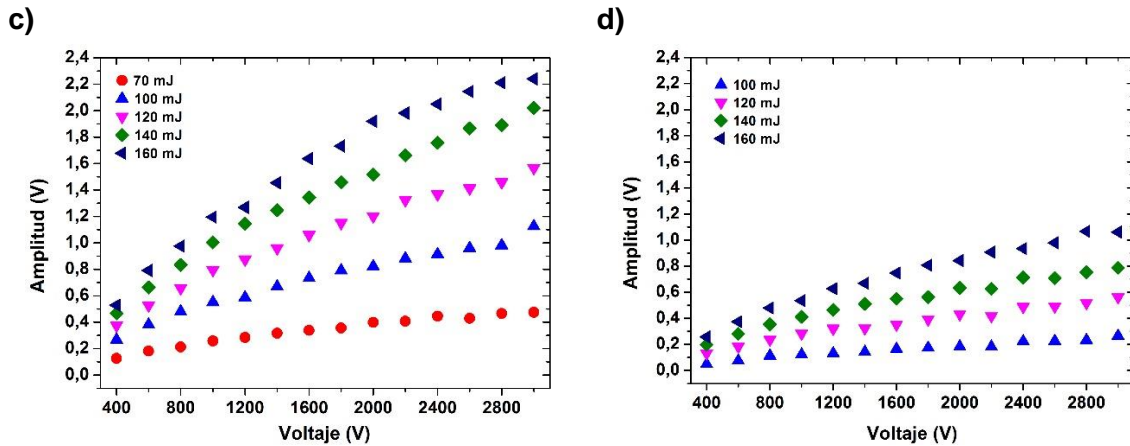


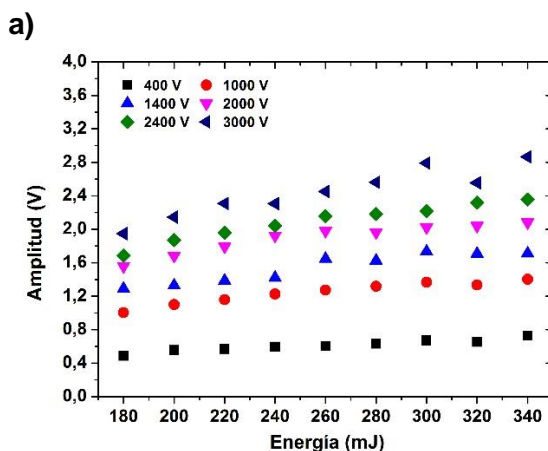
Figura 3.4. Gráficos de amplitud contra voltaje y energía para dos distancia focales, 100 mm y 150 mm.

3.3.2 Rango de 180 mJ a 340 mJ

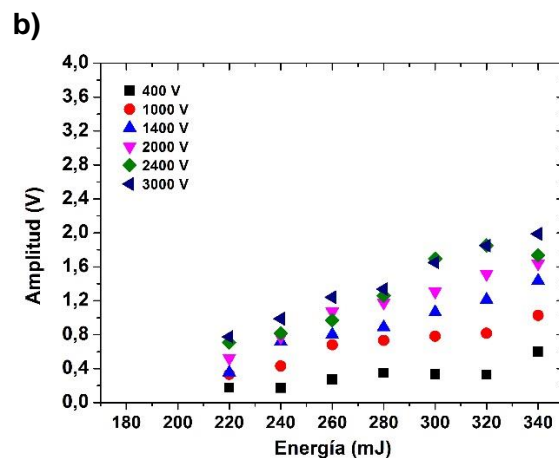
De la misma forma que en el anterior rango, se tomaron solo dos distancias focales, 100 mm y 250 mm. En este rango se observa que para el gráfico de la amplitud de la señal eléctrica contra energía para la lente de 100 mm (Figura 3.5 a)) el incremento del voltaje produce una amplificación de la misma, que es mayor que para la lente de 250 mm. Aunque en el gráfico la pendiente de los diferentes conjuntos de datos es menor que en el rango de energía anterior, de manera que la amplitud de la señal eléctrica tiende a ser asintótica a medida que se aumenta la energía. Esto habla de una saturación de la misma probablemente debido a la cantidad limitada del material (densidad del aire) que puede formar el plasma.

De igual forma, para la lente de 250 mm en el gráfico de amplitud de la señal eléctrica contra energía (ver Figura 3.5 b)) el incremento del voltaje produce una amplificación. La pendiente de este gráfico tiene una inclinación mayor que para lente de 100 mm. Lo que muestra que la amplitud de la señal eléctrica no ha llegado a una saturación y la cantidad de material para formar el plasma aun puede seguir interactuando con los pulsos de láser en el foco de la lente. La diferencia entre las amplitudes de las señales eléctricas para los gráficos **a)** y **b)** se debe a que la densidad de potencia en la lente con distancia focal de 100 mm es mayor como se mostró en el subsección 1.1.3

100 mm



250 mm



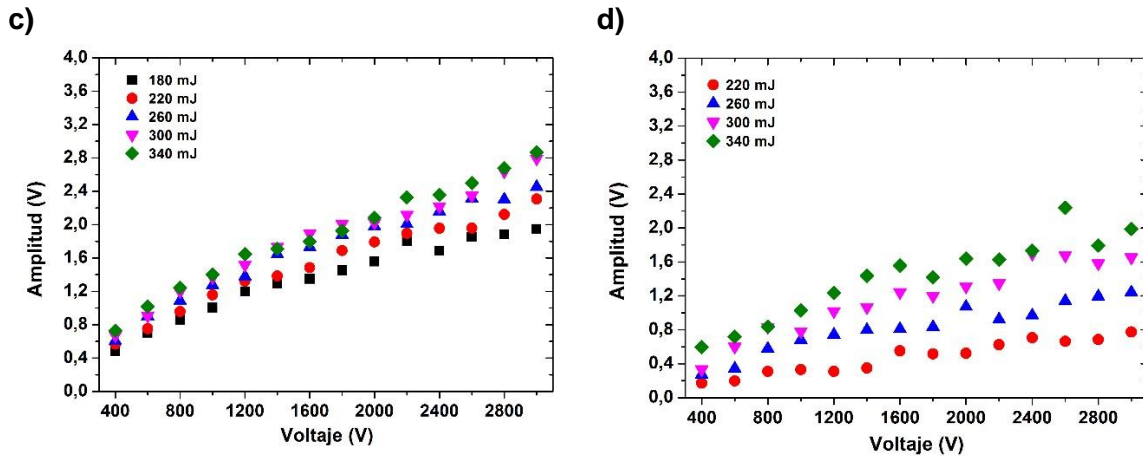


Figura 3.5. Gráficos amplitud contra voltaje y amplitud contra energía para dos distancias focales, 100 mm y 250 mm.

Es pertinente mencionar que a partir de la lente de 250 mm el plasma ya no es uno, sino un conjunto de plasmas formados a lo largo del eje del pulso láser (un “tren de plasmas” ver Figura 3.6).

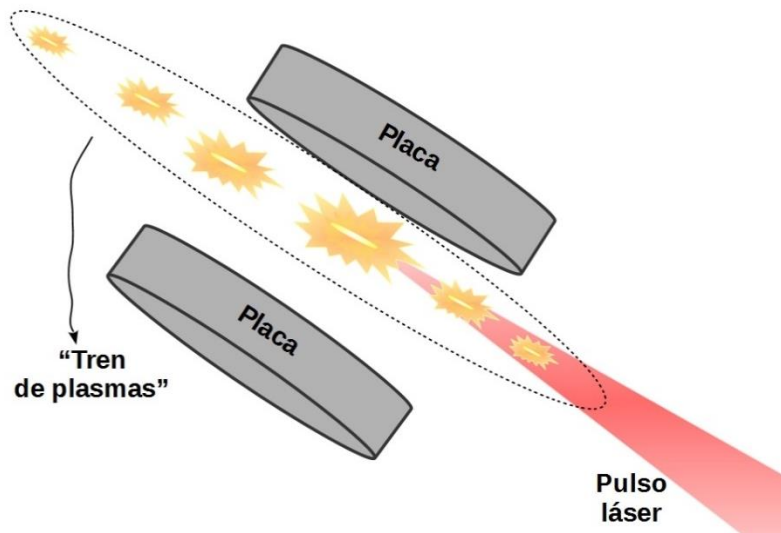


Figura 3.6. Esquema de la formación del “tren de plasmas”.

3.3.3 Rango 360 mJ a 660 mJ

Para los gráficos que se presentan para este rango de energías se utilizaron las lentes con distancias focales de 100 mm y 500 mm. Se observa para la lente de 100 mm en el gráfico de amplitud de la señal eléctrica contra energía que el voltaje sigue teniendo el efecto de amplificador de la señal (ver Figura 3.7 a)). Sin embargo, en este rango se hace más evidente que la amplitud de la señal eléctrica llegó a una saturación, lo que quiere decir que en el área donde se enfocó el pulso de láser no hay más material para la generación del plasma. Entonces, se tiene una densidad de potencia límite con esta lente para la cual la amplitud de la señal eléctrica no se incrementara aunque se aumente la energía. Por otro lado, se observa que para la lente de 500 mm en el gráfico de amplitud de la señal eléctrica contra la energía el voltaje de igual forma que en los dos anteriores rangos sigue teniendo el efecto amplificador (ver Figura 3.7 b)). A diferencia de la lente de 100 mm, se puede

observar que el aumento de la energía permite el aumento de la amplitud de la señal eléctrica. Esto es debido a que el área de enfoque es mayor, entonces se tiene una mayor cantidad de elementos que interaccionan con la luz láser y por ende se ve reflejado en la formación del plasma. Sin embargo, debido a la formación de varios plasmas (“tren de plasmas” (ver Figura 3.6)) los datos se vuelven caóticos y no se observa una curva suave.

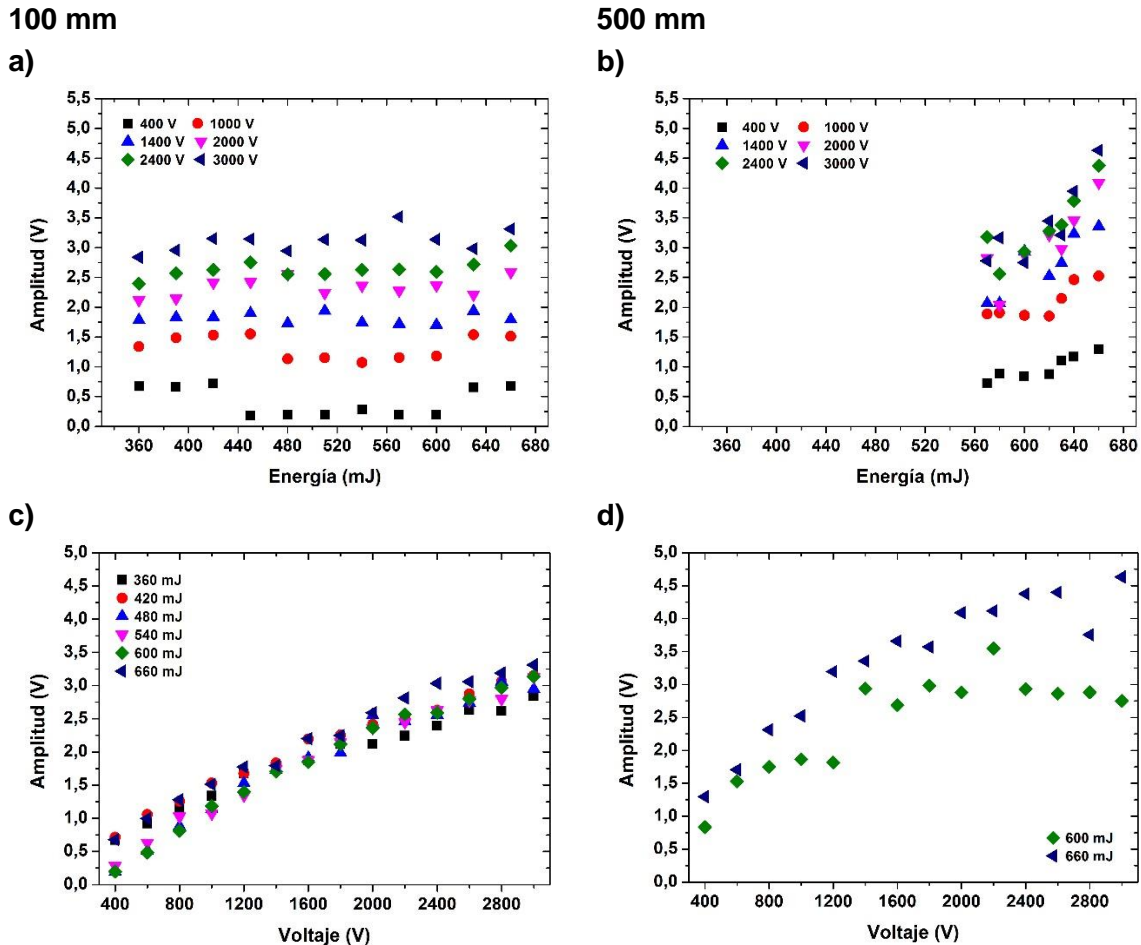


Figura 3.7 Gráficos de amplitud contra voltaje y amplitud contra energía para dos distancias focales, 100 mm y 500 mm.

3.3.4 Ecuación Amplitud (Energía, Voltaje)

Para poder desarrollar un medidor de energía se necesita encontrar una relación entre el parámetro que se quiere estudiar y sus variables. Aún más, es conveniente obtener una ecuación que refleje esta relación. En nuestro caso el parámetro a estudiar es la amplitud del primer pico de la señal eléctrica (A_M) y las variables son la energía (E), el voltaje (V) y la distancia focal. En este trabajo se observó que las características óptimas de estas variables para los cuales es viable desarrollar el medidor de energía son las siguientes:

- Utilizar distancias focales cortas, que en nuestro caso fueron menores a 150 mm, para tener densidades de potencia mayores.
- Utilizar los rangos de [50,160] mJ y [180,340], que es donde se observa un mayor cambio o pendiente de la amplitud de la señal eléctrica con el cambio de la energía.
- Utilizar voltajes mayores a los 1000 V, de modo que la pendiente mencionada en el punto anterior se incrementa.

Entonces, para cada lente y para cada rango de energía tendremos una ecuación $A_M(E, \mathcal{V})$. A continuación mostraremos el desarrollo de como se llegó a la ecuación que corresponde a la lente de 100 mm, en el rango [50 – 160] mJ.

En la Figura 3.8 se muestran los gráficos de amplitud contra energía y amplitud contra voltaje con sus respectivas curvas de ajuste. En estos gráficos se muestran solamente algunas de las funciones para no saturar la imagen. En cambio en el apéndice 7.15 se presentan todas.

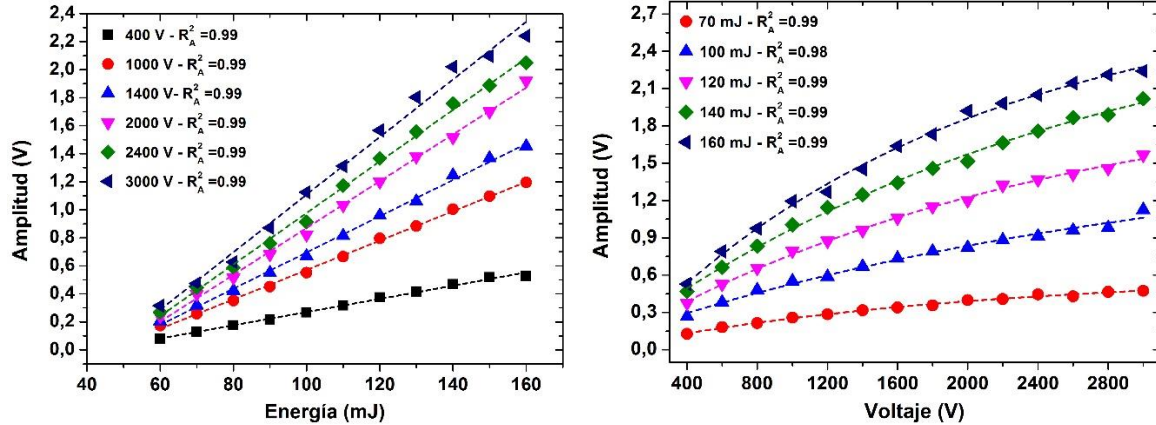


Figura 3.8. Gráficos de amplitud en función del voltaje y la energía para una distancia focal de 100 mm. Se muestra las curvas de ajuste.

Dentro de las funciones que se pueden utilizar para ajustar una curva a datos experimentales se encuentran: las rectas, polinómicas, potencias, exponenciales, etc. La Figura 3.8 a) permite ajustar sus curvas con una función lineal:

$$A_M(E, \mathcal{V}) = m(\mathcal{V}) E + b(\mathcal{V}) \quad (3-1)$$

Donde m y b son parámetros encontrados a partir de los ajuste de los datos recabados. En la Figura 3.8 se muestra el coeficiente de determinación ajustado R_A^2 para corroborar la calidad del modelo ajustado. Para los seis voltajes mostrados en la Figura 3.8 se tiene un R_A^2 de 0.99. Como se muestra en el apéndice 7.15 este valor es homogéneo para todos los voltajes analizados. Para cada conjunto de datos correspondientes a un voltaje, (3-1), obteniendo los valores de los coeficientes m y b (ver apéndice 7.15). De esta manera es posible graficar los valores de m y b con respecto al voltaje (ver Figura 3.9). A cada una de estas curvas $m(\mathcal{V})$ y $b(\mathcal{V})$ se les ajustó una función. A continuación, mostramos sus ecuaciones:

$$m(\mathcal{V}) = 0,027 \frac{V}{mJ} - 0,027 \frac{V}{mJ} e^{[-4,696 \times 10^{-4} \frac{1}{V}] \mathcal{V}} \quad (3-2)$$

$$b(\mathcal{V}) = -1,366V + 1,392V e^{[-4,369 \times 10^{-4} \frac{1}{V}] \mathcal{V}} \quad (3-3)$$

Sustituyendo estas dos ecuaciones en la ecuación de la recta (ecu. ((3-1)), se obtiene:

$$A_M(E, \mathcal{V}) = \left[0,027 \frac{V}{mJ} - 0,027 \frac{V}{mJ} e^{[-4,696 \times 10^{-4} \frac{1}{V}] \mathcal{V}} \right] E + \left[-1,366V + 1,392V e^{[-4,369 \times 10^{-4} \frac{1}{V}] \mathcal{V}} \right] \quad (3-4)$$

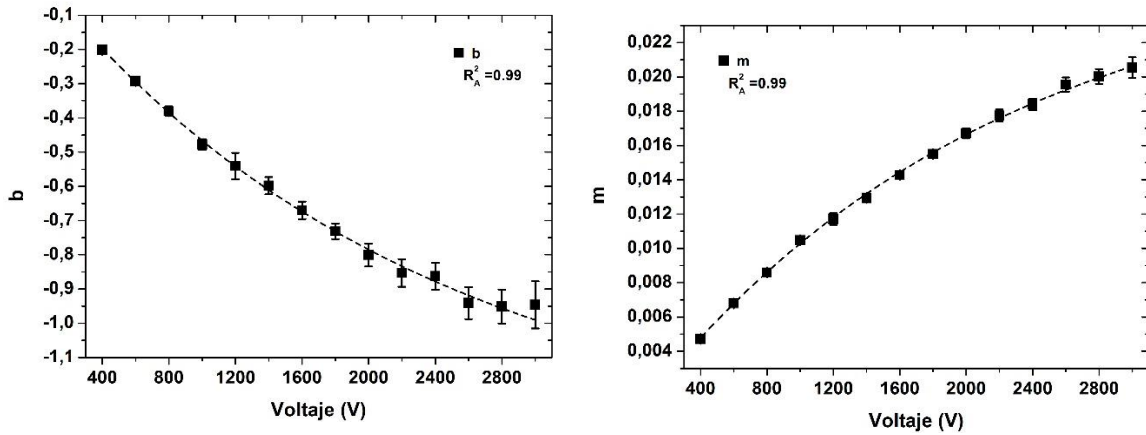


Figura 3.9. . Conjunto de datos y curvas de ajuste para cada uno de los coeficientes m y b .

Así esta ecuación puede reproducir los resultados obtenidos para la lente de 100 mm, en el rango de energía de [50 – 160] mJ. Además es posible obtener los valores intermedios, aunque no se hayan medido. En la Figura 3.10 se realizó una comparación de los datos que se obtuvieron experimentalmente y los datos que se generaron de la ecuación ((3-4). Para el caso de la Figura 3.10 a), donde se grafica la dependencia de la amplitud de la señal eléctrica con respecto a la energía, la obtención de los datos mediante la ecuación ((3-4) se obtuvo de la siguiente manera: se fija un valor del voltaje y después se varía la energía. Aquí se utilizaron los mismos valores de energía del experimento.

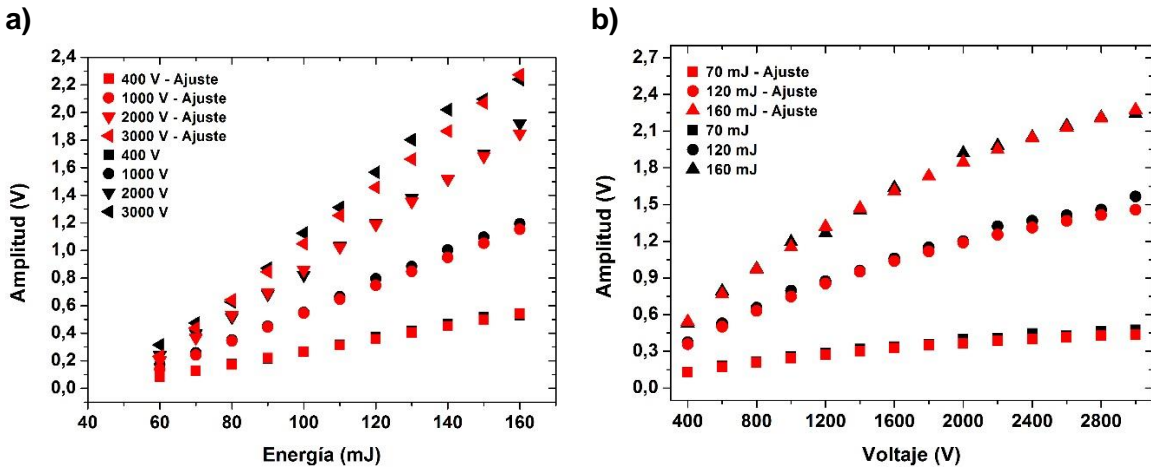


Figura 3.10. Gráficos de $A_M(E)$ y $A_M(V)$ para una distancia focal de 100 mm, considerando los datos experimentales (negro) y los datos obtenidos de la ecuación $A_M(E, V)$.

Considerando los resultados aquí presentados es posible concluir que el sistema desarrollado permite conocer los valores de energía de los pulsos de láser, convirtiéndose en una instrumentación que sirve como medidor de energía.

3.4 Duración del pulso

En la sección 2.2.3.2, se mencionó que se aplicaría la integración en el tiempo a las señales eléctricas como uno de los procesamientos. La integral aplicada (**Integral(t')**) es una función acumulativa de los valores de la amplitud de la señal eléctrica ($A(t)$), desde el $t = 0$ hasta t' y se expresa como:

$$\text{Integral}(t') = \int_0^{t'} A(t) dt$$

En la ecuación anterior solo se realiza la integral sin incluir la constante de permitividad eléctrica como en la ecuación (1-9), porque aquí solo se quiere conocer los valores relativos de la integral. No obstante si se quiere conocer el cambio del flujo de campo eléctrico ($\Delta\Phi_{ep}$) se debería multiplicar por $\frac{1}{\epsilon_0 R^2}$. En la Figura 3.11, se muestra la integral de tres señales eléctricas (normalizadas), cada una con los siguientes parámetros: a) y b) energía: 140 mJ, voltaje: 3000 V y f : 75 mm; c) energía: 340 mJ, voltaje: 400 V y f : 250 mm; y d) energía: 570 mJ, voltaje: 400 V y f : 500 mm. En a) se muestra la integral completa de la señal eléctrica comparada con el pulso óptico y en b), c) y d) se muestra un acercamiento de las señales eléctrica y óptica. Como se puede observar en los cuatro gráficos, la función de la integral tiene una coincidencia adecuada con la primera mitad del pico de la señal óptica.

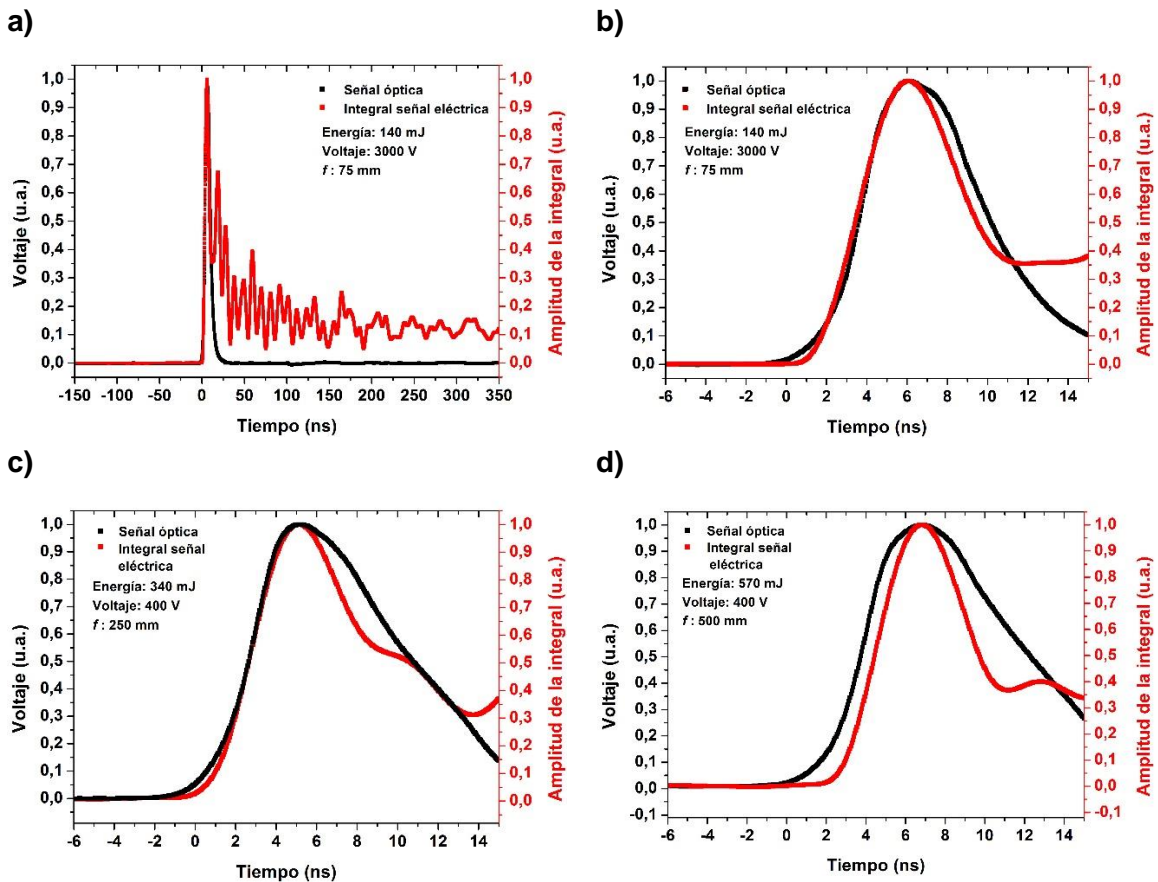


Figura 3.11. Integrales en el tiempo de las señales eléctricas con diferentes parámetros comparado con el pulso óptico.

Dado que para los cuatro gráficos sigue habiendo una correspondencia buena entre estas dos señales, nos indica que la interpretación física dada en mi tesis de licenciatura sigue siendo válida sin importar la distancia focal utilizada en las lentes.⁶

Por otro lado, observamos que mediante la integración es viable medir la duración del pulso de láser, usando la subida del primer pico de la integral. Lo anterior, dado que para distancias focales mayores a 100 mm la bajada de la integral ya no empata adecuadamente. En la siguiente tabla se muestra las mediciones de la duración de pulso de las señales de la Figura 3.11:

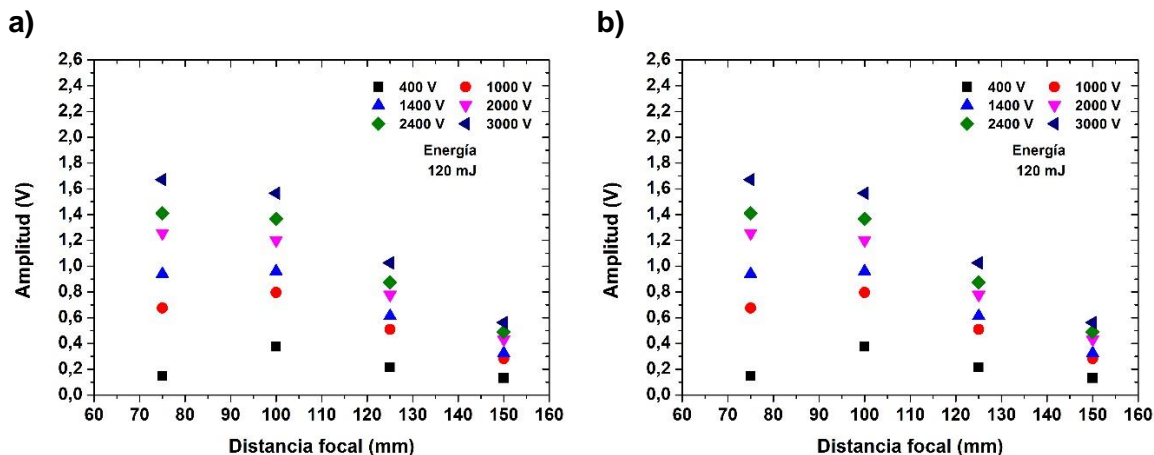
Señal eléctrica	Tiempo (Integral señal eléctrica)	Tiempo (señal óptica)
a) f : 75 mm Energía: 140 mJ Voltaje: 3000 V	5,4 ns	5,0 ns
c) f : 250 mm, Energía: 340 mJ Voltaje: 400 V	4,9 ns	5 ns
d) f : 500 mm, Energía: 570 mJ Voltaje: 400 V	4,9 ns	6 ns

Tabla 3.1. Mediciones de la duración de los pulsos mediante la integral de la señal eléctrica.

Se realizaron las mediciones de duración de pulso del láser utilizando como referencia la señal óptica. Se midió la mitad de la anchura a media altura de ambas señales. La anchura a media altura es lo que se conoce en inglés como Full Width at Half Maximum (FWHM). En nuestro trabajo para calcular el tiempo de duración del pulso de láser multiplicamos lo medido anteriormente por 2. Esto teniendo en cuenta que para la integral de la señal eléctrica se utilizó solo la subida del primer pico. Entonces, se obtuvo para el gráfico a) un error entre ambas mediciones de la duración del pulso de un 7.4%, para c) un error de 2% y para d) un error del 18.3% (ver Tabla 3.1). Dado los resultados anteriores se puede concluir que el sistema desarrollado permite conocer la duración de los pulsos de láser.

3.5 Área de enfoque

Uno de los objetivos de este trabajo fue determinar la distancia focal haciendo uso de la variación de los distintos parámetros experimentales. El primer intento fue relacionar la amplitud de la señal eléctrica con la distancia focal de manera biunívoca. Para encontrar esta relación se necesita encontrar una función monótona creciente o decreciente y con una derivada significativa. Por desgracia no fue posible cumplir este requisito de manera general, por ejemplo, como se muestra en la Figura 3.12. Lo más cercano a lo descrito anteriormente sería la Figura 3.12 a), donde si se acota a las mediciones de las lentes con distancia focal entre 100 mm y 150 mm tendríamos una función monótona decreciente y biunívoca. Sin embargo, en los demás gráficos (b) y c)) no se muestra funciones monótonas y biunívocas.



c)

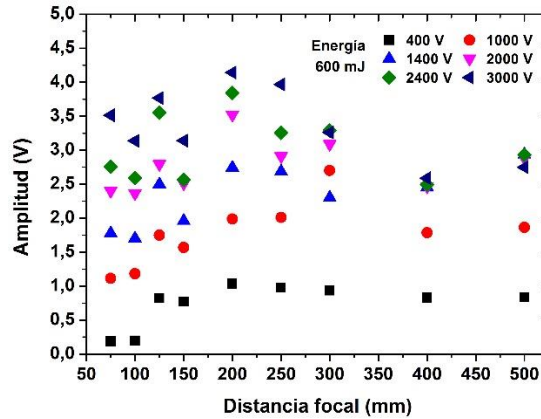


Figura 3.12. Gráficos de la amplitud de la señal eléctrica contra la distancia focal para tres energías.

Este hecho hace difícil poder determinar con la amplitud de la señal eléctrica la distancia focal utilizada. Sin embargo, podemos utilizar los resultados obtenidos en la sección 3.3. Si tenemos una lente que conocemos su distancia focal y definimos su valor de energía, por ejemplo, energía = 110 mJ y variamos el voltaje podemos encontrar una curva característica como las que se muestran en la ver Figura 3.13 a) para las distancia focales de 75 mm, 125 mm y 150 mm. De esta manera se puede tener una huella característica que permite determinar las distancias focales.

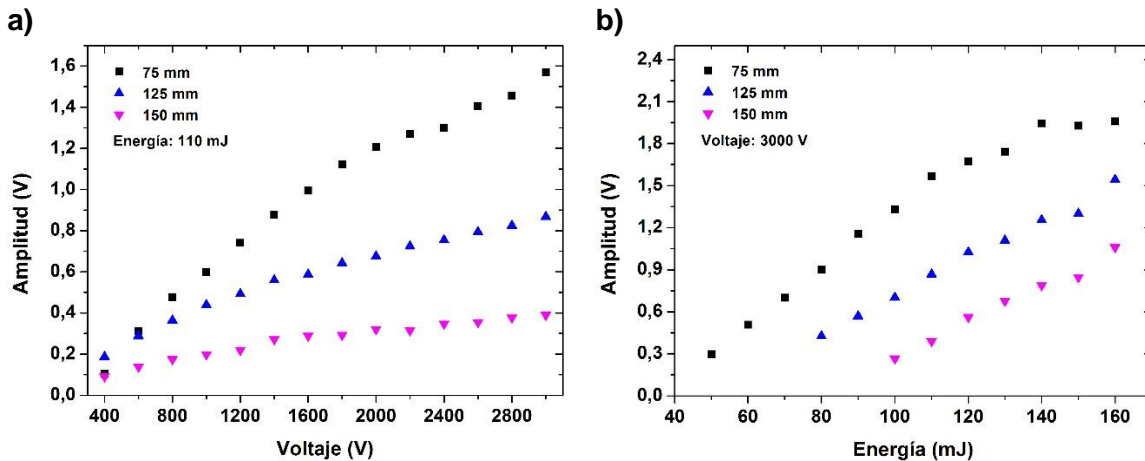


Figura 3.13. Gráficos de a) Amplitud en función del voltaje y b) Amplitud en función de la energía, para tres lentes con distancias focales distintas.

De igual manera haciendo uso de los resultados de esa misma sección, teniendo nuevamente una distancia focal específica y definiendo un voltaje, por ejemplo, voltaje = 3000 V y variando la energía de los pulsos del láser, también se obtienen curvas características como las de la Figura 3.13 b) de manera que es posible determinar la distancia focal. Pero, para propósitos experimentales es más sencillo fijar una energía y variar el voltaje. Entonces, de esta manera se puede determinar la distancia focal de una lente. Así, se puede tener una base de datos de las curvas características para cada lente. Por otro lado, utilizando los datos obtenidos del medidor de energía y teniendo en cuenta que el voltaje no afecta la energía transmitida,⁶ se graficó la transmitancia en función de la distancia focal (ver Figura 3.14). Para cada uno de los rangos de energía se observa que tenemos una función monótona creciente, lo que nos permite identificar la distancia focal sin importar el voltaje o la energía utilizada. Por otro lado, se observa que para lentes con

distancia focal pequeña se tiene una transmitancia menor comparada con las lentes de mayor distancia focal.

Dado los resultados anteriores podemos concluir que el sistema desarrollado permite conocer la distancia focal utilizada. Agregando este resultado a los dos anteriores podemos decir que se tiene una instrumentación que nos permitiría conocer la densidad de potencia.

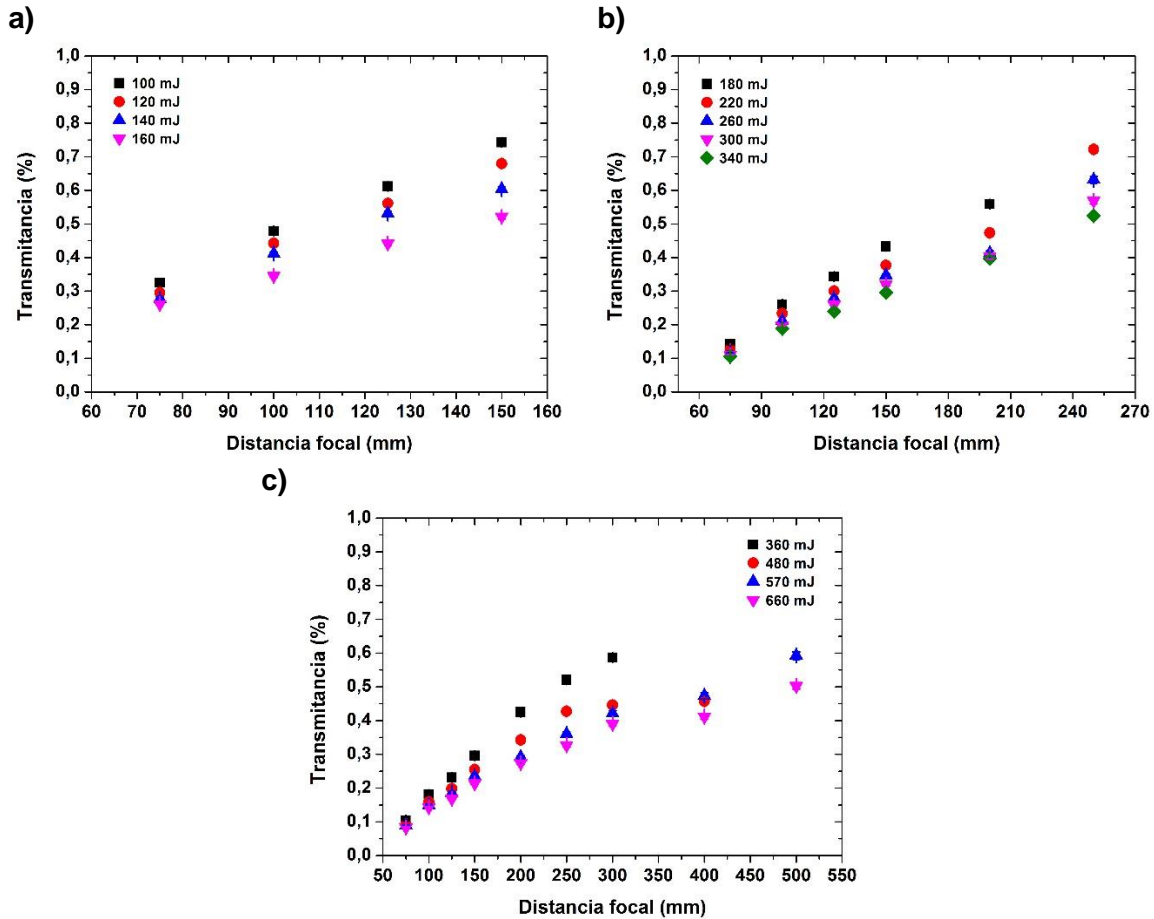


Figura 3.14. Gráficos de la transmitancia en función de la distancia focal para los tres rangos de energía.

4 Conclusiones

Como resultado del trabajo aquí realizado se desarrolló un sistema instrumental automatizado que permite el estudio de plasmas en aire producidos por pulso de láser, a la vez que sirve como medidor de energía y duración de los pulsos del láser, así como de la distancia focal de la lente utilizada. Esto fue posible mediante el análisis de la señal eléctrica, medición que se realizó a través de una resistencia de un circuito eléctrico y que es consecuencia de la formación del plasma producido en el aire en el interior del par de placas paralelas, y la energía transmitida, que es la energía de los pulso de láser que no se absorbió ni esparció resultado de la formación del plasma. La automatización del sistema permite realizar de manera eficiente la adquisición, transferencia y procesamiento de los datos. Es importante hacer énfasis que sin el proceso de automatización este trabajo no se hubiera realizado en los tiempos estipulados por el posgrado de Ingeniería de la Universidad Nacional Autónoma de México, debido a la gran cantidad de experimentos realizados.

Como anteriormente se mencionó, el sistema instrumental puede ser utilizado como un medidor de energía de pulsos de láser. Esto se debe a que del análisis de la energía, se observa un comportamiento creciente. Aunque, de los tres rangos de energía estudiados ([50,160] mJ, [180,340] mJ y [360,660] mJ), el de [50,160] mJ muestra mayor potencial para este uso, debido a que se tiene una pendiente positiva con mayor inclinación. Por otro lado, el voltaje aplicado al par de placas aumenta la amplitud de la señal eléctrica, lo que permite mejorar la resolución del medidor de energía. La transmitancia de los pulsos de láser tiene un comportamiento creciente y monótono con respecto al cambio de la distancia focal de las lentes. Lo anterior abre la posibilidad de usar la transmitancia como indicador de la distancia focal que se está utilizando. No obstante, no hay una tendencia definida del estudio de la señal eléctrica y el cambio de la distancia focal. Asimismo, se puede medir la duración de los pulsos del láser utilizando la señal eléctrica, esto sin importar los parámetros experimentales utilizados (energía de los pulsos del láser, voltaje, distancia focal).

Los resultados anteriores abren la posibilidad de construir un sistema que nos permite medir de manera indirecta la densidad de potencia en el punto de formación el plasma.

Asimismo, se tiene una relación señal a ruido de 3 a 1 con la amplitud de la señal eléctrica, que es adecuada para los estudios realizados en este trabajo.

Por otro lado, del análisis del voltaje y la energía de los pulsos del láser se desarrolla una ecuación que relaciona estos dos parámetros. La ecuación es particular, ya que depende del rango de energía en el que se esté trabajando y de la distancia focal utilizada.

5 Trabajos futuros

A lo largo de este trabajo se encontraron detalles que no pudieron ser resueltos debido a los tiempos de realización. Entonces los detalles que se abordaran en un futuro tienen que ver con la parte instrumental:

- Del análisis de las señales se observó que los rangos de frecuencias a trabajar generan fenómenos de inductancia. Por lo que se requiere de un análisis de la instrumentación (circuito) más profundo para entender las oscilaciones de la señales electricas. Se realizará un estudio profundo de los circuitos a altas frecuencias.
- Desarrollar el circuito impreso que nos ayude a reducir los ruidos externos.

Una vez afinado los detalles experimentales, se buscará variar parámetros de los que depende el plasma que nos permitirá extender el estudio. Entonces:

- Se utilizará diferentes longitudes de onda para la generación del plasma: 532 nm y 355 nm.
- Se usarán diferentes retrasos del Q-switch y se observará si guarda una relación el ancho del pulso del láser y el ancho del primer pico de la señal eléctrica.
- Se utilizará una cámara de presión para poder variar este parámetro y observar la dependencia que tiene con la señal eléctrica.

6 Referencias

- [1] A. W. Miziolek, V. Palleschi, and I. Schechter, *Laser-Induced Breakdown Spectroscopy (LIBS) Fundamentals and Applications*. 2006.
- [2] F. F. Chen, *Introduction to Plasma Physics and Controlled Fusion*. 2016.
- [3] E. K. Damon and R. G. Tomlinson, "Observation of ionization of gases by a ruby laser," *Appl. Opt.*, vol. 2, no. 5, pp. 546–547, 1963, doi: 10.1364/ao.2.000546.
- [4] J. Kiefer, J. W. Tröger, Z. S. Li, and M. Aldén, "Laser-induced plasma in methane and dimethyl ether for flame ignition and combustion diagnostics," *Appl. Phys. B Lasers Opt.*, vol. 103, no. 1, pp. 229–236, 2011, doi: 10.1007/s00340-010-4338-6.
- [5] M. V. Muniz, H. Sobral, C. S. Aké, R. Sanginés De Castro, E. Sterling, and F. Bredice, "Diagnostics techniques of plasmas produced by laser ablation," *AIP Conf. Proc.*, vol. 759, pp. 14–27, 2005, doi: 10.1063/1.1928154.
- [6] U. Vargas Nolasco, "Estudio de un plasma generado por un pulso láser de nanosegundos en aire mediante señales eléctricas producidas por la perturbación del campo eléctrico de un capacitor.," Universidad Nacional Autónoma de México, 2020.
- [7] S. Orazio, *Principles of Lasers*. 2010.
- [8] J. Martínez de Mendivil Varas, "Láseres integrados en dobles tungstos y niobato de litio basados en guías de onda ridge.," Universidad Autónoma de Madrid, 2015.
- [9] R. Rodríguez-Gonzalez, "Automatización de un sistema de irradiación de superficies con láser pulsado e implementación de monitoreo óptico del proceso a tiempo real.," Universidad Nacional Autónoma de México, 2019.
- [10] E. Optics, "Gaussian beam propagation." <https://www.edmundoptics.com/knowledge-center/application-notes/lasers/gaussian-beam-propagation/> (accessed Jul. 20, 2022).
- [11] M. Young, M. Hercher, and C. Y. Wu, "Some characteristics of laser-induced air sparks," *J. Appl. Phys.*, vol. 37, no. 13, pp. 4938–4940, 1966, doi: 10.1063/1.1708169.
- [12] J. M. Paulin-Fuentes, C. Sánchez-Aké, F. O. Bredice, and M. Villagrán-Muniz, "Quadrupole distribution generated by a laser induced plasma (LIP) in air in earliest instants using pulses of 532 or 355 nm," *J. Phys. D. Appl. Phys.*, vol. 48, no. 28, 2015, doi: 10.1088/0022-3727/48/28/285204.
- [13] L. J. Radziemski, T. R. Loree, D. A. Cremers, and N. M. Hoffman, "Time-Resolved Laser- Induced Breakdown Spectrometry of Aerosols," *Anal. Chem.*, vol. 55, no. 8, pp. 1246–1252, 1983.
- [14] J. M. Paulin Fuentes, "Diagnósticos Eléctricos y Magnéticos de un Plasma Inducido por Láser en Aire," 2014.

- [15] L. J. Radziemski, "From LASER to LIBS, the path of technology development," *Spectrochim. Acta - Part B At. Spectrosc.*, vol. 57, no. 7, pp. 1109–1113, 2002, doi: 10.1016/S0584-8547(02)00052-6.
- [16] V. P. Ageev *et al.*, "Electric field of a plasma generated by optical breakdown in air," *Zhurnal Eksp. i Teroreticheskoi Fiz.*, vol. 76, no. January, pp. 158–163, 1979.
- [17] J. Mauricio Paulin Fuentes, C. Sánchez-Aké, F. O. Bredice, and M. Villagrán-Muniz, "Dipolar distribution generated by laser-induced plasma (LIP) in air in earliest instants," *J. Phys. D. Appl. Phys.*, vol. 46, no. 49, 2013, doi: 10.1088/0022-3727/46/49/495202.
- [18] A. Robledo-Martinez, H. Sobral, and M. Villagrán-Muniz, "Dipolar field and plasma expansion at the onset of laser-induced breakdown in a uniform dc field," *J. Phys. D. Appl. Phys.*, vol. 37, no. 20, pp. 2819–2826, 2004, doi: 10.1088/0022-3727/37/20/008.
- [19] T. X. Phuoc, "An experimental and numerical study of laser-induced spark in air," *Opt. Lasers Eng.*, vol. 43, no. 2, pp. 113–129, 2005, doi: 10.1016/j.optlaseng.2004.07.003.
- [20] V. I. Konov, P. I. Nikitin, and A. M. Prokhorov, "Air breakdown near the target under the action of two successive CO₂-laser pulses, and magnetic fields," *Bull. Academy Sci. Phys. c/c Izv. Akad. Nauk Seriya Fiz.*, vol. 48, no. 6, pp. 1208–1213, 1985.
- [21] B. Barbieri and N. Beverini, "Optogalvanic spectroscopy," *Rev. Mod. Phys.*, vol. 62, no. 3, pp. 603–644, 1990, doi: 10.1103/RevModPhys.62.603.
- [22] C. Sánchez-Aké, F. Bredice, and M. Villagrán-Muniz, "Electric field-perturbation measurement of the interaction between two laser-induced plasmas," *Rev. Sci. Instrum.*, vol. 83, no. 2, 2012, doi: 10.1063/1.3683453.
- [23] S. Soubacq, P. Pignolet, and S. Mendonça, "Effective laser field influence on the transverse triggering of a spark gap in air," *J. Phys. D. Appl. Phys.*, vol. 35, no. 16, pp. 1955–1960, 2002, doi: 10.1088/0022-3727/35/16/309.
- [24] A. Robledo-Martinez, H. Sobral, and M. Villagrán-Muniz, "Measurement of the electrical size of a laser-induced plasma in a uniform field," *Appl. Surf. Sci.*, vol. 248, no. 1–4, pp. 32–35, 2005, doi: 10.1016/j.apsusc.2005.03.095.
- [25] D. Okano, "Detection of long-lived electrons around CO₂-laser-produced plasma in atmospheric gases using symmetric double-probe system," *IEEE Trans. Plasma Sci.*, vol. 35, no. 4 III, pp. 1143–1148, 2007, doi: 10.1109/TPS.2007.897904.
- [26] F. O. Bredice, D. J. O. Orzi, D. Schinca, H. Sobral, and M. Villagrán-Muniz, "Characterization of pulsed laser generated plasma through its perturbation in an electric field," *IEEE Trans. Plasma Sci.*, vol. 30, no. 6, pp. 2139–2143, 2002, doi: 10.1109/TPS.2002.807331.
- [27] H. Sobral, M. Villagrán-Muniz, and F. Bredice, "Energy balance in laser ablation of metal targets," *J. Appl. Phys.*, vol. 98, no. 8, 2005, doi: 10.1063/1.2108154.
- [28] F. Bredice *et al.*, "Real time measurement of the electron density of a laser generated plasma using a RC circuit," *Spectrochim. Acta - Part B At. Spectrosc.*, vol. 62, no. 8, pp. 836–840, 2007, doi: 10.1016/j.sab.2007.06.011.
- [29] P. Pacheco *et al.*, "Real time determination of the laser ablated mass by means of electric field-perturbation measurement," *Spectrochim. Acta - Part B At. Spectrosc.*, vol. 142, pp. 50–54, 2018, doi: 10.1016/j.sab.2018.02.003.
- [30] J. Miranda, "Evaluación de la incertidumbre en datos experimentales," no. June 2000, p. 43, 2000.
- [31] University of Pennsylvania Department of Physics & Astronomy, "Averaging , Errors and Uncertainty," *Undergrad. Labs*, no. the X, p. 8, 2015, [Online]. Available:

- http://virgo-physics.sas.upenn.edu/uglabs/lab_manual/Error_Analysis.pdf.
- [32] C. G. Rosales Guzmán, “Medida de la energía liberada en un plasma de ablación láser,” Universidad Nacional Autónoma de México, 2007.
- [33] E. Córdoba Nieto, “Manufactura y automatización,” *Rev. Ing. e Investig.*, vol. 26, no. 3, pp. 120–128, 2006, [Online]. Available: <http://revistas.unal.edu.co/index.php/ingainv/article/view/14761>.
- [34] E. A. Rubio, “Control de instrumentos mediante el bus GPIB programado con Matlab,” Universidad Autónoma de Barcelona, 2010.
- [35] Universidad Europea de Madrid, “Fundamentos de computadores.” p. 22.
- [36] C. A. Vargas-Nolasco, “Desarrollo de plataforma para experimentos de ionización de gases por RF mediante el protocolo GPIB,” Universidad Nacional Autónoma de México, Cd. Mx., 2017.
- [37] A. L. Barragán, “Puertos USB - Bus Serie Universal y descripción de la norma IEEE 1349.” p. 18, 2002, [Online]. Available: <http://bibliotecadigital.usb.edu.co:2087/visor/17195>.
- [38] M. Periago de la Torre, “Compresión y comunicación de datos con un microcontrolador PIC,” Universitat Rovira I Virgili, 2007.
- [39] “USB in a NutShell,” 2018. <https://www.beyondlogic.org/usbnutshell/usb1.shtml#Introduction> (accessed Jun. 27, 2022).
- [40] C. W. Aldama-Reyna, J. D. Aldama-Guardia, and J. F. Agreda-Delgado, “Detector de energía de pulsos láser infrarrojo basado en el efecto fotoacústico,” *Rev. Física*, no. 56, pp. 87–103, 2018, doi: 10.15446/mo.n56.69828.
- [41] M. Elias, “Desarrollo de un sistema fotopiroeléctrico para la caracterización de propiedades térmicas de materiales,” Universidad Autónoma de Nuevo León, 2019.
- [42] S. Satapathy and V. K. Wadhawan, “Fabrication of pyroelectric laser-energy meters and their characterization using Nd:YAG laser of variable pulse-width,” *Sensors Actuators, A Phys.*, vol. 121, no. 2, pp. 576–583, 2005, doi: 10.1016/j.sna.2005.04.004.
- [43] U. Zammit, M. Marinelli, F. Mercuri, S. Paoloni, and F. Scudieri, “Invited review article: Photopyroelectric calorimeter for the simultaneous thermal, optical, and structural characterization of samples over phase transitions,” *Rev. Sci. Instrum.*, vol. 82, no. 12, 2011, doi: 10.1063/1.3663970.
- [44] J. E. González, L. Lira Cortés, Á. Sánchez Rodríguez, and et al., “Calorimetría Adiabática Y Sus Aplicaciones,” *Simp. Metrol.*, vol. 1, p. 7, 2010.
- [45] I. Sistemas Opticos, “Sensores de potencia y energía láser.” https://www.iberoptics.com/es/sensores-de-potencia-y-energia-laser_211 (accessed Jun. 28, 2022).
- [46] A. I. Ferguson, “A simple and inexpensive laser power meter meter,” *J. Phys. E Sci. Instrum.*, vol. 17, pp. 4–6, 1984.
- [47] P. Pinot and Z. Silvestri, “Optical power meter using radiation pressure measurement,” *Meas. J. Int. Meas. Confed.*, vol. 131, no. July, pp. 109–119, 2019, doi: 10.1016/j.measurement.2018.07.087.
- [48] Surelite, “Operation and Maintenance Manual for Surelite Lasers,” 2000.
- [49] Stanford Research Systems, “MODEL DG535 Digital Delay / Pulse Generator,” 2006.
- [50] Altechna, “Watt Pilot Motorized Attenuator User Manual,” pp. 1–43, 2015.
- [51] Stanford Research Systems, “Series Ps300. High Voltage, Power Supplies,” 2007.

- [52] OPHIR Photonics, "User Commands," 2017.
- [53] Tektronix, "MSO/DPO5000, DPO7000/C, DPO70000/B/C, DSA70000/B/C, and MSO70000/C Series Digital Oscilloscopes Programmer Manual." .
- [54] Matlab, "MathWorks." <https://la.mathworks.com/products/matlab.html> (accessed Jul. 01, 2022).

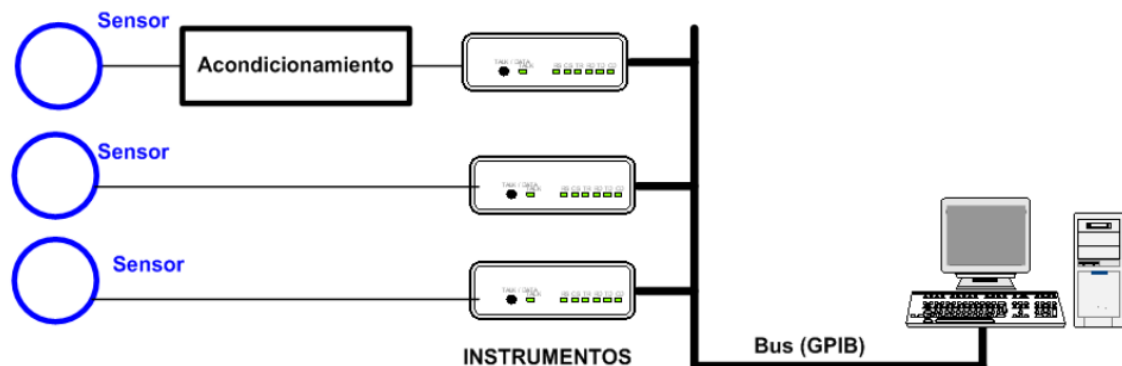
7 Apéndices

7.1 Automatización

El concepto de automatización (del griego *autos* que significa “por sí mismo” y *maiomai* que significa “lanzar”) concierne a la necesidad de minimizar la intervención humana en los procesos de producción, vale decir, ahorrar esfuerzo laboral.³³ En la actualidad, una buena parte de los procesos realizados en la industria se realizan de forma automatizada. Estos procesos se encomiendan a maquinas y ordenadores, que permiten realizarlos de una manera más eficiente y eficaz que en la manera que suele realizarlos el hombre.

El concepto de automatización no es exclusivo de la tecnología. En la actualidad, la automatización está inmersa en la ciencia, debido que surge la necesidad de mejorar la capacidad de percibir la realidad valiéndose de cualquier tipo de técnicas y sistemas. Y así, poder medir variables físicas de forma más eficiente. Los instrumentos utilizados para obtener la información de un determinado proceso son los sistemas de adquisición de datos. Dichos sistemas pueden estar basados en instrumentos de adquisición independientes o sistemas modulares (ver Figura 7.1 a) y b) respectivamente). Los sistemas de adquisición de datos pueden funcionar de forma autónoma, conectarse a otros instrumentos o bien a un computador a través de un bus de comunicación.

a)



b)

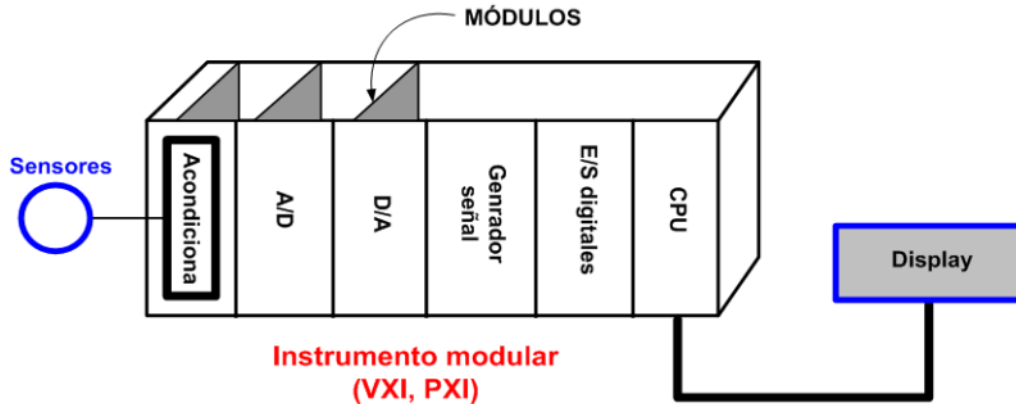


Figura 7.1. Sistema de adquisición de datos. a) Instrumentos de adquisición independientes. b) Sistema modular. Imágenes tomada de un trabajo de Rubio.³⁴

7.1.1 Buses de comunicación

Los buses son elementos importantes en un computador, tienen la función de realizar la transferencia de datos. En informática, un bus es un camino de comunicación entre uno, dos o más dispositivos.³⁵ En esta sección nos centraremos en dos específicamente, el bus *GPiB* y el *USB*.

7.1.1.1 GPiB

A finales de los años 60's, *Hewlett-Packard* (HP) con el propósito de comunicar y administrar de manera segura instrumentos de medición y testeo fabricados por ellos mismos, diseñó un canal de comunicación de corto alcance nombrado HP-IB o como actualmente se le conoce *GPiB* (*General Purpose Interface Bus*). Así, empezó una época donde la proliferación de controladores digitales y específicamente de equipos de pruebas programables por parte de diferentes compañías, obligó al desarrollo de una interfaz de alta velocidad para poder establecer comunicación entre los instrumentos.³⁶ En 1975, el *Institute Electrical and Electronics Engineers* (o mejor conocido como la *IEEE*) estandarizó el protocolo, dando especificaciones eléctricas, mecánicas y funcionales del sistema de la interfaz, bajo el nombre de *ANSI/IEEE 488-1975*.³⁶

El protocolo *IEEE 488* (o como lo renombraron *IEEE 488.1*), no especifica normas de sintaxis, ni convenciones de formato para la programación de instrumentos. Por lo que se le agregó un apéndice conocido como *IEEE 488.2*. Éste incluyó formatos, protocolos y comandos comunes para ser utilizados con el *IEEE 488.1*. En 1990, se incluyó en las especificaciones del *IEEE 488.2* un documento de los Comandos Estándares para Instrumentación Programable (*SCPI* por sus siglas en inglés), donde se definen los comandos que todo instrumento debe obedecer, sin importar el fabricante.

El objetivo principal del bus *GPiB* es administrar la información que se transfiere entre dos o más dispositivos en una misma red de comunicación.

7.1.1.1.1 Tipos de dispositivos y topología de conexión

Existen tres tipos diferentes de dispositivos dentro de una red de comunicación *GPiB*: hablantes, escuchas y controladores. Cada uno tiene una función específica, pero no quiere decir que un dispositivo no pueda realizar más de un papel en una red de comunicación.

El dispositivo controlador (*controller*) se encarga de supervisar todas las operaciones. Define qué tipo de dispositivo envía información, quien recibe, en que momento deben hacerlo y hacerles saber con quién comunicarse cada dispositivo.³⁶

Al enviar un mensaje de un dispositivo a otro, se dice que un dispositivo hablante (*talker*) envía un mensaje a un dispositivo escucha (*listener*). En esta parte es donde el dispositivo controlador administra el flujo de información. En una red puede haber varios escuchas activos, pero solo un hablante activo. La función del controlador también es establecer la comunicación entre dos dispositivos asignando direcciones, para determinar quién habla y quien escucha. Las direcciones de cada dispositivo conectado al bus *GPIB* deben ser exclusivas. Las direcciones están formadas por dos partes: la dirección primaria y la dirección secundaria. La primera le corresponde un número identificador entero entre 0 y 30. La segunda configura el estado del dispositivo, es decir, modo hablante (*talker* TA), escucha (*listener* LA) o no escucha (0). Este último estado se refiere a que no está en estado hablante ni escucha.

Posición de bit	7	6	5	4	3	2	1	0
Significado	0	TA	LA	Dirección primaria <i> GPIB </i> (de 0 a 30)				

Tabla 7.1. Bits de las direcciones del bus *GPIB* .

Por otro lado, en una red de transmisión de datos que utiliza el bus *GPIB* se debe cumplir con cierto requerimientos para que su velocidad sea óptima (hasta 8 Mbyte/s). Estos son los siguientes:

- Debe haber un número máximo de 15 dispositivos conectados. Dos tercios de estos deben estar encendidos.
- La longitud de bus entre dos dispositivos debe ser máximo de 4 metros.
- La longitud máxima total no debe ser mayor a los 20 metros.

Las configuraciones para crear la red basada en el bus *GPIB* son: topología estrella, lineal e híbrida combinación de las dos anteriores (ver Figura 7.2).

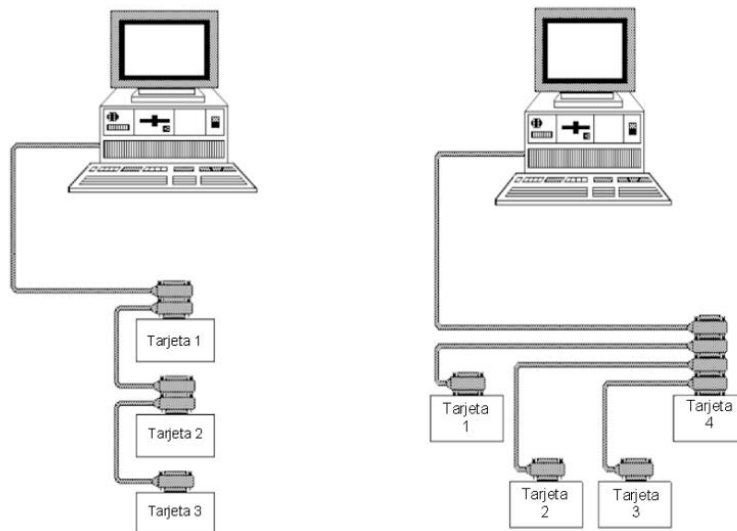


Figura 7.2. Topología lineal (izquierda). Topología estrella (derecha). Figura tomada de un trabajo de Vargas.³⁶

7.1.1.1.2 Mensajes y protocolo de transferencia

Los dispositivos *GPIB* se comunican con otros dispositivos *GPIB* enviando dos tipos de mensajes: ³⁶

- Mensajes de dispositivo: a menudo llamado mensaje de datos, contienen información respecto a un dispositivo específico. Puede ser estados de los instrumentos, archivos de datos, configuraciones del instrumento o resultados de medición.
- Mensaje de gestión: son conocidos como “mensaje de comandos” o “comandos”. Se encargan de administrar el flujo de mensajes del bus. Estos comandos habilitan o deshabilitan dispositivos, los inicializan, preguntan el estado del dispositivo, etc. Los comandos más comunes son los siguientes:

Comando	Nombre de la orden	Función
*CLS	Clear Status Command	Despejar el registro de estado y los registros de incidencia.
*ESE	Event Status Enable Command	Habilita bits del registro de habilitación de incidencias.
*ESE?	Event Status Enable Query	Interroga el registro de habilitación de incidencias estándar.
*ESR?	Event Status Register	Interroga el registro de incidencias estándar.
*IDN?	Identification Query	Identifica el tipo de instrumento y versión de software.
*LRN?	Learn Device Setup	Devuelve el estado actual del equipo.
*OPC	Operation Complete Command	Fija bit de “Operación completa” del registro estándar.
*OPC?	Operation Complete Query	Responde con “1” si se han ejecutado órdenes previas.
*OPT?	Option Identification Query	Devuelve las opciones instaladas en el equipo. (Ejemplos: memoria extendida, Seguridad).
*RCL	Recall Command	Restaura el estado del equipo del registro save/recall.
*RST	Reset Command	Sitúa al equipo en el estado básico de referencia.
*SAV	Save Command	Almacena el estado actual en el registro save/recall.
*SER	Service Request Enable Command	Habilita los bits del registro de habilitación de Byte de estado.
*SER?	Service Request Enable Query	Devuelve el contenido del registro SER de habilitación del byte de estado.
*STB?	Read Status Byte Query	Devuelve el estado del registro resumido del Byte de estado.
*TRG	Trigger Command	Arranca o dispara la operación del equipo de forma remota.
*TST?	Self-Test Query	Devuelve el resultado del autotest del equipo.
*WAI	Wait-to-Continue Command	Espera a que se realicen todas las operaciones pendientes.

Tabla 7.2. Tabla de comandos comunes del protocolo *GPIB*.³⁶

Para la transferencia de información, el bus *GPiB* consta de 24 líneas de transmisión (ver Figura 7.3), de estas 16 líneas corresponden a la interfaz *GPiB* y las 8 restantes a las líneas de retorno a tierra. De las 16 líneas, 8 son de datos (información que queremos transmitir) y 8 para comandos (mensajes de control y estados de los dispositivos). De estas últimas 8 líneas de comandos, 3 son para el control de transferencia de datos (*handshake*) y 5 para el control general de la interfaz.³⁴ Las 8 líneas de datos DIO1-DIO8 pueden transportar tanto datos como comandos. Transfieren la información de un byte a la vez (1 bit por cada línea).^[34,36]

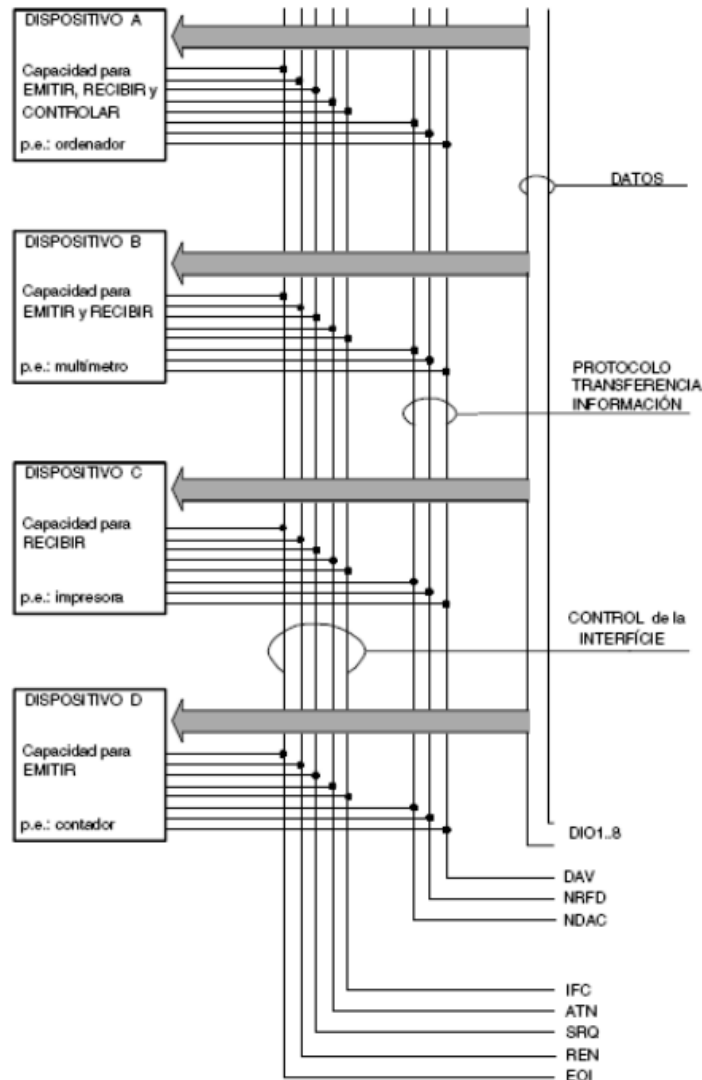


Figura 7.3. Líneas y señales del bus *GPiB*. Imagen tomada de un trabajo de Vargas.³⁶

Las 5 líneas de control de la interfaz aseguran la administración y control del flujo de datos del bus. Cada una de estas líneas cumple una función específica, como se muestra a continuación:

- **ATN (ATtention):** Se usa para la distinción entre datos e instrucciones. Si se encuentra en nivel bajo son instrucciones o direcciones, si está en nivel alto los datos son un comando de instrumento.

- **IFC (InterFace Clear)**: Se utiliza para dar un reset a los parámetros y dirección del bus.
- **REN (Remote ENable)**: Se utiliza para establecer el control remoto de un instrumento. Cuando REN está en alto los instrumentos bloquean el modo local. Sin embargo, si éstos cuentan con el botón LOCAL, será la excepción y no estará inhabilitado.
- **EOI (End Or Identify)**: Es usado como indicador de que la transferencia total de datos ha terminado.
- **SRQ (Service ReQuest)**: Se utiliza para solicitar turno o algún servicio del controlador.

Las 3 líneas de control de transferencia de datos (*handshake*) sirven para dotar de fiabilidad y credibilidad la transmisión de datos del bus *GPiB*. Una de estas tres líneas es controlada por el hablante, las otras dos por el o los dispositivos escuchas. La manera en la que las líneas de control de transferencia de datos garantizan la fiabilidad de los datos es estableciendo como velocidad de transmisión la que corresponde al dispositivo más lento de la red *GPiB*. Así, se logra que la transmisión sea lo suficientemente lenta para que todos los instrumentos reciban correctamente la información.³⁶ Las 3 líneas de control son las siguientes:

- **DAV (DAta Valid)**: El hablante controla esta línea para hacer saber al o los dispositivos escucha si la información está en condiciones de ser transmitida a través del bus.
- **NRFD (Not Ready For Data)**: El escucha controla esta línea para hacer saber al dispositivo hablante hasta qué momento debe retener la información para enviarla.
- **NDAC (Not Data ACcepted)**: Esta línea está controlada por el dispositivo escucha. El estado de NDAC hace saber al dispositivo hablante si ha recibido correctamente el byte.³⁷

7.1.1.1.3 Comandos

En la mayoría de los casos los comandos consisten en un encabezado y uno o más parámetros. Los encabezados y parámetros están separados por un espacio en blanco. Todos los caracteres son código ASCII. Como se mencionó anteriormente la norma SCPI consigue estandarizar los comandos de control y los formatos de datos de los instrumentos. La IEEE desarrolló un modelo general de instrumento que permite la compatibilidad de un conjunto de comandos para múltiples dispositivos. También, se estandarizó un modelo de bloques funcionales, que tiene asignado a su vez un conjunto de comandos. Así, un dispositivo está compuesto por subconjunto de bloques funcionales, cuya programación dependerá de estos bloques funcionales. En la Figura 7.4 se muestra el flujo de datos del modelo general. Las flechas continuas representan el flujo de datos, las flechas discontinuas representan el flujo de control.

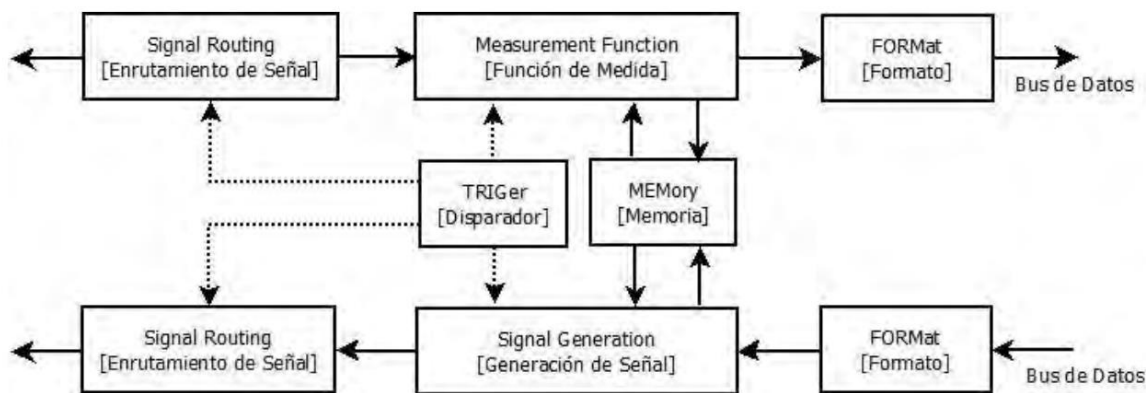


Figura 7.4. Modelo general de un dispositivo programable con comunicación GPIB, según la norma SCPI. Imagen tomada de un trabajo de Vargas.³⁶

A continuación, se describe cada uno de los bloques del modelo general:³⁶

FORMat: Se encarga de transformar los datos internos del instrumento en otras representaciones de datos, sobre los datos que llegan o salen, a transferirse por el bus. Un ejemplo podría ser la conversión de datos internos a ASCII.

Signal Generation: Es responsable de la conversión de datos a una señal física. Al igual que el bloque MEASurement, este bloque contiene sub-bloques: CALCulate, SOURce y OUTput. Estos bloques permiten la comunicación con los otros bloques principales como TRIGger y MEMory.

SOURce (subbloque): Se encarga de generar una señal con base en las características requeridas y el flujo de datos suministrados.

OUTput (subbloque): Se encarga de acondicionar la señal de salida, ya que ha sido generada. Se pueden realizar acondicionamientos como: filtrado, polarización, conversión de frecuencia y atenuación.

CALCulate (subbloque): Transforma los datos adquiridos en información específica para una aplicación concreta. Por ejemplo: amplitud, frecuencias, tiempos de subida, periodos, etc.

TRIGger: Se encarga de proveer al instrumento de medios para poder sincronizar con eventos externos.

MEMory: Permite al instrumento almacenar información. Hay información almacenada que puede ser inaccesible al usuario, como los datos de calibración interna.

MEASurement Function: Este bloque es el que realiza las mediciones; convierte una señal física en datos que después podrán ser procesados. El bloque está dividido en tres sub-bloques: INPut, SENSE y CALCulate (explicado anteriormente). No se muestran explícitamente los subbloques dentro del modelo general de un instrumento, debido a que existen instrumentos que también realizan la misma medida utilizando comandos distintos.

INPut (subbloque): El propósito de este bloque es adaptar la señal recibida antes de que esta sea transformada a datos.

SENSE (subbloque): Transforma las señales eléctricas ya adaptadas en datos que pueden ser manipulados por un ordenador.

Signal ROUTing: Representa la capacidad de ciertos instrumentos de direccionar sus señales de entrada a la entrada de otros circuitos internos. Los comandos se pueden encontrar en la jerarquía ROUTe.

Los comandos en la norma SCPI se agrupan de una forma específica, que usa una jerarquía de tipo árbol. En la raíz del árbol se encuentran los comandos mencionados en el párrafo

anterior. Cada comando se divide en ramas y es identificado por palabras claves que a su vez identifican a funciones subordinadas a la raíz, y así sucesivamente. La notación utilizada para los comandos es la siguiente: ^[34,36]

- : indican el paso a un nivel jerárquico inferior. Para clarificar la notación también se ha utilizado identificación de los niveles inferiores.
- [] Palabras claves opcionales.
- < > Encierran el tipo de parámetro.
- | separa parámetros opcionales (solo se puede poner uno de ellos).
- ; separa comandos que están en la misma línea (no cambia el nivel del último comando).
- , se usa para separar distintos parámetros dentro de un mismo nivel.
- ? Indica que es un comando de consulta y que se espera una respuesta del equipo al que se envía. Es muy importante leer el dato solicitado; de no ser así al enviar otro comando se crea una situación de error en el instrumento.

7.1.1.1.4 Controlador NI – VISA y NI – 488.2

Para poder realizar la comunicación del bus GPIB con la computadora se necesita una tarjeta controladora, para acceder a dicha tarjeta se utilizan las funciones propias del bus GPIB. Pero la utilización de estas funciones hace que al desarrollar aplicaciones para determinado instrumento sirvan únicamente para el bus y ese instrumento. Para solucionar este problema se desarrolló una API (*Application Programming Interface*) llamada VISA (por sus siglas en inglés *Virtual Instrument Software Architecture*). Este software permite utilizar, además del bus GPIB, otro tipo de buses. En este trabajo se utilizarán dos controladores desarrollados por la compañía *National Instrument (NI)*: ³⁴

- NI – VISA: esta es una API que proporciona una interfaz de programación para controlar instrumentos Ethernet/LXI, GPIB, seriales, USB, PXI y VXI en entornos de desarrollo de aplicaciones de NI como LabVIEW, LabVIEW NXG, LabWindows/CVI y Measurement Studio. La API se instala a través del controlador NI-VISA.
- NI-488.2: es un controlador de instrumentos de NI con varias utilidades que ayudan en el desarrollo y depuración de un programa de aplicación. NI-488.2 incluye comandos de alto nivel que manejan automáticamente toda la gestión del bus, por lo que no es necesario conocer los detalles de programación del producto de hardware GPIB o el protocolo IEEE 488.2. Los comandos de bajo nivel también están disponibles para una máxima flexibilidad y rendimiento.

7.1.1.2 Bus Serial Universal (USB, Universal Serial Bus)

El desarrollo del sistema de Bus Serial Universal (*USB*) surgió con el objetivo de eliminar la necesidad de adquirir tarjetas separadas para poner en los puertos bus ISA o PCI. Además, de mejorar las capacidades *plug-and-play* permitiendo a los dispositivos ser conectados o desconectados del ordenador sin necesidad de reiniciarlo. ^[34,37]

7.1.1.2.1 Definición y descripción general del *USB*

Las siglas *USB* corresponden a *Universal Serial Bus*, como el nombre indica es un sistema que permite la comunicación entre dispositivos electrónicos que sólo transmiten una unidad de información a la vez. En la actualidad, la clasificación de los sistemas *USB* se da a partir de su velocidad de transferencia de datos, en estos momentos se clasifican en 6:

- Baja velocidad (*USB 1.0*): la tasa de transferencia 1.5 Mbit/s (188 kB/s).

- Velocidad completa (*USB 1.1*): tiene una tasa de transferencia de 12 Mbit/s (1.5 MB/s).
- Alta velocidad (*USB 2.0*): tiene una tasa de transferencias de hasta 480 Mbit/s (60 MB/s).
- Velocidad superalta (*USB 3.0*): tiene una tasa de transferencia de hasta 4.8 Gbit/s (600 MB/s).
- Velocidad superalta + (*USB 3.1*): tiene una tasa de transferencia de datos de hasta 10 Gbit/s (1.25GB/s).
- Velocidad superalta + (*USB 3.2*): tiene una tasa de transferencia de datos de hasta 20 Gbit/s (2.5 GB/s).

En este trabajo se utilizó el *USB 2.0*. El *USB 2.0* es una versión que salió a mitad del 2000³⁸ y tiene como principal característica un aumento en la velocidad de transferencia de datos de casi 40 veces en comparación con la velocidad del *USB 1.1*. El *USB 2.0* incluye las características del *USB 1.1*. La forma de organización del *USB* es mediante una estructura de árbol descendente, con múltiples dispositivos conectados a un mismo bus, en la que los *hubs* (periféricos descritos más adelante) llevan las señales desde un dispositivo al *host* o viceversa. El *USB 2.0* cuenta con su propio controlador de *host*. El bus *USB* funciona a tres velocidades, baja (1,5 Mbit/s), completa (12 Mbit/s) y alta velocidad (480 Mbit/s). Además, el *USB 2.0* permite la conexión de hasta 127 dispositivos a un solo controlador. Todos los dispositivos conectados deben seguir ciertas reglas básicas y estandarizadas.³⁷

7.1.1.2.2 Host – USB

El computador o PC que soporta este tipo de bus se le denomina *host – USB*. Por otro lado, a todo periférico que utiliza este tipo de comunicación se denomina función – *USB* (periférico – *USB*).

Dentro del sistema *USB*, existe un solo *host*, que sirve de interfaz con los diferentes dispositivos. Su implementación es una combinación entre hardware y software. El *host-USB* puede proveer de puntos de conexiones iniciales en donde de forma ramificada se conectan los periféricos. A nivel de hardware el *host-USB* se encarga de los siguientes aspectos dentro del sistema *USB*:^[37,38]

- Detectar la conexión de nuevos dispositivos *USB* al sistema como la remoción de aquellos ya conectados, configurarlos y ponerlos a disposición del usuario, tarea que involucra acciones por software.
- Administrar y controlar el flujo de datos entre el *host* y los dispositivos *USB*.
- Administrar y regular los flujos de control entre el *host* y los dispositivos *USB*.
- Recolectar y resumir estadísticas de actividad y estado de los elementos del sistema.
- Proveer de energía eléctrica a los dispositivos que se abastecen del computador (teclado y ratón son dos ejemplos claros).

Por otra parte, a nivel de software algunas funciones del controlador del *host – USB* son:³⁷

- Enumeración y configuración de los dispositivos conectados al sistema.
- Administración y control de transferencias isocrónicas de información.
- Administración y control de transferencias asincrónicas.
- Administración avanzada del suministro eléctrico a los diferentes dispositivos.
- Administración de la información del bus y de los dispositivos *USB*.

7.1.1.2.3 Controlador del host

El controlador del *host* esta compuesto por el hardware y el software que permiten a los sistemas *USB* conectarse al *host*. En la arquitectura *USB* solo hay un controlador de *host* por cada bus, este es el agente encargado de comenzar la transferencia en el bus *USB*.³⁶

El software consiste en el driver del controlador del *host*, que interactúa con el hardware del mismo controlador a través de la interfaz hardware/software (HW/SW). La parte del hardware del controlador del *host* consiste en un hub raíz y los búfers de datos. Las funciones básicas del controlador son:³⁸

- Detectar la inserción y desconexión de sistemas *USB*.
- Gestionar el flujo de control entre el *host* y los sistemas *USB*.
- Gestionar el flujo de datos entre el *host* y los sistemas *USB*.
- Coleccionar estadísticas del estado y la actividad.
- Proveer de energía eléctrica a los sistemas conectados.

En la parte del hardware existen tres implementaciones estandarizadas de los controladores del *host*: el *Universal Host Controller Interface* (UHCI), definido por Intel; el *Open Host Controller Interface* (OpenHCI o OHCI), definido por Microsoft; y el *Enhanced Host Controller Interface* (EHCI), definido por Intel, Compaq, NEC, Lucent y Microsoft. Los dos primeros implementados para el *USB* 1.1 y el tercer para el *USB* 2.0.³⁹

7.1.1.2.4 Hub - *USB* y topología del *USB*

Los *hubs* son concentradores cableados que permiten múltiples conexiones simultáneas (ver Figura 7.5). El *hub* – *USB* tiene la capacidad de detectar si algún otro periférico fue conectado e informar al *host*. De igual forma los *hubs* tienen la capacidad de reconocer cuando un periférico es desconectado y enviar al *host* la señal para remover estructuras de datos y programas de administración del dispositivo retirado.^[37,38]

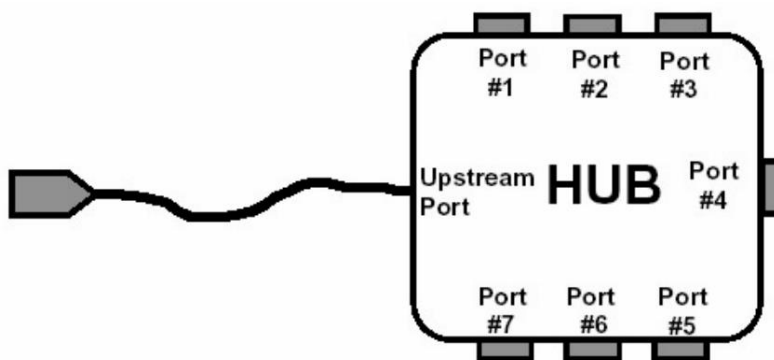


Figura 7.5. Esquema de hub típico. Imagen tomada del trabajo de Rubio.³⁴

El *hub* está conformado por dos partes: el controlador del *hub* y el repetidor de *hub*. El controlador se asemeja a un pequeño CPU, este realiza la supervisión de las múltiples funciones que realiza el *hub*. Por otro lado, el repetidor tiene la función de analizar, corregir y retransferir la información que llega al *hub* hacia los mismos puertos.^[37,38]

La interconexión física del *USB* tiene una topología de árbol descendente donde un *hub* es el centro de cada rama (ver Figura 7.6). Cada segmento o línea del diagrama representa una conexión entre el *host* y los *hubs*, o un *hub* conectado a otro *hub*.

Anteriormente se mencionó que se pueden conectar un máximo de 127 dispositivos, esto distribuido en un número máximo de siete capas, incluyendo la raíz (ver Figura 7.6).

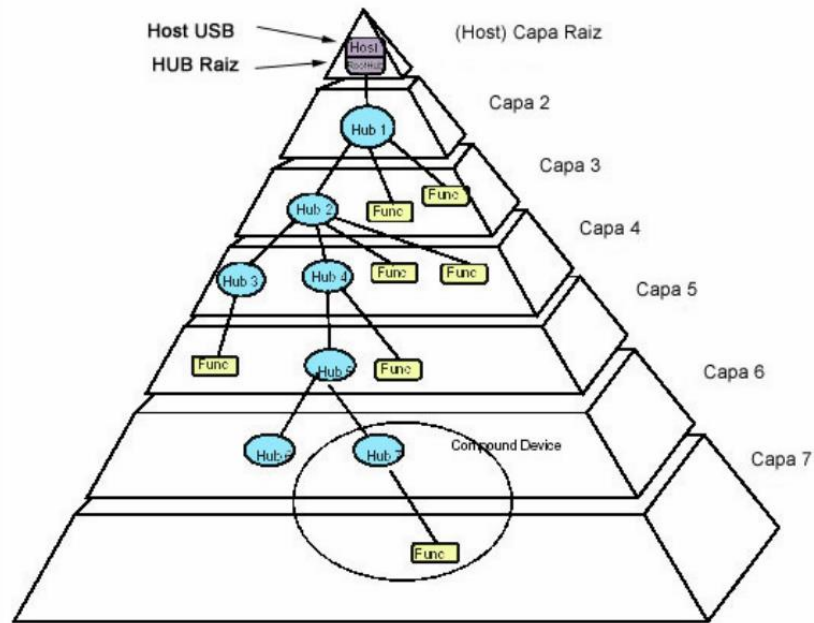


Figura 7.6. Estructura de capas del USB. Imagen tomada del trabajo de Periago.³⁹

7.1.1.2.5 Arquitectura general del USB y modelo lógico funcional

El USB está compuesto por un cable especialmente diseñado para la transmisión de datos entre la computadora y diferentes periféricos. Los dispositivos conectados siguen las normas de la administración del *host* regido por un protocolo. Los dispositivos USB responden también a un mismo patrón estandarizado, que independiente del fabricante comprenden los mismo elementos funcionales.^[37,38] Estos elementos son: ³⁷

TRANSCIEVER: El cable USB está compuesto por solo cuatro cables: Vbus, D+, D- y GND. La información y los datos se mueven por los cables D+ y D-, con tres velocidades: 480 Mbit/s, 12 Mbit/s o 1.5 Mbit/s. Vbus y GND (tierra) tiene la tarea de llevar suministro eléctrico a los dispositivos, con un voltaje de +5V para Vbus.

SERIAL INTERFACE ENGINE – SIE: El SIE tiene la función de seriar y agrupar las transmisiones, además maneja los protocolos de comunicación, las secuencias de paquetes, el control de Código de Redundancia Cíclica (CRC) y la codificación *Non Return to Zero Invert* (NRZI).

FUNCTION INTERFACE UNIT – FIU: Este elemento administra los datos que son transmitidos y recibidos por el cable USB.

El diagrama de la Figura 7.7, ilustra el flujo de datos a partir de tres capas lógicas: capa de función, capa de dispositivos USB y capa de interfaz del bus USB.

La capa de función corresponde al nivel lógico superior que tiene como función transportar los datos que se mueven entre el software Cliente y el dispositivo USB. Hay un software Cliente en el *host*, y un software de atención a cada uno de los periféricos. En el nivel que se encuentra, el *host* se comunica con los periféricos y el software Cliente solicita a los dispositivos tareas y recibe respuestas de ellos.

La capa de dispositivos USB es una capa lógica intermedia que es manejada por el software de Sistema USB. Esta tiene la función de facilitar tareas particulares de comunicación a la capa superior. Administra la información de control y comando del dispositivo USB, así como la comunicación del periférico con la capa superior. El objetivo principal de esta capa es permitir a la capa superior concentrarse en las tareas específicas que satisfacen las necesidades del usuario y también gestiona el control interno de los periféricos.

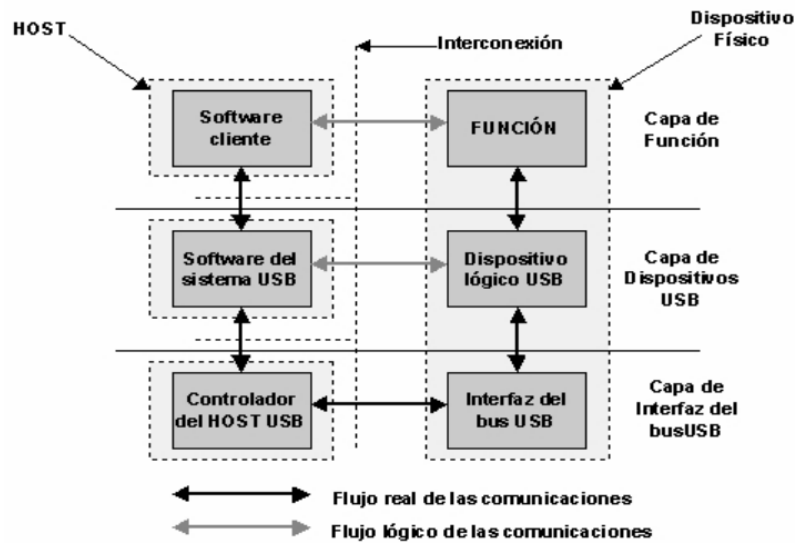


Figura 7.7. Diagrama del flujo de datos del *USB*. Imagen tomada de un trabajo de Perriago.³⁸

Finalmente, la capa de interfaz del bus *USB* o capa física, comprende el hardware que incluye: los puertos físicos, el cable, los voltajes y señales, y funcionamiento del mismo. El objetivo de la esta capa es liberar a las capas anteriores de los problemas relacionados con la modulación, los voltajes de transmisión, los saltos de fase, las frecuencias y las características netamente físicas de la transmisión.^[37,38]

7.2 Medidores de energía

En diversas investigaciones y aplicaciones que se relacionan con los láseres es de gran importancia conocer la energía o la potencia de la radiación láser. Para ello, se hacen uso de medidores de energía. En el mercado existe una gran variedad de medidores de energía, sin embargo, no hay uno que sirva para un propósito general. Es decir, que presente un rango de respuesta espectral amplio, rangos de energía amplios y que la luz externa no afecte a la medición.⁴⁰ Los medidores más comunes son los que utilizan los efectos de piroelectricidad y calorimetría. Dentro de estos medidores hay de dos tipos: los que miden la energía (J) directamente y los que miden la potencia (J/s). En este último obteniendo la energía multiplicando la potencia por el tiempo de medición. Por otro lado, se han desarrollado métodos alternativos con otros tipos de fenómenos. Esto dependiendo del rango de energía, la longitud de onda, si el láser utilizado es continuo o pulsado va a depender el tipo de medidor utilizado. A continuación se hará una breve revisión de éstos.

7.2.1 Sensores piroeléctricos

Los sensores piroeléctricos utilizados en los medidores de energía se basan en el efecto piroeléctrico. Este efecto sucede cuando hay un cambio de polarización en un material, como consecuencia de variaciones de temperatura.⁴¹ Los materiales piroeléctricos presentan polarización espontánea. Por ejemplo, en la Figura 7.8, se muestra un material piroeléctrico P_s , donde en la primera figura se muestra con una temperatura constante, donde no hay flujo de corriente por el circuito. Sin embargo, al haber un cambio en la temperatura la polarización espontánea se reduce, lo que provoca una redistribución de cargas, que causa un flujo de corriente en el circuito externo.

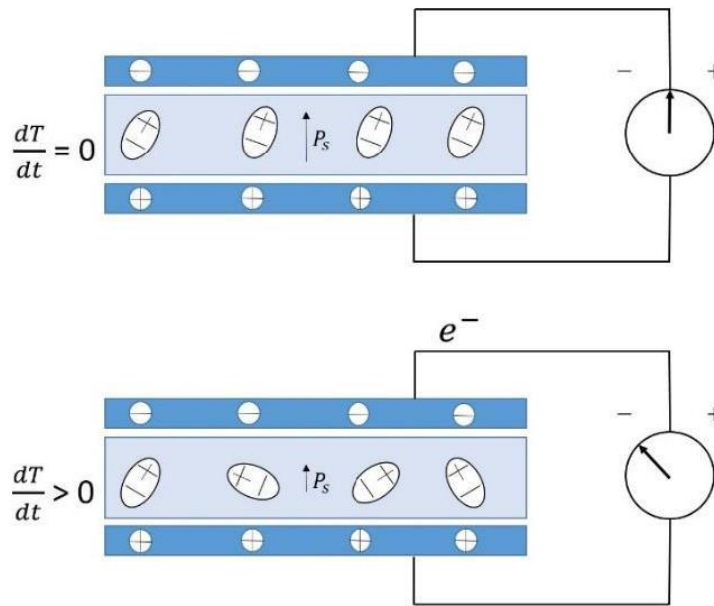


Figura 7.8. Efecto piroeléctrico. Polarización de un material. Imagen tomada de un trabajo de Elias.⁴¹

Los detectores piroeléctricos son adecuados para medir la energía de pulsos de láser de alta potencia,⁴² de manera que se utilizan en algunos de los medidores de energía más sofisticados y costosos.

7.2.2 Sensores calorimétricos

Generalmente, los detectores más sencillos están basados en efectos calorimétricos, debido a que hoy en día las técnicas fototérmicas están bien establecidas. Estas técnicas se basan en el cambio de temperatura de una muestra, asociado a la absorción de la radiación electromagnética, que depende de las propiedades térmicas y ópticas del material de la muestra.⁴⁰

El calorímetro más común consiste en un recipiente cerrado y aislado con agua, un dispositivo para agitar y un termómetro. Al hacer incidir pulsos de láser se provoca un calentamiento del material del recipiente, se agita el agua hasta llegar a un equilibrio y el aumento de temperatura se comprueba con el termómetro. Conociendo la capacidad calorífica de nuestro material y del agua se puede saber la energía liberada.⁴¹

Generalmente, para medir un amplio rango de temperaturas los sensores calorimétricos suelen utilizar termopares y termopilas.⁴⁰ Las termopilas no son otra cosa que varios termopares unidos. El termopar es un transductor formado por dos metales distintos y funcionan de la siguiente manera: cuando hay una diferencia de temperatura entre uno de los extremos “punto caliente” o de medida (donde los dos metales están en contacto) y el otro llamado “punto frío” o de referencia se produce una diferencia de potencial (del orden de los milivoltios) entre los dos finales del “punto frío” (ver Figura 7.9).

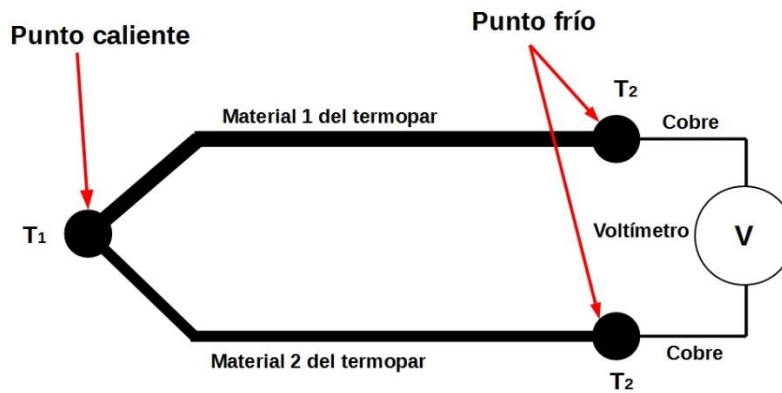


Figura 7.9. Diagrama de un termopar.

De igual forma los termopares y las termopilas son utilizados como medidores de energía de los pulsos laser, de forma independiente.⁴⁵ Esto hace que los rangos de medición de energía en que pueden utilizarse sean amplios.

7.2.3 Sensores alternativos

Se han desarrollado otras técnicas para la medición de la energía de los láseres. En 1984, Ferguson⁴⁶ describe la construcción de un medidor de potencia de láser, simple y económico. La potencia del láser fue monitoreada por un sensor calorimétrico que se basa en el uso de una resistencia comercial de platino. Usando un vidrio absorbente permitió soportar densidades de potencias pico en la región de los 100 MWcm^{-2} y potencias promedio a 10 W para una longitud de onda del láser de $1.06 \mu\text{m}$. Así como medir potencias promedio a 1mW .

Por otro lado, en 2018, Pinot et. al.^[47] describen un medidor de potencia basado en la medición de la presión de radiación con un resorte diamagnético. Realizaron un sistema de levitación aprovechando la propiedad diamagnética del carbón pirolítico. El carbón fue levitado sobre un matriz de imanes permanentes, equivalente a un sistema resorte-masa-amortiguador. Por otro lado, ese mismo año Aldama-Reyna et. al.⁴⁰ reportan la respuesta fotoacústica de un sensor piezoeléctrico que se irradió con pulsos laser de 1064 nm y diferentes energías emitidas por un láser Nd:YAG. La medida de los pulsos se realizó con un medidor de energía con un sensor piroeléctrico. Se relacionó la intensidad de la señal fotoacústica del sensor piroeléctrico con la energía de los pulsos, encontrando una dependencia lineal. La sensibilidad dependió del material protector del sensor piezoeléctrico. Para una placa de aluminio se tuvo una sensibilidad de $6.52 (0.06) \text{ mV/mJ}$. Este tipo de sensor piezoeléctrico es muy sensible a los cambios de energía de láseres pulsados de 1064 nm . No se ve afectado por condiciones de iluminación del laboratorio, ni por ruidos acústicos.

7.3 Sistema de generación de los pulsos y variación de energía

El sistema que se desarrolla en esta sección constó de dos partes: el control del láser y el control del atenuador de energía. El control del láser se realizó a través del control del generador de pulsos que estaba conectado al primero. La conexión de la computadora con el generador de pulsos se llevo acabo a través del dispositivo controlador GPIB-USB-HS. Para el correcto funcionamiento de este dispositivo controlador se instalaron los controladores NI.488.2 y NI-VISA. El programa Matlab en conjunto con los controladores antes mencionados facilitarón la programación de los aparatos. Por otro lado, el atenuador

de energía se conectó a la PC mediante el bus *USB*. Para este bus no hubo necesidad de instalar algún driver.

7.3.1 Programa de control del láser

Para generar un pulso de luz láser mediante el generador de pulsos, se necesitó mandar del generador de pulsos al láser dos señales TTL con una amplitud de 5 V y una polaridad negativa.⁴⁸ La primera señal acciona el funcionamiento de la lámpara flash y la segunda acciona el *Q-switch*. La segunda señal se activa un tiempo después de la primera por las características del funcionamiento del láser. La variación de este intervalo de tiempo define en que momento se permite el paso de la radiación a través del resonador, lo que a su vez permite el control de la energía del pulso del láser. No obstante, en este trabajo ese intervalo de tiempo se mantuvo fijo en 230 μ s, que corresponde al máximo valor de energía de emisión del láser.

El programa completo de control del generador se encuentra en el apéndice 7.6. No obstante, a continuación se da una descripción de su estructura:

- Se declaran variables generales que permiten manipular el generador de señales.
- Se inicializa y vincula el generador de señales, mediante el comando `obj = visa('vendor', 'rsrsrcname')`.
- Se utilizan comandos propios del generador⁴⁹ para definir las características de los pulsos. El comando `fprintf(obj, "comandos del generador")` se utilizó para escribir los datos relevantes del generador. Para definir las características de las señales TTL se utilizaron los comandos: `fprintf(DG, 'LAMP 2,5')` para definir la amplitud, `fprintf(DG, 'LPOOL 2,0')` para la polaridad y el comando `fprintf(DG, 'DLAY A,B,C')` se utilizó para fijar los tiempos entre los pulsos y el ancho de estos. Para la generación de un pulso del láser es necesario utilizar el modo *burst*. Este modo es un tren de pulsos que permite mandar las dos señales TTL y una señal como disparador del osciloscopio para cada pulso (ver Figura 7.10). Cada señal se mandó mediante un cable BNC propio, de forma paralela. Dado que un *burst* genera un pulso del láser, dependiendo del número de pulsos de láser requeridos serán la misma cantidad de *burst* utilizados. Para configurar el modo *burst* se utilizaron los comandos: `fprintf(DG, 'BURP A')` que define el periodo entre los *burst*, `fprintf(DG, 'BURM 1')` que habilita el modo *burst* y `fprintf(DG, 'BURC A')` que da el número de *burst* utilizados.

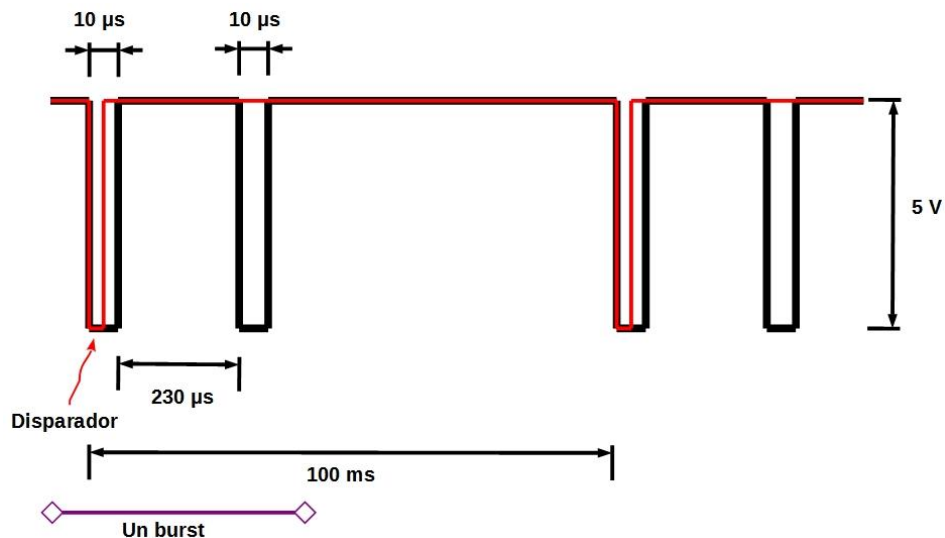


Figura 7.10. Señales TTL en modo burst.

- Se inicia el proceso del envío de los pulsos. El comando para iniciar los disparos es `fprintf(DG, '*TRG')`. Posteriormente, se pone una pausa en el flujo de los comandos para dejar que se manden todos los pulsos. Esto debido a que el software Matlab ejecuta los comandos uno tras otro, lo que puede ocasionar errores de programación con otros dispositivos conectados.
- Se cierra la vinculación con el generador de pulsos y se limpia la memoria del objeto VISA donde se guardaron las variables y los datos de los comandos antes utilizados.

7.3.2 Programa de control del atenuador de energía

El programa de control del atenuador de energía consistió en girar la placa de onda por medio del motor integrado para así poder variar la energía de los pulsos del láser.⁵⁰ El programa completo de control del atenuador se encuentra en el apéndice 7.7. No obstante, a continuación se da una descripción de su estructura.

Para la programación de control del atenuador se necesitó escribir los comandos en código ASCII. Sin embargo, Matlab permite mediante el comando `writeline(device,data)` hacer la programación más sencilla, escribiendo el código ASCII en el puerto serial especificado.

Lo primero que se realizó fue la definición de los parámetros de las variables del puerto serial del atenuador. Después, se realizó la vinculación del atenuador mediante el comando `serialport(port,bauterate)`. Seguido, se configuró mediante el comando `configureTerminator(device,"CR")` el terminador de datos, que ayudó al comando `writeline` a poner la terminación "CR" (retorno de carro) cada vez que se escriba un comando. Una vez escribió los dos comandos principales, se pasó a definir mediante el comando `writeline` los parámetros iniciales del atenuador: corriente de movimiento aplicada, corriente de ralentí, velocidad de movimiento y micro pasos del motor. El manual indica las especificaciones que vienen por defecto para el atenuador.⁴⁷ Entre cada comando se colocó una pausa de 0.5 s para dar tiempo de enviar cada comando. Se configuró el atenuador para que el motor retorne a su posición inicial cada vez que se inicie de nuevo el programa, utilizando el comando `writeline(watt_pilot,'zs 0')`. Se utilizó un ciclo `while` para verificar que el motor retornó a su posición inicial. Una vez el motor retornó a su posición inicial, se utilizó otro ciclo `while` para mover el motor a la posición deseada, el comando que se utilizó para mover el motor es `writeline(watt_pilot,'m posicion del motor')`, donde `m` es el comando para mover el motor y `posicion del motor` es una variable que indica cuantos pasos se moverá el motor. Este comando mueve el motor en pequeños pasos, que tendrán el nombre de pasos `m`. Una vez que no se necesitó mover el motor, se cerró la vinculación con el atenuador y se limpió el puerto serial.

7.4 Sistema de medición: fuente de alto voltaje, medidor de energía y osciloscopio

El desarrollo del sistema de medición se basó en el uso de dos aparatos: un medidor de energía que nos permitió conocer la energía incidente y transmitida de los pulso de láser; y un osciloscopio, que midió la caída del voltaje en la resistencia del sensor eléctrico y el monitoreo de los pulsos del láser a través de un fotodiodo. Además, este sistema usa una fuente de alto voltaje que permitió variar la tensión entre el par de placas del capacitor del sensor eléctrico. El medidor de energía se conectó mediante el bus `USB` y los otros dos se conectaron mediante el dispositivo controlador `USB – GPIB – HS`.

7.4.1 Programa de control de la fuente de alto voltaje

El programa completo de control de la fuente de alto voltaje se encuentra en el apéndice 7.8. No obstante, a continuación se da una descripción de su estructura.

Para el manejo de la fuente de alto voltaje se definieron primero los parámetros iniciales de las variables. Después se inicializó la fuente con el comando `obj = visa('vendor','rsrcname')` y mediante el comando `fprintf(PS350,'VLIM 3000;VLIM?')` se fija un límite para el voltaje. Esto como protección en caso de introducir un voltaje incorrecto en la programación. Con el comando `fprintf(PS350,'HVON')` se puso a la fuente en modo encendido. Para configurar el voltaje se usó el comando `fprintf(PS350,'VSET 300')`, en este caso se puso un voltaje de 300 V.⁵¹ Finalmente se puso la fuente en modo apagado con el comando `fprintf(PS350,'HVOF')` y se desvinculó y limpió la memoria del `obj`.

7.4.2 Programa de control del medidor de energía

El programa completo de control del medidor de energía se encuentra en el apéndice 7.9. No obstante, a continuación se da una descripción de su estructura.

Para el programa de control del medidor de energía se inició definiendo los parámetros iniciales de las variables del medidor de energía: longitud de onda, rango de energía, modo de medición, duración de medición de los pulsos de láser, límite inferior de medición y tipo de promediación de los pulsos medidos.⁵² Siguiendo, se creó la vinculación del medidor con la computadora mediante el comando `ophirApp = actxserver('OphirLMMMeasurement.CoLMMMeasurement')` para crear el objeto COM. Una vez que se creó el objeto se usó el comando `SerialNumbers = ophirApp.ScanUSB` para obtener el número de serie del medidor. Con este número de serie se abrió el dispositivo mediante el comando `h_USB = ophirApp.OpenUSBDevice(SerialNumbers{1})`. Después se establecieron los parámetros para la medición del sensor. Los comandos utilizados fueron los siguientes: `ophirApp.SetWavelength(h_USB(1),0,wavelength)`, definió la longitud de onda; `ophirApp.SetMeasurementMode(h_USB(1),0,mododemedicion)`, definió el modo de medición; `ophirApp.SetRange(h_USB(1),0,rangodeenergia)`, definió el rango de energía a medir; `ophirApp.SetPulseLength(h_USB(1),0,duracionpulsodelaser)`, definió la duración de la medición; `ophirApp.SetThreshold(h_USB(1),0,limiteinferior)`, definió el límite inferior de la energía a medir; y `ophirApp.Write(h_USB(1),['AQ ' num2str(average)])`, definió qué tipo de promediación se va a utilizar. Se utilizaron los comandos, `ophirApp.StartStream(h_USB(1),0)` y `ophirApp.GetData(h_USB(1),0)` para la transmisión de datos, para el almacenamiento de los datos se utilizó el comando `fopen([pathname '/archivo.txt'],'w')`. Se desvinculó y limpió la memoria del objeto COM.

7.4.3 Programa de control del osciloscopio

El programa completo de control del osciloscopio se encuentra en el apéndice 7.10. No obstante, a continuación se da una descripción de su estructura.

En la parte inicial del programa de control del osciloscopio se definieron los parámetros iniciales de las variables del osciloscopio⁵³ como: escala vertical y horizontal, impedancia de entrada, posición de la señal en la escala vertical, posición del *trigger*, modo de adquisición, número de promediaciones por señal, número de puntos por señal, etc. Mediante el comando `TDS = visa('ni','GPIB0::5::INSTR')` se hizo la creación del objeto VISA y con `fopen(TDS)` se conectó el dispositivo. Mediante la consulta del comando

recordlengthstring = query(TDS, 'horizontal:recordlength?') se determinó la porción de puntos de la señal a utilizar. Después, se desvinculó y se limpió la memoria del objeto VISA. Este paso es esencial, ya que se necesita saber el número de puntos a trabajar por señal, el cual no se debe cambiar una vez que el osciloscopio esté funcionando. Se repite el proceso de vincular el objeto VISA. Después se define el tamaño del buffer y el orden de los bytes mediante los comandos: *TDS.InputBufferSize = 2*recordlength+1000* y *TDS.ByteOrder = 'littleEndian'*, respectivamente. Se conectó el objeto VISA con el dispositivo. Seguido, se estableció los controles vertical, horizontal y de disparo del instrumento para adquirir y mostrar automáticamente la forma de la señal seleccionada con el comando, *fprintf(TDS, 'AUTOSet EXECute')*. Seguido, se apagó el modo de retardo de activación de la base de tiempo con el comando *fprintf(TDS, 'HORizontal:MAIn:DELay:MODE OFF')*. Después de realizar los preparativos se dispuso a configurar los canales, se utilizaron los siguientes canales del osciloscopio: el canal 1 se le asignó al monitoreo de los pulsos utilizando el fotodiodo rápido (señal óptica); el canal 2 se le asignó a la señal de disparo que provino del generador de señales; y el canal 3 se le asignó a la medición de la señal eléctrica. Mediante el comando *fprintf(TDS, 'SElect:CH X ON')* se prendieron los canales, *CH1*, *CH2* y *CH3*. Después se configuró la parte horizontal que corresponde a la medición del tiempo mediante los comandos: *fprintf(TDS, ['HORizontal:POStion ' num2str(Triggerposition)]),* *fprintf(TDS, ['HORizontal:RECOrdlength ' num2str(Numerode datapoints)])* y *fprintf(TDS, ['HORizontal:SCAle ' num2str(Escala_horizontal)])*, que corresponden a la posición del disparador, el número de puntos por señal y la escala temporal, respectivamente. Cada uno de los canales se configuró con parámetros diferentes, pero usando los mismos comandos. Se mostrará los comandos solo para el canal 1. *fprintf(TDS, 'CH1:BAWdwidth FULL')* estableció el ancho de banda; *fprintf(TDS, 'CH1:COUPling DC')* dio la configuración del atenuador de entrada; *fprintf(TDS, ['CH1:POStion'num2str(CH1_Posicion)])* dio la posición vertical de la señal; *fprintf(TDS, ['CH1:SCAle ' num2str(CH1_Escala)])* configuró la escala vertical de la señal; y *fprintf(TDS, ['CH1:TERmination 'num2str(CH1_Impedancia)])* estableció la impedancia de la conexión de entrada del canal en 50 ohms. Una vez que se definieron los parámetros para cada canal, se pasó a configurar el modo de adquisición de datos en modo promedio y el número de promediación en 100. Ésto se realizó mediante los comandos: *fprintf(TDS, ['acquire:mode ' ModoDeAquisicion])* y *fprintf(TDS, ['acquire:numavg'num2str(Numerodepromediaciones)])*, respectivamente. Después, se configuró la parte de la señal de disparo definiendo el acoplamiento, la pendiente, el canal usado y la señal de entrada utilizada mediante los siguientes comandos: *fprintf(TDS, 'TRIGger:A:EDGE:COUPling DC')*, *fprintf(TDS, 'TRIGger:A:EDGE:SLOpe FALL')*, *fprintf(TDS, 'TRIGger:A:EDGE:SOUrce CH2')* y *fprintf(TDS, 'TRIGger:A:LEVel TTL')*. Se desvinculó y limpió el objeto *USB – GPIB*.

7.5 Comandos fundamentales utilizados en los programas

A continuación se presentan los comandos principales utilizados en los programas presentados en la sección 2.2. Estos comandos corresponden tanto a los propios del software Matlab, como al de los aparatos utilizados.

7.5.1 Comandos de Matlab

A continuación se muestran los comandos utilizados en la programación de los experimentos. Si se desea profundizar se puede revisar los comandos en la página oficial de Matlab. ⁵⁴

Actxserver(progid). Crea un servidor de automatización OLE local, donde *progid* es el identificador programático (ProgID) de un servidor COM compatible con OLE. La función devuelve un identificador a la interfaz predeterminada del servidor.

Binblockread(obj). Lee datos de bloque binario (binblock) del instrumento conectado a *obj* y devuelve los valores. El formato binblock se describe en las páginas de referencia de binblockwrite.

Tobj.ByteOrder = 'A'. Configura *ByteOrder* para que sea *littleEndian* o *bigEndian*. Si *A* es *littleEndian*, el instrumento almacena el primer byte en la primera dirección de memoria. Si *A* es *bigEndian*, el instrumento almacena el último byte en la primera dirección de memoria. En el caso de este trabajo se utilizó *littleEndian* ya que es predeterminado para objetos serie, GPIB y visa.

configureTerminator(device,terminator). Define el terminador para las comunicaciones de lectura y escritura con el host remoto especificado por el cliente TCP/IP en *device*. Los valores de terminador permitidos son "LF" (predeterminado), "CR", "CR/LF" y valores enteros de 0 a 255. La sintaxis establece la propiedad Terminator de *device*.

dlmwrite(nombre de archivo, M). Escribe datos numéricos en la matriz *M* en un archivo de formato ASCII, *nombre de archivo*, utilizando el delimitador predeterminado (,) para separar los elementos de la matriz. Si el archivo, *nombre de archivo*, ya existe, *dlmwrite* sobrescribe el archivo.

fopen(filename). Abre el archivo, *filename*, para acceso de lectura binaria y devuelve un identificador de archivo entero igual o mayor que 3. MATLAB® reserva los identificadores de archivo 0, 1 y 2 para la entrada estándar, la salida estándar (la pantalla), y el error estándar, respectivamente. Si *fopen* no puede abrir el archivo, entonces *fileID* es -1.

fprintf(fileID,formatSpec,A1,...,An). Aplica *formatSpec* a todos los elementos de las matrices *A1,...,An* en orden de columna y escribe los datos en un archivo de texto. *fprintf* usa el esquema de codificación especificado en la llamada a *fopen*.

InputBufferSize. Configure *InputBufferSize* como el número total de bytes que se pueden almacenar en el búfer de entrada del software durante una operación de lectura.

interp1(x,v,xq) devuelve valores interpolados de una función 1-D en puntos de consulta específicos mediante interpolación lineal. El vector *x* contiene los puntos de muestra y *v* contiene los valores correspondientes, *v(x)*. El vector *xq* contiene las coordenadas de los puntos de consulta.

num2str(A). Convierte un arreglo numérico en un arreglo de caracteres que representa los números. El formato de salida depende de las magnitudes de los valores originales. *num2str* es útil para etiquetar y titular gráficas con valores numéricos.

obj = visa('vendor','rsrctype'). Crea el objeto VISA *obj* con un nombre de recurso proporcionado por *rsrctype* para el proveedor especificado por *proveedor*. Primero debe configurar sus recursos VISA en la herramienta del proveedor y luego crear estos objetos VISA.

query(obj,'cmd'). Escribe la cadena *cmd* en el instrumento conectado a *obj*. Los datos leídos del instrumento se devuelven a la salida. De forma predeterminada, el formato *%s\n* se usa para *cmd* y el formato *%c* se usa para los datos devueltos.

serialport (*port*, *baudrate*). Se conecta al puerto serial especificado por *port* con una tasa de baudios de *baudrate*.

str2double (*str*). Convierte el texto de una cadena *str* a un valor de doble precisión.

Writeline (*device*, *data*). Escribe *data* en texto ASCII seguidos del terminador en el puerto serie especificado. La función suspende la ejecución de MATLAB® hasta que se escriben los datos y el terminador.

7.5.2 Comandos del generador de pulsos

**TRG*. Cuando el DG645 está configurado para disparadores de disparo único, este comando inicia un disparador único. Cuando se configura para disparos únicos activados externamente, este comando prepara el DG645 para disparar en el próximo activador externo detectado.

BURC (?) *A*. Establece (consulta) el recuento de ráfagas {a *A*}. Cuando el modo de ráfagas está habilitado, el DG645 emite ciclos de retardo de conteo de ráfagas por disparador.

BURP (?) *A*. Establece (consulta) el período de ráfaga {to *A*}. El período de ráfaga establece el tiempo entre ciclos de retardo durante una ráfaga. El período de ráfaga puede oscilar entre 100 ns a 2000 ns, en 10 ns en 10 ns por paso.

DLAY (?) *A*{*B*, *C*}. Establece (consulta) el retraso para el canal *A* {a *C* relativo al canal *B*}.

LAMP (?) *A*{*B*}. Establece (consulta) la amplitud para la salida *A* {a *B*}.

LPOL (?) *A*{*B*}. Establece (consulta) la polaridad para la salida *A* {a *B*}. Si *B* es 1, la polaridad es positiva. Si *B* es 0, la polaridad es negativa.

7.5.3 Comandos del atenuador de energía

m x. Mueva el motor *x* pasos. Parámetro *x*: número entero. Puede ser positivo (el motor gira en el sentido de las agujas del reloj) y negativo (el motor gira en el sentido contrario a las agujas del reloj). Coloque "-" para la notación negativa. *x* puede estar en el rango de -2147483646..+ 2147483646.

zs x. Restablece el contador de coordenadas y luego presiona el interruptor de posición cero del hardware. Parámetro *x*: 1, el contador de posición se convertirá en 0 cada vez que presione el interruptor de posición cero; 0, el contador de posición aumenta la posición continuamente.

7.5.4 Comandos de la fuente de alto voltaje

HVOF. El comando *HVOF* apaga el alto voltaje.

HVON. El comando *HVON* enciende el alto voltaje siempre que el interruptor de alto voltaje del panel frontal no esté en la posición de apagado. Si el interruptor está en la posición apagado, el alto voltaje se deja apagado y se informa un error de ejecución. Este comando también borra automáticamente cualquier disparo por voltaje o corriente.

VLIM (?) *x*. El comando *VLIM* establece el valor del límite de voltaje en *x*, donde *x* es el valor en voltios. El signo del valor *x* DEBE coincidir con la configuración de polaridad del PS300. La consulta *VLIM?* devuelve la configuración *VLIM* actual. Al igual que con el control del panel frontal, el valor *VLIM* debe ser mayor o igual que el valor *VSET* o se devolverá un error de ejecución.

VSET (?)x. El comando *VSET* establece el valor establecido de voltaje en *x* si el control del panel frontal está habilitado. Si el control del panel posterior está habilitado, se devuelve un error. El valor *x* es un número en unidades de voltios y el signo del número DEBE coincidir con la configuración de polaridad del PS300. La consulta *VSET?* devuelve el valor *VSET* actual. Al igual que con el control del panel frontal, el valor de *VSET* debe ser menor o igual que el valor de *VLIM* o se devolverá un error de ejecución.

7.5.5 Comandos del medidor de energía

GetData. Este método sube la carga útil de datos que el dispositivo Ophir ha entregado a *OphirLMMeasurement*. Debe llamarse después de que se haya recibido el evento *DataReady*. Si se llama cuando no hay nuevos datos disponibles, las matrices estarán vacías.

OpenUSBDevice(SerialNumbers{1}). Dado *SerialNumber* que se encontró con el método *ScanUSB*, se abrirá una sesión de comunicación con un dispositivo.

SetMeasurementMode(h_USB(1),0,mododemedicion). Establece el modo de medición del sensor seleccionado en el índice. *mododemedicion* puede ser 0='Potencia', 1='Energía' o 2='Exposición'.

SetPulseLength(h_USB(1),0,duracionpulsodelaser). Establece la configuración de longitud de pulso del sensor seleccionado en el índice. *duracionpulsodelaser* puede ser 0='1.0ms', 1='2.0ms', 2='5.0ms', 3='10ms' o 4='20ms'.

SetRange(h_USB(1),0,rangodeenergia). Establece el rango de medición del sensor seleccionado en el índice. *rangodeenergia* puede ser 0='10.0J', 1='2.00J', 2='200mJ', 3='20.0mJ' o 4='2.00mJ'.

SetThreshold(h_USB(1),0,limiteinferior). Establece el umbral del sensor seleccionado en el índice. *limiteinferior* puede tomar valores de 0 a 24, correspondientes a 1% = 'Min', hasta max=25%.

SetWavelength(h_USB(1),0,wavelength). Establece la longitud de onda del sensor seleccionado en el índice. *wavelength* puede ser, 0='248', 1='355' o 2='532', 3='1064'.

StartStream(h_USB(1),0). Indique al sensor seleccionado que comience a transmitir datos a la PC. El formato de la transmisión (en búfer, turbo, inmediato) se puede configurar con el método *ConfigureStreamMode*. Cuando se llama a *StartStream*, el objeto COM vacía el búfer de las mediciones recopiladas previamente. *OphirLMMeasurement* activará el evento *DataReady* cuando llegue un nuevo paquete de datos y esté listo para que la aplicación cliente lo lea. Consulte la sección flujos de datos para obtener explicaciones detalladas sobre cómo configurar el flujo, iniciarlo y detenerlo, y analizar la respuesta.

Write(h_USB(1),comando). Escribe *comando* en el *h_USB(1)*. Comando es una cadena en código ASCII.

7.5.6 Comandos del osciloscopio

acquire:mode. Este comando establece o consulta el modo de adquisición del instrumento.

acquire:numavg. Este comando establece o consulta el número de adquisiciones de formas de onda que componen una forma de onda promediada.

AUTOSet. Este comando (sin formato de consulta) establece los controles vertical, horizontal y de disparo del instrumento para adquirir y mostrar automáticamente la forma de onda seleccionada.

CH x :BANdwidth. Este comando establece o consulta el filtro de límite de ancho de banda de paso bajo seleccionable del canal especificado. El canal se especifica mediante *x*. El valor de *x* puede oscilar entre 1 y 4.

CH x :COUpling. Este comando establece o consulta la configuración de acoplamiento del atenuador de entrada para el canal especificado. El canal se especifica mediante *x*. El valor de *x* puede variar de 1 a 4.

CH x :POSition. Este comando establece o consulta la posición vertical del canal especificado. El canal se especifica mediante *x*. El valor de *x* puede oscilar entre 1 y 4. El valor de posición se aplica a la señal antes de digitalizarla.

CH x :TERmination. Este comando establece el estado conectado/desconectado de una resistencia de 50 Ω , que se puede conectar entre la entrada acoplada del canal especificado y la tierra del instrumento. El canal se especifica mediante *x*. El valor de *x* puede oscilar entre 1 y 4.

Curve. Este comando transfiere datos de forma de onda hacia y desde el instrumento. Cada forma de onda que se transfiere tiene un preámbulo de forma de onda asociado que contiene información como formato de datos y escala. La consulta *Curve?* transfiere datos desde el instrumento. El origen de datos se especifica mediante el comando *DATA:SOURce*. Los puntos de datos primero y último se especifican mediante los comandos *DATA:STARTt* y *DATA:STOP*.

data:encdg. Este comando establece o consulta el formato de los datos de forma de onda salientes. Este comando es equivalente a configurar *WFMOutpre:ENCdg*, *WFMOutpre:BN_Fmt* y *WFMOutpre:BYT_Or*. Establecer el valor *DATA:ENGdg* hace que se actualicen los valores *WFMOutpre* correspondientes y viceversa.

data:source. Este comando establece o consulta la ubicación de los datos de forma de onda que se transfieren desde el instrumento mediante la consulta *curve?*

data:start. Este comando establece o consulta el punto de datos inicial para la transferencia de forma de onda. Este comando permite la transferencia de formas de onda parciales hacia y desde el instrumento.

data:stop. Este comando establece o consulta el último punto de datos que se transferirá al usar la consulta *CURVe?*. Este comando permite la transferencia de formas de onda parciales al controlador.

HORizontal:MAIn:DELay:MODE. Este comando establece o consulta el tiempo de retardo del disparador de la base de tiempo.

HORizontal:POSition. Este comando coloca la forma de onda horizontalmente en la pantalla o devuelve su posición horizontal en la pantalla.

HORizontal:RECOrdlength. Este comando establece o consulta la longitud del registro.

HORizontal:SCAle. Este comando establece o consulta la escala horizontal.

measurement: meas x :value?. Este comando de solo consulta devuelve el valor que se calcula para la medida especificada por *x*, que va de 1 a 8.

SElect:CH x. Este comando establece o consulta el estado mostrado de la forma de onda del canal especificado. La *x* puede ser del canal 1 al 4.

TRIGger:A:EDGE:COUpling. Este comando establece o consulta el tipo de acoplamiento para el disparador de borde.

TRIGger:A:EDGE:SLOpe. Este comando establece o consulta la pendiente del disparador de borde.

TRIGger:A:EDGE:SOURce. Este comando establece o consulta la fuente del disparador de borde.

TRIGger:A:LEVel. Este comando establece o consulta el nivel del disparador.

wavfrm?. Este comando de solo consulta devuelve *WFMOutpre?* y *CURVe?* datos para la forma de onda según lo especificado por el comando *DATA:SOURce*. Este comando es equivalente a enviar tanto *WFMOutpre?* y *CURVe?*, con la disposición adicional de que la respuesta a *WAVFrm?* está garantizado para proporcionar un preámbulo y una curva sincronizados.

WFMOutpre:XZEr0?. Este comando establece o consulta el tiempo de submuestra entre la muestra de disparo (designada por *OPT_OFF*) y la ocurrencia del disparo real en la forma de onda entrante. Este valor se utiliza para calcular *TTOFF* para la forma de onda entrante y se expresa en términos de *WFMInpre:XUNit*.

wfmoutpre:BN_FMT?. Este comando establece o consulta el formato de los datos binarios para las formas de onda entrantes.

WFMOutpre:XINcr?. Este comando establece o consulta el intervalo horizontal entre los puntos de forma de onda entrantes en unidades especificadas por *WFMInpre:XUNit*.

7.6 Programa de control del láser

```
%% Limpiar los espacios
clear all;
clc;
close all;
instreset; % Clear MATLAB workspace of any previous instrument connections
-----
%% Parámetros iniciales generador de pulsos.
numeropulsosburstQswitch = 500;
frecuencialamparalaser = 10; % Frecuencia de disparo de la lámpara del láser. Normalmente debe ser 10
Hz, que es la frecuencia a la que mejor trabaja el láser.
numeroQswitchdelay = 13; % Número de valores de retrasos del disparo del Q-switch a analizar
tipodetrigger = 5; % TSRC(?)i Trigger Source Set (query) the trigger source {to i}. The parameter i
determines the trigger source to the following table: i Trigger Source, 0 Internal, 1 External rising edges, % 2
External falling edges, 3 Single shot external, rising edges, 4 Single shot external falling edges, 5 Single shot,
6 Line
Qswitchdelay = 230e-6; %Valor del q-switch
pulsotriggerQs = 36e-8 + Qswitchdelay; % valor real del q-switch, hay un retraso. Se le suma 360 ns
-----
%% Habilitar DG645 (Delay/pulse generator)
%Se crea el objeto, se conecta el generador y se inicializa
DG = visa('ni','GPIB0::6::INSTR'); % create a VISA-GPIB objec.
fopen(DG); %connect the VISA-GPIB object to the instrument.
fprintf(DG,'*IDN?') % Query the instrument identification string.
x=fscanf(DG)
% Limpiamos de la memoria lo que haya. Definimos el trigger, la amplitud y la polaridad en este caso se
utilizan 5V y polaridad negativa.
fprintf(DG,'*CLS'); % Clear Status immediately clears the ESR and INSR registers as well as the LERR (Last
Error) error buffer.
fprintf(DG,['TSRC ' num2str(tipodetrigger)]); % TSRC(?)i Trigger Source Set (query) the trigger source {to
i}.
fprintf(DG,'LAMP 2,5'); % LAMP(?)b{,v} Level Amplitude Set (query) the amplitude for output AB {to v}.
fprintf(DG,'LPOL 2,0'); % Set (query) the polarity for output AB {to i}. If i is 1, the polarity is positive. If i is 0,
the polarity is negative.
fprintf(DG,'LAMP 3,5'); % LAMP(?)b{,v} Level Amplitude Set (query) the amplitude for output CD {to v}.
fprintf(DG,'LPOL 3,0') % Set (query) the polarity for output CD {to i}. If i is 1, the polarity is positive. If i is 0,
the polarity is negative..
```

```

fprintf(DG,'LAMP 4,5'); % LAMP(?)b{,v} Level Amplitude Set (query) the amplitude for output CD {to v}.
fprintf(DG,'LPOL 4,0') % Set (query) the polarity for output CD {to i}. If i is 1, the polarity is positive. If i is 0,
the polarity is negative..
% Las dos lineas de codigo [CD] son para la lampara y el [EF] es para el Q-switch
fprintf(DG,'DLAY 4,0,0'); % [CD] Set channel A delay to equal channel T0 plus 0 s.
fprintf(DG,'DLAY 7,6,10e-6'); % [EF] Set channel D delay to equal channel C plus 10 micros.
fprintf(DG,'DLAY 5,4,10e-6'); % [CD] Set channel B delay to equal channel A plus 10 micros.
% Las lineas siguientes son un pulso que servira para triggerear.
fprintf(DG,'DLAY 9,8,200e-9'); % Set channel A delay to equal channel T0 plus 0 s.
fprintf(DG,'BURD 0'); % Set (query) the burst delay {to t}. When burst mode is enabled the DG645 delays the
first burst pulse relative to the trigger by the burst delay.
fprintf(DG,['BURP ' num2str(1/frecuencialamparalaser)]); % Set (query) the burst period {to t}. The burst
period sets the time between delay cycles during a burst. The burst period may range from 100 ns to 2000 –
10 ns in 10 ns steps.
fprintf(DG,'BURM 1'); % Set (query) the burst mode {to i}. If i is 0, burst mode is disabled. If i is 1, burst mode
is enabled.

-----

%% SE DISPARARAN CIERTA CANTIDAD DE PULSOS
fprintf(DG,['DLAY 8,5,' num2str(pulsotriggerQs)]); % Set channel A delay to equal channel T0 plus 0 s.
fprintf(DG,['DLAY 6,5,' num2str(Qswitchdelay)]); % Set channel C delay to equal channel B plus Qswitchdelay
time.
fprintf(DG,['BURC ' num2str(meropulsosburstQswitch)]); % Set (query) the burst count {to i}. When burst
mode is enabled, the DG645 outputs burst count delay cycles per trigger.
fprintf(DG,'*TRG'); % When the DG645 is configured for single shot triggers, this command initiates a single
trigger. When it is configured for externally triggered single shots, this command arms the DG645 to trigger
on the next detected external trigger.
pause ((meropulsosburstQswitch)/frecuencialamparalaser)
pause(3)

-----

%% SE CIERRA EL OBJETO VISA
fclose(DG); %cierra VISA-GPIB objec.
delete(DG); %Borra la variable de VISA-GPIB objec.
clear DG; %limpia la memoria de VISA-GPIB objec.

```

7.7 Programa de control del atenuador Watt Pilot

```

%% Limpiar los espacios
clear all;
clc;
close all;
instrreset; % Clear MATLAB workspace of any previous instrument connections

-----

%% Serial port parameters WATT PILOT
% Baud rate 38400
baud_rate = 38400;
% Parity None
parity = 'none'; % En matlab no es necesario definir esto, ya que de entrada viene por default el valor none
% Handshaking None
% Stop bits 1
stop_bits = 1; % En matlab no es necesario definir esto, ya que de entrada viene por default el valor none

-----

%% Inicializar y vincular
watt_pilot = serialport('COM9', baud_rate);
configureTerminator(watt_pilot,'CR');
watt_pilot.Terminator;
%Usualmente esta es la configuracion comun (predeterminada)
writeline(watt_pilot,'1');% Boolean 1 or 0. Current operating mode: "Command" mode if 1, and "Step-Dir"
mode if 0;
pause(0.5);
writeline(watt_pilot,'s 55000');%Integer 1..65500. Speed value;
pause(0.5);
writeline(watt_pilot,'wn 114');%Integer 0..255. Motor motion current value;
pause(0.5);

```

```

writeline(watt_pilot,'ws 36');%Integer 0..255. Motor idle current value;
pause(0.5);
writeline(watt_pilot,'wt 114');%Integer 0..255. Motor current value in "Step-Dir" mode;
pause(0.5);
writeline(watt_pilot,'r 2');%Integer 1, 2, 4, 8 or 6. Micro stepping resolution value (full, half, quarter, eight or
sixteen);
pause(0.5);
writeline(watt_pilot,'en 1');%Boolean 1 or 0. Motor enable: motor is enabled if 1 and motor is disconnected if
0;
pause(0.5);
writeline(watt_pilot,'zr 0');%Boolean 1 or 0. Whether to report coordinate then hitting zero position switch (if
1) or do not report (if 0). If this option is on (1), controller sends string "zp: [integer_position]" on each zero
position button press;
pause(0.5);
writeline(watt_pilot,'zs 0');%Boolean 1 or 0. Whether to reset position counter on each zero position switch
press or no. If this option is disabled (0), zero position button press does nothing to step counter. If waveplate
holder is turned twice, command "o" will show us that position is more than 15600 (15600 is full waveplate
turn in full stepping mode). If this position is enabled (1), position counter will become 0 on each zero position
switch press.
pause(0.5);
writeline(watt_pilot,'zp') %regresa a su posicion inicial el motor
s=1
while s==1
    salirdelloop = 'PONGA UN NUMERO DISTINTO DE 1 CUANDO EL MOTOR ESTE EN LA POSICION
INICIAL Y SE HAYA DETENIDO: ';
    s=input(salirdelloop)
end

n=1;
valor1=0;
while n == 1
moverdis = 'introduzca un valor positivo o negativo para mover el disco y presione enter: ';
valor = input(moverdis)
writeline(watt_pilot,['m ' num2str(valor)]) %mueve el motor el numero de pasos indicado.
pause(2)
valor1 = valor1+valor
salirde = 'ponga 1 si desea continuar moviendo el valor de la energia. si desea salir tecles cualquier letra y
presione enter: ';
n=input(salirde)
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% SE CIERRA EL OBJETO SERIAL
delete (watt_pilot)
clear watt_pilot

```

7.8 Programa de control de la fuente de alto voltaje

```

%% Limpiar los espacios
clear all;
clc;
close all;
instrreset; % Clear MATLAB workspace of any previous instrument connections
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Parámetros iniciales del la fuente de poder SRS PS350.
MaxVoltaje = 3000 ; % Voltaje máximo que se puede utilizar. El máximo voltaje del equipo es de 5000 V,
con 5 mA de máxima corriente de salida.
MinVoltaje = 30; % Voltaje mínimo que se puede obtener.
Voltajedeseado = 800;
if MinVoltaje < 20 || MaxVoltaje > 5000
    error('Error, MinVoltaje < 0 V OR MaxVoltaje > 1200 V')
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

%% Habilitar High Voltage Power Supplies SRS PS350
PS350 = visa('ni','GPIB0::10::INSTR'); % create a VISA-GPIB objec.
fopen(PS350); %connect the VISA-GPIB object to the instrument.
fprintf(PS350,'*IDN?'); % Query the instrument identification string.
fprintf(PS350,'*CLS'); % CThe *CLS common command clears all status registers
-----

%% Se pone a funcionar la Fuente.
fprintf(PS350,'VLIM 3000;VLIM?') %VLIM(?) x The VLIM command sets the value of the voltage limit to x
where x is the value in Volts. The sign of value x MUST match the PS300?s polarity setting. The VLIM? query
returns the present VLIM setting. As with front panel control, the VLIM value must be greater than or equal to
the VSET value or an execution error will be returned.
fprintf(PS350,'HVON') % The HVON command turns the high voltage ON provided that the front panel high
voltage switch is not in the OFF position. If the switch is in the OFF position, the high voltage is left off and an
execution error is reported. This command also automatically clears any voltage or current trips.
fprintf(PS350,['VSET' num2str(Voltajedeseado)]); %VLIM(?) x The VLIM command sets the value of the
voltage limit to x where x is the value in Volts. The sign of value x MUST match the PS300?s polarity setting.
The VLIM? query returns the present VLIM setting. As with front panel control, the VLIM value must be greater
than or equal to the VSET value or an execution error will be returned.
pause(20)
fprintf(PS350,'HVOF') %The HVOF command turns the high voltage OFF.
-----

%% SE CIERRA EL OBJETO VISA
fclose(PS350); %cierra VISA-GPIB objec.
delete(PS350); %Borra la variable de VISA-GPIB objec.
clear PS350; %limpia la memoria de VISA-GPIB objec.

```

7.9 Programa de control del medidor de energía

```

%% Limpiar los espacios
clear all;
clc;
close all;
instrreset; % Clear MATLAB workspace of any previous instrument connections
-----

%% Datos iniciales para el medidor OPHIR
wavelength = 3; %Define la longitud de onda que se quiere analizar, 0='248', 1='355', 2='532', 3='1064'.
rangodeenergia = 2; % Define el rango de energia a medir, 0='10.0J',
1='2.00J',2='200mJ',3='20.0mJ',4='2.00mJ'.
mododemedicion = 1; % Define el modo de medicion, 0='Power', 1='Energy', 2='Exposure'.
duracionpulsodelaser = 0; % Define para que duracción de pulsos de láser 0=<'1.0ms' 1='2.0ms', 2='5.0ms',
3='10ms', 4='20ms'.
limiteinferior = 0; % Define el límite inferior para que se dispare el sensor (considerando la máxima escala,
toma valores de 0 a 24, correspondientes a 1% = 'Min', hasta %25.
average = 1; % Define qué tipo de promediación utilizar, 1='None', 2='0.5 3='1 sec', 4='3 sec', 5='10 sec',
6='30 sec'.
-----

%% Set up/initialise the object. MEDIDOR DE ENERGIA OPHIR/VEGA
% Use the actxserver command to create a link to the COM object
try
    ophirApp = actxserver('OphirLMMeasurement.CoLMMeasurement');
catch COM_error
    disp(COM_error.message);
    error('Could not establish a link to OphirLMMeasurement');
end
% Use some of the methods of the object to modify some settings, do some initialisation etc
% Request the object scans for USB devices:
SerialNumbers = ophirApp.Scan USB;
if isempty(SerialNumbers)
    warndlg('No USB devices seem to be connected. Please check and try again',...
        'Ophir Measurement COM interface: Scan USB error')
end
% Open the first USB device found:

```

```

h_USB = ophirApp.OpenUSBDevice(SerialNumbers{1});
% Query the instrument for information about the sensor connected to the USB device
[Sensor_SN Sensor_Type Sensor_Name] = ophirApp.GetSensorInfo(h_USB(1),0)
%% Se establecen los parámetros para la medición del sensor. OPHIR/VEGA
% ophirApp.GetSensorInfo(h_USB(1),0)
ophirApp.SetWavelength(h_USB(1),0,wavelength); % Define la longitud de onda que se quiere analizar,
0='248', 1='355', 2='532', 3='1064'.
ophirApp.SetMeasurementMode(h_USB(1),0,mododemedicion); % Define el modo de medición, 0='Power',
1='Energy', 2='Exposure'.
ophirApp.SetRange(h_USB(1),0,rangodeenergia); % Define el rango de energía a medir, 0='10.0J',
1='2.00J', 2='200mJ', 3='20.0mJ', 4='2.00mJ'.
ophirApp.SetPulseLength(h_USB(1),0,duracionpulsodelaser); % Define para que duración de pulsos de
láser 0=<'1.0ms', 1='2.0ms', 2='5.0ms', 3='10ms', 4='20ms'.
ophirApp.SetThreshold(h_USB(1),0,limiteinferior); % Define el límite inferior para que se dispare el sensor
(considerando la máxima escala, toma valores de 0 a 24, correspondientes a 1% = 'Min', hasta %25.
ophirApp.Write(h_USB(1),['AQ ' num2str(average)]) % Define que tipo de promediación utilizar, 1='None',
2='1 sec', 3='3 sec', 4='10 sec', 5='30 sec'.

-----
%% Medición sensor OPHIR
pause(2);
ophirApp.StartStream(h_USB(1),0);
-----

%% Toma de datos del medidor de energía ophir
pathname = uigetdir; % Abre la carpeta donde se guardaran los archivos.

[Value Timestamp Status] = ophirApp.GetData(h_USB(1),0);
ValorE = transpose(Value);
length(ValorE);

contador = zeros(1,length(ValorE));
for k = 1:1:(length(ValorE));
    contador(k) = k;
end
contadort = transpose(contador);
figure; plot(contadort,ValorE);
    titulo1 = 'Energia';
    titulo2 = 'numero de pulsos';
    datos = fopen([pathname '/EnerTransmitida_80mJ_1mm_1mm_230_2800V.txt'],'w');
    fprintf(datos,[titulo1 ' ' titulo2 '\n']);
    fprintf(datos,'%f %f \n', [ValorE contadort]);
    fclose(datos);

-----
%% SE CIERRA EL OBJETO COM
ophirApp.StopAllStreams;
ophirApp.CloseAll;
ophirApp.delete;
clear ophirApp

```

7.10 Programa de control del osciloscopio

```

% %% Limpiar los espacios
clear all;
clc;
close all;
instrreset; % Clear MATLAB workspace of any previous instrument connections
-----
%% Parámetros de entrada osciloscopio. Datos iniciales
ModoDeAquisicion = 'AVERAGE'; % Pueden ser: SAMPLE|PEAKdetect|HIREs|AVERAGE|ENVELOPE. Modo de
adquisición del osciloscopio. En string
Numerodepromediaciones = 100; % Número de promediaciones en el modo Average del osciloscopio.

```

```

Level_trigger_CH2 = 1; %Nivel del trigger para el canal 2
CH1_TakeAllWaveform = 1; % 1 se toma toda la señal, 0 se toma solo desde la posición del trigger
CH3_TakeAllWaveform = 1; % 1 se toma toda la señal, 0 se toma solo desde la posición del trigger
Triggerposition = 30; % This argument can range from 0 to ?100% and is the percentage of the waveform
that is displayed left of the center graticule. Note: The upper limit of the waveform position is slightly limited
by a value that is determined from the record length (upper limit = 100 - 1/record length).
Numerodedatapoints = 10e3; % Este es el valor de numero de puntos por dato. Puede ir de 500 a 400K
Escala_horizontal = 50e-9; %Se refiere a la escala por cuadro. Es decir, 80ns por division.
CH1_Posicion = -2; % indica la posicion del canal, si es positivo indica que esta arriba de la reticula central.
Sin embargo, si es negativo entonces esta por debajo de la reticula.
CH1_Escala = 500e-3; % Establece la escala. Por ejemplo: 500 mV por division
CH1_Impedancia = 50 ; %Establece la impedancia. Puede ser 50 ohms o 1Mohm
CH2_Posicion = -2;
CH2_Escala = 1;
CH2_Impedancia = 50;
CH3_Posicion = -1;
CH3_Escala = 200e-3;
CH3_Impedancia = 50;
cambioautomaticoescala = 1;
-----
%% Determinar en el osciloscopio DPO5054 el número de puntos a obtener, que debe ser definido
manualmente en el equipo. Luego de que esté corriendo este programa no se deben cambiar. Determinar el
número de puntos que se estudiará pues define la memoria que se necesita para trabajar. OSCILOSCOPIO
TDS = visa('ni', 'GPIB0::5::INSTR'); % create a VISA-GPIB object.
fopen(TDS); %connect the VISA-GPIB object to the instrument.
recordlengthstring = query(TDS, 'horizontal:recordlength?')%Specify the portion of the waveform that you
want to transfer.
recordlength = str2double(recordlengthstring);
fclose(TDS); %Se cierra el objeto pues se necesita establecer el InputBufferSize y para eso se necesita
cerrado el objeto v.
delete(TDS);
clear TDS;
-----
%% Habilitar Osciloscopio Tektronix TDS5052B
TDS = visa('ni', 'GPIB0::5::INSTR'); % create a VISA-GPIB objec.
TDS.InputBufferSize = 2*recordlength+1000; %Set the buffer size para obtener waveform.
TDS.ByteOrder = 'littleEndian'; % Set the |ByteOrder| to match the requirement of the instrument
fopen(TDS); %connect the VISA-GPIB object to the instrument.
fprintf(TDS, 'AUTOSet EXECute'); %This command (no query format) sets the vertical, horizontal, and trigger
controls of the instrument to automatically acquire and display the selected waveform.
fprintf(TDS, 'HEADER OFF'); % Turn headers off, this makes parsing easier
fprintf(TDS, 'HORizontal:MAIn:DElay:MODE OFF'); % This command sets or queries the time base trigger
delay mode. This command is equivalent to choosing Delay Mode On from the Horiz/ Acq menu.
fprintf(TDS, 'SElect:CH1 ON');
fprintf(TDS, 'SElect:CH2 ON');
fprintf(TDS, 'SElect:CH3 ON');
-----
%% Se colocan los parametros del Osciloscopio para la parte HORIZONTAL
fprintf(TDS, ['HORizontal:POSition ' num2str(Triggerposition)]);
fprintf(TDS, ['HORizontal:RECOrdlength ' num2str(Numerodedatapoints)]); %This command sets the
horizontal record length to the number of data points in each frame.
fprintf(TDS, ['HORizontal:SCAle ' num2str(Escala_horizontal)]); %This command sets or queries the time per
division for the time base
-----
%% Se colocan los parametros del Osciloscopio para la parte VERTICAL
% Se pondran los parametro correspondientes al canal 1 (fotodiodo)
fprintf(TDS, 'CH1:BANdwidth FULL');
fprintf(TDS, 'CH1:COUPling DC');
fprintf(TDS, ['CH1:POSition ' num2str(CH1_Posicion)]); %This command sets or queries the vertical position
of the specified channel.
fprintf(TDS, ['CH1:SCAle ' num2str(CH1_Escala)]); %This command sets or queries the vertical scale of the
specified channel.

```



```

fprintf(TDS,['CH1:TERmination ' num2str(CH1_Impedancia)]); %This command sets the
connected/disconnected status of a 50 Ω resistor, which may be connected between the specified channel's
coupled input and instrument ground.
% Se pondran los parametros correspondientes al canal 2 (trigger)
fprintf(TDS,'CH2:BANdwidth FULL');
fprintf(TDS,'CH2:COUPling DC');
fprintf(TDS,['CH2:POSition ' num2str(CH2_Posicion)]);
fprintf(TDS,['CH2:SCAle ' num2str(CH2_Escala)]);
fprintf(TDS,['CH2:TERmination ' num2str(CH2_Impedancia)]);
% Se pondran los parametros correspondientes al canal 3 (fausto señal)
fprintf(TDS,'CH3:BANdwidth FULL');
fprintf(TDS,'CH3:COUPling DC');
fprintf(TDS,['CH3:POSition ' num2str(CH3_Posicion)]);
fprintf(TDS,['CH3:SCAle ' num2str(CH3_Escala)]);
fprintf(TDS,['CH3:TERmination ' num2str(CH3_Impedancia)]);

-----

%% Modo de adquisicion
fprintf(TDS,['acquire:mode ' ModoDeAquisicion]); %This command sets or queries whether the specified
measurement slot is computed and displayed.
if strcmp(ModoDeAquisicion,'AVERAGE')
fprintf(TDS,['acquire:numavg ' num2str(Numerodepromediaciones)]); %This command sets or queries the
number of waveform acquisitions that make up an averaged waveform.
end
fprintf(TDS, 'acquire:state OFF'); %This command starts or stops acquisitions. When state is set to ON or
RUN, a new acquisition will be started. If the last acquisition was a single acquisition sequence, a new single
sequence acquisition will be started. If the last acquisition was continuous, a new continuous acquisition will
be started.

-----

%% Preparacion del trigger para el Channel 2
fprintf(TDS, 'TRIGger:A:EDGE:COUPling DC'); %This command sets or queries the type of coupling for the
A edge trigger. This command is equivalent to selecting A Event (Main) Trigger Setup from the Trig menu,
selecting Edge Trigger, and choosing from the Coupling drop--down list.
fprintf(TDS, 'TRIGger:A:EDGE:SLOpe FALL'); %This command sets or queries the slope for the A edge
trigger. This command is equivalent to selecting A Event (Main) Trigger Setup from the Trig menu and then
choosing the desired Slope.
fprintf(TDS, 'TRIGger:A:EDGE:SOUrce CH2'); %This command sets or queries the source for the A edge
trigger. This command is equivalent to selecting A Event (Main) Trigger Setup from the Trig menu and then
choosing from the Source drop--down list.
fprintf(TDS, 'TRIGger:A:LEVel TTL');

```

7.11 Programa calibración energía

```

%% Limpiar los espacios
clear all;
clc;
close all;
instrreset; % Elimina el espacio de trabajo de MATLAB de cualquier conexión de instrumento anterior

CORRIDA = 5; %MODIFICAR CADA CORRIDA

-----

%% Datos iniciales para el medidor OPHIR
%Se definen las variables del medidor de energía
wavelength = 3; %Define la longitud de onda que se quiere analizar, 0='248', 1='355', 2='532', 3='1064'.
rangodeenergia = 1; % Define el rango de energia a medir, 0='10.0J',
1='2.00J',2='200mJ',3='20.0mJ',4='2.00mJ'.
mododemedicion = 1; % Define el modo de medicion, 0='Power', 1='Energy', 2='Exposure'.
duracionpulsodelaser = 0; % Define para que duración de pulsos de láser 0=<'1.0ms' 1='2.0ms', 2='5.0ms',
3='10ms', 4='20ms'.
limiteinferior = 0; % Define el límite inferior para que se dispare el sensor (considerando la máxima escala,
toma valores de 0 a 24, correspondientes a 1% = 'Min', hasta %25.
average = 1; % Define que tipo de promediación utilizar, 1='None', 2='0.5 3='1 sec', 4='3 sec', 5='10 sec',
6='30 sec'.

```

```

frecuencialaser = 10; % Frecuencia de disparo del láser.
-----
%% Parámetros iniciales Digital Delay Generator. Se definen las variables del generados de pulsos.
frecuencialamparalaser = frecuencialaser; % Frecuencia de disparo de la lámpara del láser. Normalmente
debe ser 10 Hz, que es la frecuencia a la que mejor trabaja el láser.
numeropulsosburstprueba = 175;
Qswitchdelaymin = 230e-6; % tiempo mínimo de retraso del disparo del Q-switch del láser con relación a la
lámpara. El valor mínimo es 230 microsegundos.
Qswitchdelaymax = 440e-6; % tiempo máximo de retraso del disparo del Q-switch del láser con relación a la
lámpara. El valor máximo no debe ser mayor a 500 microsegundos (Comprobar este valor).
numeroQswitchdelay = 3; % Número de intervalos para los valores de retrasos del disparo del Q-switch
tipodetrigger = 5; % TSRC(?) Trigger Source Set (query) the trigger source {to i}.
if Qswitchdelaymin < 200e-6 || Qswitchdelaymax > 500e-6
    error('Error, Qsiwtchdelaymin < 230e-6 OR Qsiwtchdelaymax > 500e-6')
end
-----
%% Parametros iniciales WATT PILOT. Serial port parameters WATT PILOT

% Baud rate 38400
baud_rate = 38400;
% Parity None
parity = 'none'; % En matlab no es necesario definir esto, ya que de entrada viene por default el valor none
% Handshaking None % Stop bits 1
stop_bits = 1; % En matlab no es necesario definir esto, ya que de entrada viene por default el valor none
mpasos = 44; % Son los pasos totales que se harán
-----
%% Configuracion/inicializacion del Medidor Ophir
try
    ophirApp = actxserver('OphirLMMeasurement.CoLMMeasurement');
catch COM_error
    disp(COM_error.message);
    error('Could not establisht a link to OphirLMMeasurement');
end
% Request the object scans for USB devices:
SerialNumbers = ophirApp.ScanUSB;
if(isempty(SerialNumbers))
    warndlg('No USB devices seem to be connected. Please check and try again',...
        'Ophir Measurement COM interface: ScanUSB error')
end
% Open the first USB device found:
h_USB = ophirApp.OpenUSBDevice(SerialNumbers{1});
% Query the instrument for information about the sensor connected to the USB device
[Sensor_SN Sensor_Type Sensor_Name] = ophirApp.GetSensorInfo(h_USB(1),0)
%% Se establecen los parámetros para la medición.
ophirApp.SetWavelength(h_USB(1),0,wavelength); % Define la longitud de onda que se quiere analizar,
0='248', 1='355', 2='532', 3='1064'.
ophirApp.SetMeasurementMode(h_USB(1),0,mododemedicion); % Define el mundo de medicion,
0='Power', 1='Energy', 2='Exposure'.
ophirApp.SetRange(h_USB(1),0,rangodeenergia); % Define el rango de energia a medir, 0='10.0J',
1='2.00J', 2='200mJ', 3='20.0mJ', 4='2.00mJ'.
ophirApp.SetPulseLength(h_USB(1),0,duracionpulsodelaser); % Define para que duración de pulsos de
láser 0=<'1.0ms', 1='2.0ms', 2='5.0ms', 3='10ms', 4='20ms'.
ophirApp.SetThreshold(h_USB(1),0,limiteinferior); % Define el límite inferior para que se dispare el sensor
(considerando la máxima escala, toma valores de 0 a 24, corresponsdientes a 1% = 'Min', hasta %25.
ophirApp.Write(h_USB(1),['AQ ' num2str(average)]) % Define que tipo de promediación utilizar, 1='None',
2='1 sec', 3='3 sec', 4='10 sec', 5='30 sec'.
-----
%% Medición
% Can optionally adjust some settings here to configure the stream mode.. Instruct the sensor to start
streaming measurements on the first channel:

```

```

pause(2);
ophirApp.StartStream(h_USB(1),0);
-----
%% Habilitar DG535 (Delay/pulse generator)
%Se crea el objeto, se conecta el generador y se inicializa
DG = visa('ni','GPIB0::6::INSTR'); % create a VISA-GPIB objec.
fopen(DG); %connect the VISA-GPIB object to the instrument.
fprintf(DG,'*IDN?') % Query the instrument identification string.
x=fscanf(DG)
% Limpiamos de la memoria lo que haya. Definimos el trigger, la amplitud y la polaridad en este caso se
utilizan 5V y polaridad negativa.
fprintf(DG,'*CLS'); % Clear Status immediately clears the
fprintf(DG,['TSRC ' num2str(tipodetrigger)]); % TSRC(?)i Trigger Source Set (query) the trigger source {to
i}.
fprintf(DG,'LAMP 2,5'); % LAMP(?)b{,v} Level Amplitude Set (query) the amplitude for output AB {to v}.
fprintf(DG,'LPOL 2,0'); % Set (query) the polarity for output AB {to i}. If i is 1, the polarity is positive. If i is 0,
the polarity is negative..
fprintf(DG,'LAMP 3,5'); % LAMP(?)b{,v} Level Amplitude Set (query) the amplitude for output CD {to v}.
fprintf(DG,'LPOL 3,0'); % Set (query) the polarity for output CD {to i}. If i is 1, the polarity is positive. If i is 0,
the polarity is negative..
fprintf(DG,'LAMP 4,5'); % LAMP(?)b{,v} Level Amplitude Set (query) the amplitude for output CD {to v}.
fprintf(DG,'LPOL 4,0'); % Set (query) the polarity for output CD {to i}. If i is 1, the polarity is positive. If i is 0,
the polarity is negative..
% Las dos lineas de codigo [CD] son para la lampara y el [EF] es para el Q-switch
fprintf(DG,'DLAY 4,0,0'); % [CD] Set channel A delay to equal channel T0 plus 0 s.
fprintf(DG,'DLAY 7,6,10e-6'); % [EF] Set channel D delay to equal channel C plus 10 micros.
fprintf(DG,'DLAY 5,4,10e-6'); % [CD] Set channel B delay to equal channel A plus 10 micros.
% Las lineas siguientes son un pulso que servira para triggerear.
fprintf(DG,'DLAY 9,8,200e-9'); % Set channel A delay to equal channel T0 plus 0 s.
fprintf(DG,'BURD 0'); % Set (query) the burst delay {to t}. When burst mode is enabled the DG645 delays the
first burst pulse relative to the trigger by the burst delay.
fprintf(DG,['BURP ' num2str(1/frecuencialamparalaser)]); % Set (query) the burst period {to t}. The burst
period sets the time between delay cycles during a burst. The burst period may range from 100 ns to 2000 –
10 ns in 10 ns steps.
fprintf(DG,'BURM 1'); % Set (query) the burst mode {to i}. If i is 0, burst mode is disabled. If i is 1, burst mode
is enabled.
-----
%% Inicializar y vincular
watt_pilot = serialport('COM9', baud_rate);
configureTerminator(watt_pilot,'CR');
watt_pilot.Terminator;
%Usualmente esta es la configuracion comun (predeterminada)
writeline(watt_pilot,'1');% Boolean 1 or 0. Current operating mode: "Command" mode if 1, and "Step-Dir"
mode if 0;
pause(0.5);
writeline(watt_pilot,'s 55000');%Integer 1..65500. Speed value;
pause(0.5);
writeline(watt_pilot,'wn 114');%Integer 0..255. Motor motion current value;
pause(0.5);
writeline(watt_pilot,'ws 36');%Integer 0..255. Motor idle current value;
pause(0.5);
writeline(watt_pilot,'wt 114');%Integer 0..255. Motor current value in "Step-Dir" mode;
pause(0.5);
writeline(watt_pilot,'r 2');%Integer 1, 2, 4, 8 or 6. Micro stepping resolution value (full, half, quarter, eight or
sixteen);
pause(0.5);
writeline(watt_pilot,'en 1');%Boolean 1 or 0. Motor enable: motor is enabled if 1 and motor is disconnected if
0;
pause(0.5);
writeline(watt_pilot,'zr 0');%Boolean 1 or 0.
pause(0.5);
writeline(watt_pilot,'zs 0');%Boolean 1 or 0.
pause(0.5);

```

```

writeline(watt_pilot,'zp') %regresamos a la posición inicial

n=1
while n==1
    salirdelloop = 'PONGA UN NUMERO DISTINTO DE 1 CUANDO EL MOTOR ESTE EN LA POSICION
INICIAL Y SE HAYA DETENIDO: ';
    n=input(salirdelloop)
end

writeline(watt_pilot,'m -550')
pause(3)

pathname = uigetdir;

pasosGM = -550;
PROMEDIOener = double(zeros(mpasos,1));
PASOSgm = double(zeros(mpasos,1));
DESVIACIONs = double(zeros(mpasos,1));
cambio_escala_energia = double(zeros(mpasos,1));
Lente = 7.5 ; %Cambiar dependiendo del lente

for i = 1:mpasos
    %Se modifica la posicion del Motor del atenuador en cada iteracion del for
    pasosmotor = 50;
    writeline(watt_pilot,['m ' num2str(pasosmotor)]); %aquí va pasosmotor porque cuando iniciamos empieza
en N numero
    y cuando pasa una vuelta del for es N+pasosmotor entonces, ese es el nuevo N por así decirlo. Ejemplo:
inicia en -500,
    entonces al pasar por esta línea tendríamos -450 y en la siguiente -400 y así hasta terminal el ciclo for.
    pause(1)
    pasosGM = pasosmotor + pasosGM %hace el conteo de los pasos del motor del atenuador de energía
    PASOSgm (i,1) = pasosGM;
    %El funcionamiento del laser inicia.
    Qswitchdelay = 230e-6;
    pulsotriggers = 36e-8 + Qswitchdelay;
    fprintf(DG,['DLAY 8,5,' num2str(pulsotriggers)]); % Set channel A delay to equal channel T0 plus 0 s.
    fprintf(DG,['DLAY 6,5,' num2str(Qswitchdelay)]); % Set channel C delay to equal channel B plus
Qswitchdelay time.
    fprintf(DG,['BURC ' num2str( Numeropulsosburstprueba)]); % Set (query) the burst count {to i}.
    fprintf(DG,['*TRG']); % When the DG645 is configured for single shot triggers, this command initiates a single
trigger.

    pause(Numeropulsosburstprueba/frecuencialaser+1) % se tiene que esperar un tiempo mayor equivalente
al tiempo de
    los pulsos para que el medidor tome los datos

    %% Analisis de los pulsos y
    [Value Timestamp Status]= ophirApp.GetData(h_USB(1),0); %Los datos son colocados en un matriz
    ValorE = transpose(Value); % aislamos el valor de cada pulso en una lista
    CONTADOR = zeros(length(ValorE),1); %Se crea una lista con el numero total de los datos
    PromedioEner = mean2(ValorE(150:length(ValorE))); %sacamos el promedio de la energía a partir del
pulso 150.
    DesviacionS = std(ValorE(150:length(ValorE))); %Sacamos la desviación estandar

    PROMEDIOener (i,1) = PromedioEner; %Se guarda en cada iteración el promedio en una lista
    DESVIACIONs (i,1) =DesviacionS; %Se guarda en cada iteración la desviación estandar

for k = 1:1:(length (ValorE))
    CONTADOR(k,1) = k;
end
    ANALISIS_ENER_PULSOS = [CONTADOR,ValorE];
    figure
    subplot(2,1,1) % add first plot in 2 x 1 grid

```

```

scatter(CONTADOR, ValorE)
title('Energia de pulso')

subplot(2,1,2) % add second plot in 2 x 1 grid
scatter(PASOSgm,PROMEDIOener) % plot using + markers
title('Energia por pasos Atenuador')

pause(5);
end

ANALISIS_ENER_PASOS = [PASOSgm,PROMEDIOener,DESVIACIONs];
dlmwrite([pathname '/Analisis_EnerPasos_corr_' num2str(CORRIDA) '_lente_' num2str(Lente)
'.txt'],ANALISIS_ENER_PASOS,'delimiter','\t','precision',[%1.4f %1.4f %1.4f]);
-----

%% interpolación.
ENERGY = dlmread(['Analisis_EnerPasos_corr_' num2str(CORRIDA) '_lente_' num2str(Lente) '.txt']);

x = ENERGY(:,1);
y = ENERGY(:,2);
Y = y*1000;
dx = -550:10:1650;
dy = interp1(x,Y,dx,'spline');
tdx = transpose(dx);
tdy = transpose(dy);
ENERGY_interpolada = [tdx,tdy];
dlmwrite([pathname '/Analisis_EnerPasosInterpo_corrada_' num2str(CORRIDA) '_lente_' num2str(Lente)
'.txt'],ENERGY_interpolada,'delimiter','\t','precision',[%1.5f %1.5f]);
-----

%% Cierro todo
ophirApp.StopAllStreams;
ophirApp.CloseAll;
ophirApp.delete;
clear ophirApp

fclose(DG); %cierra VISA-GPIB objec.
delete(DG); %Borra la variable de VISA-GPIB objec.
clear DG; %limpia la memoria de VISA-GPIB objec.

delete (watt_pilot)
clear watt_pilot

```

7.12 Programa del experimento

```

% %% Limpiar los espacios
clear all;
clc;
close all;
instrreset; % Clear MATLAB workspace of any previous instrument connections
-----

%% Datos iniciales para el medidor OPHIR
wavelength = 3; %Define la longitud de onda que se quiere analizar, 0='248', 1='355', 2='532', 3='1064'.
rangodeenergia = 2; % Define el rango de energia a medir, 0='10.0J',
1='2.00J',2='200mJ',3='20.0mJ',4='2.00mJ'.
mododemedicion = 1; % Define el modo de medicion, 0='Power', 1='Energy', 2='Exposure'.
duracionpulsodelaser = 0; % Define para que duración de pulsos de láser 0=<'1.0ms' 1='2.0ms', 2='5.0ms',
3='10ms', 4='20ms'.
limiteinferior = 0; % Define el límite inferior para que se dispare el sensor (considerando la máxima escala,
toma valores de 0 a 24, correspondientes a 1% = 'Min', hasta %25.
average = 1; % Define que tipo de promediación utilizar, 1='None', 2='0.5 3='1 sec', 4='3 sec', 5='10 sec',
6='30 sec'.
frecuencialaser = 10; % Frecuencia de disparo del láser.
%% Parámetros iniciales generador de pulsos.

```

```

numeropulsosburstQswitch = 600;
frecuencialamparalaser = frecuencialaser; % Frecuencia de disparo de la lámpara del láser. Normalmente
debe ser 10 Hz, que es la frecuencia a la que mejor trabaja el láser.
Qswitchdelaymin = 230e-6; % tiempo mínimo de retraso del disparo del Q-switch del láser con relación a la
lámpara. El valor mínimo es 230 microsegundos (230e-6).
Qswitchdelaymax = 350e-6; % tiempo máximo de retraso del disparo del Q-switch del láser con relación a la
lámpara. El valor máximo no debe ser mayor a 500 microsegundos (Comprobar este valor).
tipodetrigger = 5; % TSRC(?) Trigger Source Set (query) the trigger source {to i}.
if Qswitchdelaymin < 230e-6 || Qswitchdelaymax > 500e-6
    error('Error, Qswitchdelaymin < 230e-6 OR Qswitchdelaymax > 500e-6')
end

-----

%% Parámetros de entrada osciloscopio % Datos iniciales
ModoDeAquisicion = 'AVERAGE'; % Pueden ser: SAMPLE|PEAKdetect|HIREs|AVERage|ENVELOpe.
Numerodepromediaciones = 100 ; % Número de promediaciones en el modo Average del osciloscopio.
Level_trigger_CH2 = 1; % Nivel del trigger para el canal 2
CH1_TakeAllWaveform = 1; % 1 se toma toda la señal, 0 se toma solo desde la posición del trigger
CH3_TakeAllWaveform = 1; % 1 se toma toda la señal, 0 se toma solo desde la posición del trigger
Triggerposition = 30; % This argument can range from 0 to 100% and is the percentage of the waveform
that is displayed left of the center graticule.
Numerodedatapoints = 10e3; % Este es el valor de número de puntos por dato. Puede ir de 500 a 400K
Escala_horizontal = 50e-9; % Se refiere a la escala por cuadro. Es decir, 80ns por division.
CH1_Posicion = -2; % indica la posición del canal, si es positivo indica que esta arriba de la reticula central.
Sin embargo, si es negativo entonces esta por debajo de la reticula.
CH1_Escala = 500e-3; % Establece la escala. Por ejemplo: 500 mV por division
CH1_Impedancia = 50 ; % Establece la impedancia. Puede ser 50 ohms o 1Mohm
CH2_Posicion = -2;
CH2_Escala = 1;
CH2_Impedancia = 50;
CH3_Posicion = -1;
CH3_Escala = 200e-3;
CH3_Impedancia = 50;
cambioautomaticoescala = 1;

-----

%% Parámetros iniciales de la fuente de poder SRS PS350.
MaxVoltaje = 4200 ; % Voltaje máximo que se puede utilizar. El máximo voltaje del equipo es de 5000 V, con
5 mA de máxima corriente de salida.
MinVoltaje = 30; % Voltaje mínimo que se puede obtener.
Numerodevoltajes = 6;
if MinVoltaje < 20 || MaxVoltaje > 5000
    error('Error, MinVoltaje < 0 V OR MaxVoltaje > 1200 V')
end

-----

%% Serial port parameters WATT PILOT
% Baud rate 38400
baud_rate = 38400;
% Parity None
parity = 'none'; % En matlab no es necesario definir esto, ya que de entrada viene por default el valor none
% Handshaking None % Stop bits 1
stop_bits = 1; % En matlab no es necesario definir esto, ya que de entrada viene por default el valor none

-----

%% Inicializar y vincular
watt_pilot = serialport('COM9', baud_rate);
configureTerminator(watt_pilot, 'CR');
watt_pilot.Terminator;
% Usualmente esta es la configuración común (predeterminada)
writeline(watt_pilot, '1'); % Boolean 1 or 0. Current operating mode: "Command" mode if 1, and "Step-Dir" mode
if 0;
pause(0.5);
writeline(watt_pilot, 's 55000'); % Integer 1..65500. Speed value;
pause(0.5);
writeline(watt_pilot, 'wn 114'); % Integer 0..255. Motor motion current value;

```

```

pause(0.5);
writeline(watt_pilot,'ws 36');%Integer 0..255. Motor idle current value;
pause(0.5);
writeline(watt_pilot,'wt 114');%Integer 0..255. Motor current value in "Step-Dir" mode;
pause(0.5);
writeline(watt_pilot,'r 2');%Integer 1, 2, 4, 8 or 6. Micro stepping resolution value (full, half, quarter, eight or
sixteen);
pause(0.5);
writeline(watt_pilot,'en 1');%Boolean 1 or 0. Motor enable: motor is enabled if 1 and motor is disconnected if
0;
pause(0.5);
writeline(watt_pilot,'zr 0');%Boolean 1 or 0. Whether to report coordinate then hitting zero position switch (if
1) or do not report (if 0). If this option is on (1), controller sends string "zp:[integer_position]" on each zero
position button press;
pause(0.5);
writeline(watt_pilot,'zs 0');%Boolean 1 or 0.
pause(0.5);
writeline(watt_pilot,'zp') %regresa el motor a la posicion inicial.
-----
%% Set up/initialise the object. MEDIDOR DE ENERGIA OPHIR/VEGA
try
    ophirApp = actxserver('OphirLMMeasurement.CoLMMeasurement');
catch COM_error
    disp(COM_error.message);
    error('Could not establish a link to OphirLMMeasurement');
end
SerialNumbers = ophirApp.Scan USB;
if isempty(SerialNumbers)
    warndlg('No USB devices seem to be connected. Please check and try again',...
        'Ophir Measurement COM interface: Scan USB error')
end
% Open the first USB device found:
h_USB = ophirApp.Open USBDevice(SerialNumbers{1});
% Query the instrument for information about the sensor connected to the USB device
[Sensor_SN Sensor_Type Sensor_Name] = ophirApp.GetSensorInfo(h_USB(1),0)
-----
%% Se establecen los parámetros para la medición del sensor. OPHIR/VEGA
ophirApp.SetWavelength(h_USB(1),0,wavelength); % Define la longitud de onda que se quiere analizar,
0='248', 1='355', 2='532', 3='1064'.
ophirApp.SetMeasurementMode(h_USB(1),0,mododemedicion); % Define el modo de medicion,
0='Power', 1='Energy', 2='Exposure'.
ophirApp.SetRange(h_USB(1),0,rangodeenergia); % Define el rango de energia a medir, 0='10.0J',
1='2.00J', 2='200mJ', 3='20.0mJ', 4='2.00mJ'.
ophirApp.SetPulseLength(h_USB(1),0,duracionpulsodelaser); % Define para que duración de pulsos de
láser 0=<'1.0ms', 1='2.0ms', 2='5.0ms', 3='10ms', 4='20ms'.
ophirApp.SetThreshold(h_USB(1),0,limiteinferior); % Define el límite inferior para que se dispare el sensor
(considerando la máxima escala, toma valores de 0 a 24, correspondientes a 1% = 'Min', hasta %25.
ophirApp.Write(h_USB(1),['AQ ' num2str(average)]) % Define que tipo de promediación utilizar, 1='None',
2='1 sec', 3='3 sec', 4='10 sec', 5='30 sec'.
-----
%% Medición sensor OPHIR
pause(2);
ophirApp.StartStream(h_USB(1),0);
-----
%% Habilitar DG535 (Delay/pulse generator)
DG = visa('ni','GPIB0::6::INSTR'); % create a VISA-GPIB objec.
fopen(DG); %connect the VISA-GPIB object to the instrument.
fprintf(DG, '*IDN?'); % Query the instrument identification string.
x=fscanf(DG)

```

```

% Limpiamos de la memoria lo que haya. Definimos el trigger, la amplitud y la polaridad en este caso se
utilizan 5V y polaridad negativa.
fprintf(DG,'*CLS'); % Clear Status immediately clears the ESR and INSR registers as well as the LERR (Last
Error) error buffer.
fprintf(DG,['TSRC ' num2str(tipodetrigger)]); % TSRC(?)i Trigger Source Set (query) the trigger source {to
i}.
fprintf(DG,'LAMP 2,5'); % LAMP(?)b{,v} Level Amplitude Set (query) the amplitude for output AB {to v}.
fprintf(DG,'LPOL 2,0'); % Set (query) the polarity for output AB {to i}. If i is 1, the polarity is positive. If i is 0,
the polarity is negative..
fprintf(DG,'LAMP 3,5'); % LAMP(?)b{,v} Level Amplitude Set (query) the amplitude for output CD {to v}.
fprintf(DG,'LPOL 3,0') % Set (query) the polarity for output CD {to i}. If i is 1, the polarity is positive. If i is 0,
the polarity is negative..
fprintf(DG,'LAMP 4,5'); % LAMP(?)b{,v} Level Amplitude Set (query) the amplitude for output CD {to v}.
fprintf(DG,'LPOL 4,0') % Set (query) the polarity for output CD {to i}. If i is 1, the polarity is positive. If i is 0,
the polarity is negative..
% Las dos lineas de codigo [CD] son para la lampara y el [EF] es para el Q-switch
fprintf(DG,'DLAY 4,0,0'); % [CD] Set channel A delay to equal channel T0 plus 0 s.
fprintf(DG,'DLAY 7,6,10e-6'); % [EF] Set channel D delay to equal channel C plus 10 micros.
fprintf(DG,'DLAY 5,4,10e-6'); % [CD] Set channel B delay to equal channel A plus 10 micros.
% Las lineas siguientes son un pulso que servira para triggerear.
fprintf(DG,'DLAY 9,8,200e-9'); % Set channel A delay to equal channel T0 plus 0 s.
fprintf(DG,'BURD 0'); % Set (query) the burst delay {to t}. When burst mode is enabled the DG645 delays the
first burst pulse relative to the trigger by the burst delay.
fprintf(DG,['BURP ' num2str(1/frecuencialamparalaser)]); % Set (query) the burst period {to t}. The burst
period sets the time between delay cycles during a burst. The burst period may range from 100 ns to 2000 –
10 ns in 10 ns steps.
fprintf(DG,'BURM 1'); % Set (query) the burst mode {to i}. If i is 0, burst mode is disabled. If i is 1, burst mode
is enabled.

-----

%% Habilitar High Voltage Power Supplies SRS PS350
PS350 = visa('ni','GP1B1::14::INSTR'); % create a VISA-GPIB objec.
fopen(PS350); %connect the VISA-GPIB object to the instrument.
fprintf(PS350,'*IDN?'); % Query the instrument identification string.
fprintf(PS350,'*CLS'); % CThe *CLS common command clears all status registers

-----

%% Determinar en el osciloscopio TDS5054B el número de puntos a obtener, que debe ser definido
manualmente en el equipo. Luego de que esté corriendo este programa no se deben cambiar. Determinar el
número de puntos que se estudiará pues define la memoria que se necesita para trabajar.
% Luego de que esté corriendo este programa no se deben cambiar. Determinar el número de puntos que
se estudiará pues define la memoria que se necesita para trabajar. %OSCILOSCOPIO
TDS = visa('ni','GP1B0::5::INSTR'); % create a VISA-GPIB object.
fopen(TDS); %connect the VISA-GPIB object to the instrument.
recordlengthstring = query(TDS, 'horizontal:recordlength?');%Specify the portion of the waveform that you
want to transfer.
recordlength = str2double(recordlengthstring);
fclose(TDS); %Se cierra el objeto pues se necesita establecer el InputBufferSize y para eso se necesita
cerrado el objeto v.
delete(TDS);
clear TDS;

-----

%% Habilitar Osciloscopio Tektronix TDS5052B
TDS = visa('ni','GP1B0::5::INSTR'); % create a VISA-GPIB objec.
TDS.InputBufferSize = 2*recordlength+1000; %Set the buffer size para obtener waveform.
TDS.ByteOrder = 'littleEndian'; % Set the |ByteOrder| to match the requirement of the instrument
fopen(TDS); %connect the VISA-GPIB object to the instrument.
fprintf(TDS,'AUTOSet EXECute'); %This command (no query format) sets the vertical, horizontal, and trigger
controls of the instrument to automatically acquire and display the selected waveform.
fprintf(TDS,'HEADER OFF'); % Turn headers off, this makes parsing easier
fprintf(TDS,'HORizontal:MAIn:DElay:MODE OFF'); % This command sets or queries the time base trigger
delay mode. This command is equivalent to choosing Delay Mode On from the Horiz/ Acq menu.
fprintf(TDS,'SElect:CH1 ON');
fprintf(TDS,'SElect:CH2 ON');
fprintf(TDS,'SElect:CH3 ON');

```



```

-----
%% Se colocan los parametros del Osciloscopio para la parte HORIZONTAL
fprintf(TDS,['HORizontal:POSition ' num2str(Triggerposition)]);
fprintf(TDS,['HORizontal:RECOrdlength ' num2str(Numerodedatapoints)]; %This command sets the
horizontal record length to the number of data points in each frame.
fprintf(TDS,['HORizontal:SCAle ' num2str(Escala_horizontal)]; %This command sets or queries the time per
division for the time base
-----

%% Se colocan los parametros del Osciloscopio para la parte VERTICAL
% Se pondran los parametro correspondientes al canal 1 (fotodiodeo)
fprintf(TDS,'CH1:BANdwidth FULL');
fprintf(TDS,'CH1:COUPling DC');
fprintf(TDS,['CH1:POSition ' num2str(CH1_Posicion)]; %This command sets or queries the vertical position
of the specified channel.
fprintf(TDS,['CH1:SCAle ' num2str(CH1_Escala)]; %This command sets or queries the vertical scale of the
specified channel.
fprintf(TDS,['CH1:TERmination ' num2str(CH1_Impedancia)]; %This command sets the
connected/disconnected status of a 50 Ω resistor, which may be connected between the specified channel's
coupled input and instrument ground.
% Se pondran los parametros correspondientes al canal 2 (trigger)
fprintf(TDS,'CH2:BANdwidth FULL');
fprintf(TDS,'CH2:COUPling DC');
fprintf(TDS,['CH2:POSition ' num2str(CH2_Posicion)];
fprintf(TDS,['CH2:SCAle ' num2str(CH2_Escala)];
fprintf(TDS,['CH2:TERmination ' num2str(CH2_Impedancia)];
% Se pondran los parametros correspondientes al canal 3 (fausto señal)
fprintf(TDS,'CH3:BANdwidth FULL');
fprintf(TDS,'CH3:COUPling DC');
fprintf(TDS,['CH3:POSition ' num2str(CH3_Posicion)];
fprintf(TDS,['CH3:SCAle ' num2str(CH3_Escala)];
fprintf(TDS,['CH3:TERmination ' num2str(CH3_Impedancia)];
-----

%% Modo de adquisicion
fprintf(TDS,['acquire:mode ' ModoDeAquisicion]); %This command sets or queries whether the specified
measurement slot is computed and displayed. if strcmp(ModoDeAquisicion,'AVERAGE')
fprintf(TDS,['acquire:numavg ' num2str(Numerodepromediaciones)]; %This command sets or queries the
number of waveform acquisitions that make up an averaged waveform.
end
fprintf(TDS, 'acquire:state OFF'); %This command starts or stops acquisitions. When state is set to ON or
RUN, a new acquisition will be started. If the last acquisition was a single acquisition sequence, a new single
sequence acquisition will be started. If the last acquisition was continuous, a new continuous acquisition will
be started.
-----

%% Preparacion del trigger para el Channel 2
fprintf(TDS, 'TRIGger:A:EDGE:COUPling DC'); %This command sets or queries the type of coupling for the
A edge trigger.
fprintf(TDS, 'TRIGger:A:EDGE:SLOpe FALL'); %This command sets or queries the slope for the A edge
trigger. fprintf(TDS, 'TRIGger:A:EDGE:SOUrce CH2'); %This command sets or queries the source for the A
edge trigger.
fprintf(TDS, 'TRIGger:A:LEVel TTL');
-----

%% Movimiento del mortor del atenuador
n=1
while n==1
    salirdelloop = 'PONGA UN NUMERO DISTINTO DE 1 CUANDO EL MOTOR ESTE EN LA POSICION
INICIAL Y SE HAYA DETENIDO: ';
    n=input(salirdelloop)
end
writeline(watt_pilot,'m -550')
pause(3)
-----

%% Listas vacia (llenas de ceros) para introducir datos.
cambio_escala_CH3 = double(zeros(Numerodevoltajes,1));

```

```

numero_de_pasos_energia = 11; % aqui va el numero de pasos, que equivale al numero de energias que
utilizaremos.
PASOS_ENERGIA = double(zeros(numero_de_pasos_energia,1)); %lista para guardar los pasos
Dfocal = 75; % distancia focal, se cambia cuando se cambia lente

%Aqui tecleamos los pasos que se ocuparan en el medidor de energia
for H = 1:1:numero_de_pasos_energia
    moverdis = 'introduzca un valor positivo o negativo para mover el disco y presione enter: ';
    d = input(moverdis)
    PASOS_ENERGIA(H,1)=d;
end

%%
%%      E M P I E Z A      E L      E X P E R I M E N T O
%%
%%
fprintf(PS350,'VLIM 5000;VLIM?') %VLIM(?) x The VLIM command sets the value of the voltage limit to x
where x is the value in Volts.
fprintf(PS350,'HVON') % The HVON command turns the high voltage ON provided that the front panel high
voltage switch is not in the OFF position.
pathname = uigetdir;
EL = 0;

-----
% se ponen dos for uno para variar la energia con el atenuador y el otro para variar el voltaje entre las placas.
for j = 1:1:numero_de_pasos_energia

    energia = PASOS_ENERGIA(j,1)
    writeline(watt_pilot,['m ' num2str(energia)])
    EL = EL +PASOS_ENERGIA(j,1);
    EnerPasos = -550 + EL
    pause(2)
    fprintf(TDS, 'CH3:SCAlE 500e-3'); %SE CMABIA PARA CADA CORRIDA DE ENERGIA Y LENTES
    fprintf(TDS, '*WAI');

    for i = 1:Numerodevoltajes

        Voltajedeseado = 200 - (30*(i-1))
        %Voltajedeseado = 200*i;
        fprintf(PS350,['VSET' num2str(Voltajedeseado)]); %VLIM(?) x The VLIM command sets the value of the
voltage limit to
x where x is the value in Volts. T
        pause(3)
        Qswitchdelay = 230e-6;
        pulsotriggerQs = 36e-8 + Qswitchdelay;

        -----
        %% EMPIEZA LOS PULSOS DEL EXPERIMENTO
        %fprintf(TDS, 'CH3:SCAlE 1'); %SE CMABIA PARA CADA CORRIDA DE ENERGIA Y LENTES
        fprintf(TDS, 'acquire:stopafter runstop'); %Da la orden de tomar datos.
        fprintf(TDS, 'acquire:state ON'); %Da la orden de tomar datos.
        fprintf(TDS, '*WAI');

        fprintf(DG,['DLAY 8,5,' num2str(pulsotriggerQs)]); % Set channel A delay to equal channel T0 plus 0 s.
        fprintf(DG,['DLAY 6,5,' num2str(Qswitchdelay)]) % Set channel C delay to equal channel B plus
Qswitchdelay time.
        fprintf(DG,['BURC ' num2str(meropulsosburstQswitch)]); % Set (query) the burst count {to i}.
        fprintf(DG, '*TRG'); % When the DG645 is configured for single shot triggers, this command initiates a single
trigger.
        pause ((meropulsosburstQswitch*0.2)/frecuencialaser)

        -----
        %%
        fprintf(TDS, 'acquire:stopafter sequence'); %Da la orden de tomar datos. sequence
        fprintf(TDS, 'acquire:state ON'); %Da la orden de tomar datos.

```

```

fprintf(TDS, '*WAI');
pause ((numeropulsosburstQswitch*0.3)/frecuencialaser)

-----

%% CAMBIO DE ESCALA SEGUN LA AMPLITUD
fprintf(TDS,'measurement:meas1:source CH3');
fprintf(TDS,'measurement:meas1:type MAXimum');
fprintf(TDS,'measurement:meas1:STATE ON');
fprintf(TDS, '*WAI');

if cambioautomaticoescala == 1
    cambio_escala_CH3 (i,1) = str2double(query(TDS, 'measurement:meas1:value?'));
    cambio_escala_CH3(i,1);
    if cambio_escala_CH3(i,1) > 4.5
        CH3_Escala = 1;
        fprintf(TDS,['CH3:SCAle ' num2str(CH3_Escala)]); %This command sets or queries the vertical scale
of the specified channel.
    else
        if cambio_escala_CH3(i,1) <= 4.5 && cambio_escala_CH3(i,1) > 4
            CH3_Escala = 900e-3;
            fprintf(TDS,['CH3:SCAle ' num2str(CH3_Escala)]);
        else
            if cambio_escala_CH3(i,1) <= 4 && cambio_escala_CH3(i,1) > 3.5
                CH3_Escala = 800e-3;
                fprintf(TDS,['CH3:SCAle ' num2str(CH3_Escala)]);
            else
                if cambio_escala_CH3(i,1) <= 3.5 && cambio_escala_CH3(i,1) > 3
                    CH3_Escala = 700e-3;
                    fprintf(TDS,['CH3:SCAle ' num2str(CH3_Escala)]);
                else
                    if cambio_escala_CH3(i,1) <= 3 && cambio_escala_CH3(i,1) > 2.5
                        CH3_Escala = 600e-3;
                        fprintf(TDS,['CH3:SCAle ' num2str(CH3_Escala)]);
                    else
                        if cambio_escala_CH3(i,1) <= 2.5 && cambio_escala_CH3(i,1) > 2
                            CH3_Escala = 500e-3;
                            fprintf(TDS,['CH3:SCAle ' num2str(CH3_Escala)]);
                        else
                            if cambio_escala_CH3(i,1) <= 2 && cambio_escala_CH3(i,1) > 1.5
                                CH3_Escala = 400e-3;
                                fprintf(TDS,['CH3:SCAle ' num2str(CH3_Escala)]);
                            else
                                if cambio_escala_CH3(i,1) <= 1.5 && cambio_escala_CH3(i,1) > 1
                                    CH3_Escala = 300e-3;
                                    fprintf(TDS,['CH3:SCAle ' num2str(CH3_Escala)]);
                                else
                                    if cambio_escala_CH3(i,1) <= 1 && cambio_escala_CH3(i,1) > 500e-3
                                        CH3_Escala = 200e-3;
                                        fprintf(TDS,['CH3:SCAle ' num2str(CH3_Escala)]);
                                    else
                                        if cambio_escala_CH3(i,1) <= 500e-3 && cambio_escala_CH3(i,1) > 450e-3
                                            CH3_Escala = 100e-3;
                                            fprintf(TDS,['CH3:SCAle ' num2str(CH3_Escala)]);
                                        else
                                            if cambio_escala_CH3(i,1) <= 450e-3 && cambio_escala_CH3(i,1) > 400e-3
                                                CH3_Escala = 90e-3;
                                                fprintf(TDS,['CH3:SCAle ' num2str(CH3_Escala)]);
                                            else
                                                if cambio_escala_CH3(i,1) <= 400e-3 && cambio_escala_CH3(i,1) > 350e-
3
                                                    CH3_Escala = 80e-3;
                                                    fprintf(TDS,['CH3:SCAle ' num2str(CH3_Escala)]);
                                                else

```

```

300e-3      if cambio_escal_a_CH3(i,1) <= 350e-3 && cambio_escal_a_CH3(i,1) >
            CH3_Escal_a = 70e-3;
            fprintf(TDS,['CH3:SCAl_e ' num2str(CH3_Escal_a)]);
            else
250e-3      if cambio_escal_a_CH3(i,1) <= 300e-3 && cambio_escal_a_CH3(i,1) >
            CH3_Escal_a = 60e-3;
            fprintf(TDS,['CH3:SCAl_e ' num2str(CH3_Escal_a)]);
            else
200e-3      if cambio_escal_a_CH3(i,1) <= 250e-3 && cambio_escal_a_CH3(i,1) >
            CH3_Escal_a = 50e-3;
            fprintf(TDS,['CH3:SCAl_e ' num2str(CH3_Escal_a)]);
            else
            if      cambio_escal_a_CH3(i,1)      <=      200e-3      &&
cambio_escal_a_CH3(i,1) > 150e-3
            CH3_Escal_a = 40e-3;
            fprintf(TDS,['CH3:SCAl_e ' num2str(CH3_Escal_a)]);
            else
            if      cambio_escal_a_CH3(i,1)      <=      150e-3      &&
cambio_escal_a_CH3(i,1) > 100e-3
            CH3_Escal_a = 30e-3;
            fprintf(TDS,['CH3:SCAl_e ' num2str(CH3_Escal_a)]);
            else
            if      cambio_escal_a_CH3(i,1)      <=      100e-3      &&
cambio_escal_a_CH3(i,1) > 50e-3
            CH3_Escal_a = 20e-3;
            fprintf(TDS,['CH3:SCAl_e ' num2str(CH3_Escal_a)]);
            else
            if      cambio_escal_a_CH3(i,1)      <=      50e-3      &&
cambio_escal_a_CH3(i,1) > 45e-3
            CH3_Escal_a = 10e-3;
            fprintf(TDS,['CH3:SCAl_e ' num2str(CH3_Escal_a)]);
            else
            if      cambio_escal_a_CH3(i,1)      <=      45e-3      &&
cambio_escal_a_CH3(i,1) > 40e-3
            CH3_Escal_a = 9e-3;
            fprintf(TDS,['CH3:SCAl_e ' num2str(CH3_Escal_a)]);
            else
            if      cambio_escal_a_CH3(i,1)      <=      40e-3      &&
cambio_escal_a_CH3(i,1) > 35e-3
            CH3_Escal_a = 8e-3;
            fprintf(TDS,['CH3:SCAl_e ' num2str(CH3_Escal_a)]);
            else
            if      cambio_escal_a_CH3(i,1)      <=      35e-3      &&
cambio_escal_a_CH3(i,1) > 30e-3
            CH3_Escal_a = 7e-3;
            fprintf(TDS,['CH3:SCAl_e
num2str(CH3_Escal_a)]);
            else
            if      cambio_escal_a_CH3(i,1)      <=      30e-3      &&
cambio_escal_a_CH3(i,1) > 25e-3
            CH3_Escal_a = 6e-3;
            fprintf(TDS,['CH3:SCAl_e
num2str(CH3_Escal_a)]);
            else
            if      cambio_escal_a_CH3(i,1)      <=      25e-3      &&
cambio_escal_a_CH3(i,1) > 20e-3
            CH3_Escal_a = 5e-3;
            fprintf(TDS,['CH3:SCAl_e
num2str(CH3_Escal_a)]);
            else

```

```

cambio_escala_CH3(i,1) > 15e-3
num2str(CH3_Escala));

cambio_escala_CH3(i,1) > 10e-3
num2str(CH3_Escala));

&& cambio_escala_CH3(i,1) > 5e-3
num2str(CH3_Escala));

num2str(CH3_Escala));

if cambio_escala_CH3(i,1) <= 20e-3 &&
    CH3_Escala = 4e-3;
    fprintf(TDS,['CH3:SCAle
else
if cambio_escala_CH3(i,1) <= 15e-3 &&
    CH3_Escala = 3e-3;
    fprintf(TDS,['CH3:SCAle
else
if cambio_escala_CH3(i,1) <= 10e-3
    CH3_Escala = 2e-3;
    fprintf(TDS,['CH3:SCAle
else
if cambio_escala_CH3(i,1) <= 5e-3
    CH3_Escala = 1e-3;
    fprintf(TDS,['CH3:SCAle
end
end
end
end
end
end
end
end
end
end
end
end
end
end
end
end
end
end
end
end
end
end
end
end

fprintf(TDS, '*WAI');
fprintf(TDS, 'DISPLAY:PERSISTENCE:RESET'); %Da la orden de resetear la pantalla.
pause ((numeropulsosburstQswitch*0.1)/frecuencialaser)

-----

%%
fprintf(TDS, 'acquire:stopafter sequence'); %Da la orden de tomar datos.
fprintf(TDS, 'acquire:state ON'); %Da la orden de tomar datos.
fprintf(TDS, '*WAI');
pause((numeropulsosburstQswitch*0.3)/frecuencialaser+2)

-----

%% Toma de Datos CH1 y graficas. Fotodiodo (C A N A L U N O)

```

```

fprintf(TDS,'data:source CH1'); %Select the waveform source.
pause(2)
fprintf(TDS,['data:stop ' recordlengthstring]); %Specify the portion of the waveform that you want to transfer.
fprintf(TDS,'data:start 1'); %Specify the portion of the waveform that you want to transfer.

wfmmoutpre_BN_FMT_CH1 = query(TDS,'wfmmoutpre:BN_FMT?'); %Set or query the format of binary dat for
outgoing waveforms specified by the DATA:Source command.
wavfrm_CH1 = query(TDS,'wavfrm?'); %Transfer waveform preamble information.

XZEro_CH1 = str2double(query(TDS,'WFMMOutpre:XZEro?')); % WFMMOutpre:XZEro Sets or returns the
(sub--sample) time between the trigger sample and the occurrence of the actual incoming waveform trigger
XINcr_CH1 = str2double(query(TDS,'WFMMOutpre:XINcr?')) ;% WFMMOutpre:XINcr Sets or returns the
horizontal sampling interval between incoming waveform points
YMUlt_CH1 = str2double(query(TDS,'WFMMOutpre:YMUlt?')); % WFMMOutpre:YMUlt Sets or returns the
vertical scale factor, per digitizing level, of the incoming waveform points
YOFF_CH1 = str2double(query(TDS,'WFMMOutpre:YOFF?')) ;% WFMMOutpre:YOFF Sets or returns the
vertical position of the incoming waveform in digitizing levels
YZEro_CH1 = str2double(query(TDS,'WFMMOutpre:YZEro?')) ;% WFMMOutpre:YZEro Sets or returns the
offset of the incoming waveform

fprintf(TDS,'data:encdg SRPBinary'); %specify the waveform data format
fprintf(TDS,'curve?'); %% Request the CURVE
y_CH1 = binblockread(TDS,'uint16'); %reads a binblock from the instrument connected to interface object
and returns the values to y
y_CH1 = YMUlt_CH1*(y_CH1-YOFF_CH1)-YZEro_CH1; %Valor real en Volts medidos en el osciloscopio.
y_trans_CH1 = transpose(y_CH1);

j_CH1 = 1:recordlength;
t_CH1 = (j_CH1-1)*XINcr_CH1 + XZEro_CH1; %tiempop en el osciloscopio, partiendo de cero para el punto
extremo de la izquierda.
size(y_trans_CH1);
size(t_CH1);
M_CH1 =[t_CH1;y_trans_CH1];
M_trans_CH1 = transpose(M_CH1);
if CH1_TakeAllWaveform == 1 %Tomar el espectro completo

    dlmmwrite([pathname '/CH1_' num2str(Dfocal) 'mm_' num2str(EnerPasos) 'mJ_' num2str(Voltajedeseado)
'V.txt'],M_trans_CH1,'delimiter','t','precision',['%0.14f %1.6f']);

else % Tomar el espectro desde la posición del trigger.
    Pt_Off_CH1 = str2double(query(TDS,'WFMMOutpre:Pt_Off?')); % WFMMOutpre:Pt_Off This query--only
command returns the trigger point relative to DATA:START for the waveform specified by the DATA:Source
command.
    x_CH1 = t_CH1(Pt_Off_CH1:recordlength)- t_CH1(Pt_Off_CH1); %Tiempo tomando cero la posición
del trigger
    z_CH1 = y_CH1(Pt_Off_CH1:recordlength); %Valores de voltaje desde la posición del trigger

    z_trans_CH1 = transpose(z_CH1);
    figure;
    plot(x_CH1,z_CH1);

    CH1_titulo1_x = 'Tiempo';
    CH1_titulo2_z = 'Voltaje';
    datos_waveform_2 = fopen([pathname '/datos_CH1_' num2str(Voltajedeseado) '.txt'],'w');
    fprintf(datos_waveform_2,[CH1_titulo1_x ' ' CH1_titulo2_z '\n']);

    fprintf(datos_waveform_2,'%15.13f %10.7f \n', [x_CH1 z_trans_CH1]);
    fclose(datos_waveform_2);
end

%% Toma de Datos CH3 y graficas. Fausto señal (C A N A L T R E S)
fprintf(TDS,'data:source CH3');

```

```

pause(2)
fprintf(TDS,['data:stop ' recordlengthstring]);
fprintf(TDS,'data:start 1');
wfmoutpre_BN_FMT_CH3 = query(TDS,'wfmoutpre:BN_FMT?');
wavfrm_CH3 = query(TDS,'wavfrm?');

XZEro_CH3 = str2double(query(TDS,'WFMOupre:XZEro?'));
XINcr_CH3 = str2double(query(TDS,'WFMOupre:XINcr?'));
YMUlt_CH3 = str2double(query(TDS,'WFMOupre:YMUlt?'));
YOff_CH3 = str2double(query(TDS,'WFMOupre:YOff?'));
YZEro_CH3 = str2double(query(TDS,'WFMOupre:YZEro?'));

fprintf(TDS,'data:encdg SRPBinary');
fprintf(TDS,'curve?');
y_CH3 = binblockread(TDS,'uint16');
y_CH3 = YMUlt_CH3*(y_CH3-YOff_CH3)-YZEro_CH3;
y_trans_CH3 = transpose(y_CH3);

j_CH3 = 1:recordlength;
t_CH3 = (j_CH3-1)*XINcr_CH3 + XZEro_CH3;
size(y_trans_CH3);
size(t_CH3);

M_CH3 =[t_CH3;y_trans_CH3];
M_trans_CH3 = transpose(M_CH3);
if CH3_TakeAllWaveform == 1 %Tomar el espectro completo
    dlmwrite([pathname '/CH3_' num2str(Dfocal) 'mm_' num2str(EnerPasos) 'mJ_' num2str(Voltajedeseado)
'V.txt'],M_trans_CH3,'delimiter','\t','precision',['%0.14f %1.6f']);

else % Tomar el espectro desde la posición del trigger.
    Pt_Off_CH3 = str2double(query(TDS,'WFMOupre:Pt_Off?'));
    x_CH3 = t_CH3(Pt_Off_CH3:recordlength)- t_CH3(Pt_Off_CH3);
    z_CH3 = y_CH3(Pt_Off_CH3:recordlength_CH3);
    z_trans_CH3 = transpose(z_CH3);
    figure;
    plot(x_CH3,z_CH3);

    CH3_titulo1_x = 'Tiempo';
    CH3_titulo2_z = 'Voltaje';
    datos_waveform_2_CH3 = fopen([pathname '/datos_CH3_' num2str(Voltajedeseado) '.txt'],'w');
    fprintf(datos_waveform_2_CH3,[CH3_titulo1_x ' ' CH3_titulo2_z '\n']);

    fprintf(datos_waveform_2_CH3,'%15.13f %10.7f \n', [x_CH3 z_trans_CH3]);

    fclose(datos_waveform_2_CH3);
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Toma de datos del medidor de enrgia ophir
[Value Timestamp Status]= ophirApp.GetData(h_USB(1),0);
ValorE = transpose(Value);
length (ValorE);
contador = zeros(1,length(ValorE)) ;
for k = 1:1:(length (ValorE));
    contador(k) = k;
end
contadort = transpose(contador);
figure; plot(contadort,ValorE);
titulo1 = 'Energia';
titulo2 = 'numero de pulsos';
datos = fopen([pathname '/EnerTransmitida_' num2str(Dfocal) 'mm_' num2str(EnerPasos) 'mJ_'
num2str(Voltajedeseado) 'V.txt'],'w');
fprintf(datos,'%f %f \n', [ValorE contadort]);
fclose(datos);

```

```

pause (6)
fprintf(TDS, 'DISPLAY:PERSISTENCE:RESET'); %Da la orden de resetear la pantalla.
fprintf(TDS, '*WAI');
end
end
fprintf(PS350,'HVOF') %The HVOF command turns the high voltage OFF.
-----

%%% Cerrar GENERADOR DE SEÑALES
fclose(DG); %cierra VISA-GPIB objec.
delete(DG); %Borra la variable de VISA-GPIB objec.
clear DG; %limpia la memoria de VISA-GPIB objec.
%%% Cerrar OSCILOSCOPIO
fclose(TDS);
delete(TDS);
clear TDS;
%%% Cerrar MEDIDOR DE ENERGIA
ophirApp.StopAllStreams;
ophirApp.CloseAll;
ophirApp.delete;
clear ophirApp
%%% Cerrar FUENTE DE ALTO VOLTAJE
fclose(PS350); %cierra VISA-GPIB objec.
delete(PS350); %Borra la variable de VISA-GPIB objec.
clear PS350; %limpia la memoria de VISA-GPIB objec.
%%% Cerrar ATENUADOR
delete (watt_pilot)
clear watt_pilot

```

7.13 Programa análisis de señales y energía transmitida

```

%%% Parámetros iniciales para el calculo del tiempo de arribo de la waveform
NumeroElementosPromediacion = 10; % Número de elementos que se toman para determinar el valor medio
de los intervalos a estudiar, para determinar el tiempo de arribo
SaltoElementosPromediacion = 1; % Indica el número de elementos que se salta para determinar el valor
medio de los intervalos a estudiar, para determinar el tiempo de arribo
NumVecesDesvEst = 1.5; % Que número de desviaciones estandar de la promediación se considera para
comparar contra el promedio del ruido.
Puntoinicial = 2781; %Este numero depende del Triggerposition y el Numerodatapoints.
-----

%%%
pathname = uigetdir;
%%% SE QUITA OFFSET DEL RUIDO
RUIDO = load('ruido_20_03_22.txt');
WaveformR = RUIDO(:,2);
TiempoR = transpose(RUIDO(:,1));
tiemponuevoR = TiempoR;
Tiemponuevo_transR = transpose(tiemponuevoR);
[val_max_waverformR,ele_max_waveformR] = max(WaveformR);
lengthwaveformR = length(WaveformR); %Length of waveform.
valor_ceroR = abs(tiemponuevoR(Puntoinicial));
elem_ceroR = Puntoinicial;
mean_vector_ruidoR = mean(WaveformR(1:elem_ceroR)); % Determina el valor medio de la señal de ruido,
para ajustar el offset.
Waveform_nuevoR = WaveformR - mean_vector_ruidoR; % A la Waveform le resta la señal de ruido,
ajustando el offset.
Tiemponuevo_transR = Tiemponuevo_transR*1e9;
SENAL_RUIDO = [Tiemponuevo_transR,Waveform_nuevoR];
dlmwrite([pathname '/Elec_RUIDO_Soff_.txt'],SENAL_RUIDO,'delimiter','\t','precision',['%0.15f %0.15f']);
-----

%%% ANALISIS SEÑALES

```



```

numero_de_pasos_energia = 7; % aqui va el numero de pasos, que equivale al numero de energias que
utilizaremos.
PASOS_ENERGIA = double(zeros(numero_de_pasos_energia,1)); %lista para guardar los pasos
ENERGIA_REAL = double(zeros(numero_de_pasos_energia,1)); %lista para guardar los pasos
lente = 75;
lente1=500;
Numerodevoltajes = 16;
ENERGIA_LISTA = double(zeros(numero_de_pasos_energia,1));
ENERGIA_LISTA_ERROR = double(zeros(numero_de_pasos_energia,1));
%Aquí tecleamos los pasos que se ocuparan en el medidor de energia
for H = 1:1:numero_de_pasos_energia
    moverdis = 'introduzca un valor positivo o negativo para poner la energia y presione enter: ';
    d = input(moverdis)
    PASOS_ENERGIA(H,1)=d;
end
for HH = 1:1:numero_de_pasos_energia
    moverdisc = 'introduzca un valor positivo o negativo para poner la energia y presione enter: ';
    dd = input(moverdisc)
    ENERGIA_REAL(HH,1)=dd;
end

-----

for b=1:1:numero_de_pasos_energia
    PASOSE = PASOS_ENERGIA(b,1);
    ENERGIA = ENERGIA_REAL(b,1)
    ENERGIA_ERROR = ENERGIA*0.04;
    ENERGIA_LISTA (b,1) = ENERGIA;
    ENERGIA_LISTA_ERROR (b,1) = ENERGIA_ERROR;

-----

%% Listas vacia (llenas de ceros) para introducir datos.
MAX_WAVEFORM_CH1 = double(zeros(Numerodevoltajes,1));
FWHM_CH1_completo = double(zeros(Numerodevoltajes,1));
MAX_WAVEFORM_CH3 = double(zeros(Numerodevoltajes,1));
FWHM_CH3_completo = double(zeros(Numerodevoltajes,1));
PEAK2PEAK_CH3 = double(zeros(Numerodevoltajes,1));
RMS_CH3 = double(zeros(Numerodevoltajes,1));
TRAP_AREA_CH3 = double(zeros(Numerodevoltajes,1));
VOLTAJE_R = double(zeros(Numerodevoltajes,1));
ENERGIA_TRANSMITIDA = double(zeros(Numerodevoltajes,1));
ENERGIA_TRANSMITIDA_STD = double(zeros(Numerodevoltajes,1));
Transmitancia_L = double(zeros(Numerodevoltajes,1));
Transmitancia_error_L = double(zeros(Numerodevoltajes,1));
Energia_breakd_L = double(zeros(Numerodevoltajes,1));
Energia_breakd_error_L = double(zeros(Numerodevoltajes,1));
EB_entre_Energia_L = double(zeros(Numerodevoltajes,1));
EB_entre_Energia_error_L = double(zeros(Numerodevoltajes,1));

-----

for B =1:1:Numerodevoltajes
    difpotencial = 200*B;
    Senal_electrica = dlmread(['CH3_' num2str(lente) 'mm_' num2str(PASOSE) 'mJ_' num2str(difpotencial)
'V.txt']);
    f = 0;
    f = f+1;
    Waveform3 = Senal_electrica(:,2);
    Tiempo3 = transpose(Senal_electrica(:,1));
    tiemponuevo3 = Tiempo3 - (15e-8);
    Tiemponuevo_trans3 = transpose(tiemponuevo3);
    [val_max_waverform3,ele_max_waverform3] = max(Waveform3);
    lengthwaveform3 = length(Waveform3); %Length of waveform.
    valor_cero3 = abs(tiemponuevo3(Puntoinicial));
    elem_cero3 = Puntoinicial;
    mean_vector_ruido3 = mean(Waveform3(1:elem_cero3)); % Determina el valor medio de la señal de
ruido, para ajustar el offset.

```

```

Waveform_nuevo3 = Waveform3 - mean_vector_ruido3; % A la Waveform le resta la señal de ruido,
ajustando el offset.
SENAL_RUIDO3 = SENAL_RUIDO(:,2);
Waveform_nuevo3 = Waveform_nuevo3 - SENAL_RUIDO3;
Tiemponuevo_trans3 = Tiemponuevo_trans3*1e9;
WF_normal = Waveform_nuevo3/max(Waveform_nuevo3);
senal_electrica = [Tiemponuevo_trans3,Waveform_nuevo3];
senal_electrica_N = [Tiemponuevo_trans3,WF_normal];
dlmwrite([pathname '/Elec_Soff_SRui_' num2str(lente1) 'mm_' num2str(ENERGIA) 'mJ_'
num2str(difpotencial) 'V.txt'],senal_electrica,'delimiter','\t','precision',['%1.15f %1.15f']);
dlmwrite([pathname '/Elec_Norm_Soff_SRui_' num2str(lente1) 'mm_' num2str(ENERGIA) 'mJ_'
num2str(difpotencial) 'V.txt'],senal_electrica_N,'delimiter','\t','precision',['%1.15f %1.15f']);
%Esta primera parte se quita el offset de la señal electrica y tambien se resta la señal de ruido sin offset
y se guardan las señales en formato txt

```

```

-----
vector_ruido3 = Waveform_nuevo3(1:elem_cero3); % Toma la parte de la señal anterior a la llegada del
pulso del láser, que se considera la señal de ruido
mean_vector_ruido3 = mean(vector_ruido3); % Determina el valor medio de la señal de ruido luego de
ajustar el offset.
desv_est_vect_ruido3 = std(vector_ruido3); % Determina la desviación estandar de los valores
absolutos de la señal de ruido, c
precision_vector_ruido3 = max(abs(vector_ruido3))+ NumVecesDesvEst * desv_est_vect_ruido3; %
Determina la precisión de medición del valor medio del ruido.
lim_superior_cero3 = mean_vector_ruido3 + precision_vector_ruido3; % Límite superior de la señal de
ruido, definido por la desviación estandar
lim_inferior_cero3 = mean_vector_ruido3 - precision_vector_ruido3; % Límite inferior de la señal de
ruido, definido por la desviación estandar
s = 0; % Contador.
p = 1; % valor para próximo ciclo while.
while p == 1
    elem = elem_cero3 + s * SaltoElementosPromediacion;
    if (elem >= ele_max_waveform3)
        error('Se llegó al valor correspondiente al máximo de la señal, sin encontrar el tiempo de arribo!');
    end
    if elem >= (lengthwaveform3 - NumeroElementosPromediacion)
        error('Se llegó al último elemento de ');
    end
    rango analisis = round(elem - NumeroElementosPromediacion/2):round(elem +
NumeroElementosPromediacion/2);
    mean_valor = mean(Waveform_nuevo3(rango analisis));
    desv_est_valor = std(Waveform_nuevo3(rango analisis));
    precision_valor = desv_est_valor;
    lim_superior_valor = mean_valor + precision_valor; % Límite superior de la parte de la señal que se
analiza, definido por la desviación estandar
    lim_inferior_valor = mean_valor - precision_valor; % Límite inferior de la parte de la señal que se
analiza, definido por la desviación estandar
    if (lim_inferior_valor > lim_superior_cero3) || (lim_superior_valor < lim_inferior_cero3)
        % Regresando en el tiempo hasta encontrar un valor cuya magnitud sea la menor antes de que
haya variación en la tendencia de reducción.
        p1 = 1;
        while p1==1
            if abs(Waveform_nuevo3(elem-1)) >= abs(Waveform_nuevo3(elem)) & (elem <=
ele_max_waveform3)
                p1=0;
            end
            if elem <= elem_cero3 % El valor correspondiente al elemento cer es el menor, por tanto se para
la búsqueda
                p1=0;
            end
            elem = elem-1;
        end
        p = 0;
        numerodepulsodellaser(f,1) = B;

```

```

        elementoinicial(f,1) = elem;
        tiempoinicialdelawaveform(f,1) = tiempounuevo3(1,elem);
        A(1,f) = B;
        A(2,f) = tiempounuevo3(1,elem);
    end
    s = s + 1;
end
%Al final del programa elem es el elemento donde empieza a subir la señal obtenida del osciloscopio
[val_max_waverform_nuevo_CH3,ele_max_waveform_nuevo_CH3] = max(Waveform_nuevo3);
[m,n] = min(abs(((Waveform_nuevo3(elem:ele_max_waveform3)))-
(val_max_waverform_nuevo_CH3/2)));
y_mitad = (Waveform_nuevo3((elem+n),1));
el=0;
datos_iguales=double(zeros(5,1));
K=0;
contador_n =2000;

while contador_n < 10000;
    contador_n = 2000 + 1 + el;
    el=el+1;
    y_ = (Waveform_nuevo3((contador_n),1));
    if (y_mitad-(0.025*y_mitad) <= y_) & (y_ <= y_mitad+(0.025*y_mitad)) ;
        K=K+1;
        datos_iguales(K,1) = contador_n;
    end
end
if datos_iguales(2,1) ~= 0
    elem_minimo = datos_iguales(1,1);
    elem_maximo = datos_iguales(2,1);
    CH3_FWHM_completo = (Tiemponuevo_trans3(elem_maximo,1))-
(Tiemponuevo_trans3(elem_minimo,1));
else
    elem_minimo = datos_iguales(1,1);
    elem_maximo = max(datos_iguales);
    CH3_FWHM_completo = (Tiemponuevo_trans3(elem_maximo,1))-
(Tiemponuevo_trans3(elem_minimo,1));
end

FWHM_CH3_completo (B,1) = CH3_FWHM_completo
MAX_WAVEFORM_CH3(B,1) = val_max_waverform_nuevo_CH3;
-----

%%
rms_CH3 = rms(Waveform_nuevo3);
peak2peak_CH3 = peak2peak (Waveform_nuevo3);
el_area=0;
datos_iguales_area=double(zeros(10,1));
G=0;
contador_n_area = ele_max_waveform_nuevo_CH3;
while contador_n_area < 10000;
    contador_n_area = ele_max_waveform_nuevo_CH3 + 1 + el_area;
    el_area=el_area+1;
    y_ = (Waveform_nuevo3((contador_n_area),1));
    if y_ <0 ;
        G=G+1;
        datos_iguales_area(G,1) = contador_n_area;
    end
end
elem; %elemento donde empieza a subir la waveform
elem_minimo_area = datos_iguales_area(1,1); %elemento donde ya se hizo 0 la waveform
Y_area_CH3 = Waveform_nuevo3 (elem:elem_minimo_area,1);
X_area_CH3 = Tiempounuevo_trans3 (elem:elem_minimo_area,1);
AREA_TRAP_CH3 = trapz(X_area_CH3,Y_area_CH3);

```

```

PEAK2PEAK_CH3 (B,1) = peak2peak_CH3;
RMS_CH3 (B,1) = rms_CH3;
TRAP_AREA_CH3 (B,1) = AREA_TRAP_CH3;

VOLTAJE_R(B,1) = difpotencial;

Senal_Optica = dlmread(['CH1_' num2str(lente) 'mm_' num2str(PASOSE) 'mJ_' num2str(difpotencial)
'V.txt']);

f = 0;
f = f+1;
Waveform1 = Senal_Optica(:,2);
Tiempo1 = transpose(Senal_Optica(:,1));
tiemponuevo1 = Tiempo1 - (15e-8);
Tiemponuevo_trans1 = transpose(tiemponuevo1);
[val_max_waveform1,ele_max_waveform1] = max(Waveform1);
lengthwaveform1 = length(Waveform1); %Length of waveform.
valor_cero1 = abs(tiemponuevo1(Puntoinicial));
elem_cero1 = Puntoinicial;

mean_vector_ruido1 = mean(Waveform3(1:elem_cero1)); % Determina el valor medio de la señal de
ruido, para ajustar el offset.
Waveform_nuevo1 = Waveform1 - mean_vector_ruido1; % A la Waveform le resta la señal de ruido,
ajustando el offset.
Tiemponuevo_trans1 = Tiemponuevo_trans1*1e9;
WF_normal1 = Waveform_nuevo1/max(Waveform_nuevo1);
senal_optica = [Tiemponuevo_trans1,Waveform_nuevo1];
senal_optica_N = [Tiemponuevo_trans1,WF_normal1];
dlmwrite([pathname 'Opt_Soff_SRui_' num2str(lente1) 'mm_' num2str(ENERGIA) 'mJ_'
num2str(difpotencial) 'V.txt'],senal_optica,'delimiter','t','precision',[%1.15f %1.15f]);
dlmwrite([pathname 'Opt_Norm_Soff_SRui_' num2str(lente1) 'mm_' num2str(ENERGIA) 'mJ_'
num2str(difpotencial) 'V.txt'],senal_optica_N,'delimiter','t','precision',[%1.15f %1.15f]);
%Esta primera parte se quita el offset de la señal eléctrica y también se resta la señal de ruido sin offset
y se guardan las señales en formato txt

vector_ruido1 = Waveform_nuevo1(1:elem_cero1); % Toma la parte de la señal anterior a la llegada del
pulso del láser, que se considera la señal de ruido
mean_vector_ruido1 = mean(vector_ruido1); % Determina el valor medio de la señal de ruido luego de
ajustar el offset.
desv_est_vect_ruido1 = std(vector_ruido1); % Determina la desviación estandar de los valores
absolutos de la señal de ruido, c
precision_vector_ruido1 = max(abs(vector_ruido1))+ NumVecesDesvEst * desv_est_vect_ruido1; %
Determina la precisión de medición del valor medio del ruido.
lim_superior_cero1 = mean_vector_ruido1 + precision_vector_ruido1; % Límite superior de la señal de
ruido, definido por la desviación estandar
lim_inferior_cero1 = mean_vector_ruido1 - precision_vector_ruido1; % Límite inferior de la señal de
ruido, definido por la desviación estandar
s = 0; % Contador.
p = 1; % valor para próximo ciclo while.
while p == 1
    elem = elem_cero1 + s * SaltoElementosPromediacion;
    if (elem >= ele_max_waveform1)
        error('Se llegó al valor correspondiente al máximo de la señal, sin encontrar el tiempo de arribo');
    end
    if elem >= (lengthwaveform1 - NumeroElementosPromediacion)
        error('Se llegó al último elemento de ');
    end
    rango_analisis = round(elem - NumeroElementosPromediacion/2):round(elem +
NumeroElementosPromediacion/2);
    mean_valor = mean(Waveform_nuevo1(rango_analisis));
    desv_est_valor = std(Waveform_nuevo1(rango_analisis));
    precision_valor = desv_est_valor;

```

```

lim_superior_valor = mean_valor + precision_valor; % Límite superior de la parte de la señal que se
analiza, definido por la desviación estandar
lim_inferior_valor = mean_valor - precision_valor; % Límite inferior de la parte de la señal que se
analiza, definido por la desviación estandar
if (lim_inferior_valor > lim_superior_cero1) || (lim_superior_valor < lim_inferior_cero1)
    % Regresando en el tiempo hasta encontrar un valor cuya magnitud sea la menor antes de que
    haya variación en la tendencia de reducción.
    p1 = 1;
    while p1==1
        if abs(Waveform_nuevo1(elem-1)) >= abs(Waveform_nuevo1(elem)) & (elem <=
ele_max_waveform1)
            p1=0;
        end
        if elem <= elem_cero1 % El valor correspondiente al elemento cer es el menor, por tanto se para
la búsqueda
            p1=0;
        end
        elem = elem-1;
    end
    p = 0;
    numerodepulsodellaser(f,1) = B;
    elementoinicial(f,1) = elem;
    tiempoinicialdelawaveform(f,1) = tiempounuevo1(1,elem);
    A(1,f) = B;
    A(2,f) = tiempounuevo1(1,elem);
end
s = s + 1;
end
%Al final del programa elem es el elemento donde empieza a subir la señal obtenida del osciloscopio

[val_max_waverform_nuevo_CH1,ele_max_waveform_nuevo_CH1] = max(Waveform_nuevo1);
[m,n] = min(abs(((Waveform_nuevo1(elem:ele_max_waveform1)))-
(val_max_waverform_nuevo_CH1/2)));

y_mitad = (Waveform_nuevo1((elem+n),1));
el=0;
datos_iguales=double(zeros(50,1));
K=0;
contador_n =300;

while contador_n < 10000;
    contador_n = 300 + 1 + el;
    el=el+1;
    y_ = (Waveform_nuevo1((contador_n),1));
    if (y_mitad-(0.005*y_mitad) <= y_) & (y_ <= y_mitad+(0.005*y_mitad)) ;
        K=K+1;
        datos_iguales(K,1) = contador_n;
    end
end

elem_minimo = datos_iguales(1,1);
elem_maximo = max(datos_iguales);
CH1_FWHM_completo = (Tiemponuevo_trans1(elem_maximo,1))-
(Tiemponuevo_trans1(elem_minimo,1));
FWHM_CH1_completo (B,1) = CH1_FWHM_completo;
MAX_WAVEFORM_CH1(B,1) = val_max_waverform_nuevo_CH1;
-----
%ENERGIA TRANSMITIDA

Ener_transmitida = dlmread(['EnerTransmitida_' num2str(lente) 'mm_' num2str(PASOSE) 'mJ_'
num2str(difpotencial) 'V.txt']);

Ener_Trans = Ener_transmitida(361:461,1);

```

```

Promedio_energia_trans = mean(Ener_Trans);
transmi_desvi_estan = std(Ener_Trans);
ENERGIA_TRANSMITIDA(B,1) = Promedio_energia_trans*1000;
ENERGIA_TRANSMITIDA_STD(B,1) = transmi_desvi_estan*1000;

Transmitancia = (Promedio_energia_trans*1000)/ENERGIA;
Transmitancia_error =
(Transmitancia)*(((transmi_desvi_estan*1000)/(Promedio_energia_trans*1000)))+(ENERGIA_ERROR/ENERGIA));
Transmitancia_L (B,1) = Transmitancia;
Transmitancia_error_L (B,1) = Transmitancia_error;

end
ANALISIS_ENERGIA_T =
[ENERGIA_TRANSMITIDA,ENERGIA_TRANSMITIDA_STD,Transmitancia_L,Transmitancia_error_L];
ANALISIS_SENAL1 = [VOLTAJE_R,MAX_WAVEFORM_CH1,FWHM_CH1_completo];
ANALISIS_SENAL3 =
[VOLTAJE_R,MAX_WAVEFORM_CH3,TRAP_AREA_CH3,PEAK2PEAK_CH3,RMS_CH3,FWHM_CH3_completo];

dlmwrite([pathname '/ANALISIS_ENERGIATRANSMITIDA_' num2str(ENERGIA)
'mJ.txt'],ANALISIS_ENERGIA_T,'delimiter','\t','precision',['%1.8f %1.8f %1.8f %1.8f ']);
dlmwrite([pathname '/CH3_VOLT_MAXWF_AREATRAP_PEAK_RMS_FWHM_' num2str(ENERGIA)
'mJ.txt'],ANALISIS_SENAL3,'delimiter','\t','precision',['%1.8f %1.8f %1.15f %1.8f %1.8f %1.15f']);
dlmwrite([pathname '/CH1_VOLT_MAXWF_FWHM_' num2str(ENERGIA)
'mJ.txt'],ANALISIS_SENAL1,'delimiter','\t','precision',['%1.8f %1.8f %1.15f']);
end

```

7.14 Programa de integración

```

clear all;
clc;
close all;
instrreset;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% En esta parte se le quita el offset a la señal
format longEng
pathname = uigetdir;
senal_electrica = load('Elec_Soff_SRui_250mm_340mJ_400V.txt');
E_x = senal_electrica (:,2);
E_t = senal_electrica (:,1);
senal_optica = load('Opt_Soff_SRui_250mm_340mJ_400V.txt');
O_x = senal_optica (:,2);
O_t = senal_optica (:,1);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% NORMALIZAR, INTEGRA
O_x_norma = O_x/max(O_x);
Q = cumtrapz(E_x);
E_x_norma = Q/max(Q);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% RESTA DE TIEMPO
[val_max_waverform_E1,ele_max_waverform_E1] = max(Q);
[val_max_waverform_O2,ele_max_waverform_O2] = max(O_x);
t3_max_1= senal_electrica(ele_max_waverform_E1,1);
t3_max_2= senal_optica(ele_max_waverform_O2,1);
t_resta_12 = t3_max_2 - t3_max_1;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Gráficos
T1 = E_t;
T2 = (E_t + t_resta_12);
figure
plot(E_t,E_x_norma,O_t,O_x_norma)

```

```

figure
plot(T2,E_x_norma,T1,O_x_norma)
figure
plot(E_t,E_x)
figure
plot(E_t,Q)
senal_elecIntegral_1 = [T2,E_x_norma];
senal_optIntegral_1 = [T1,O_x_norma];
senal_elecIntegral_2 = [E_t,Q];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Guardo de señales
dlmwrite([pathname
'/Elec_Norm_Integral_250mm_340mJ_400V.txt'],senal_elecIntegral_1,'delimiter','\t','precision',    ['%0.15f
%0.15f']);
dlmwrite([pathname
'/Opt_Norm_Integral_250mm_340mJ_400V.txt'],senal_optIntegral_1,'delimiter','\t','precision',['%0.15f
%0.15f']);
lmwrite([pathname
'/Elec_Integral_250mm_340mJ_400V.txt'],senal_elecIntegral_2,'delimiter','\t','precision',['%0.15f %0.15f']);

```

7.15 Análisis Amplitud(Energía,Voltaje)

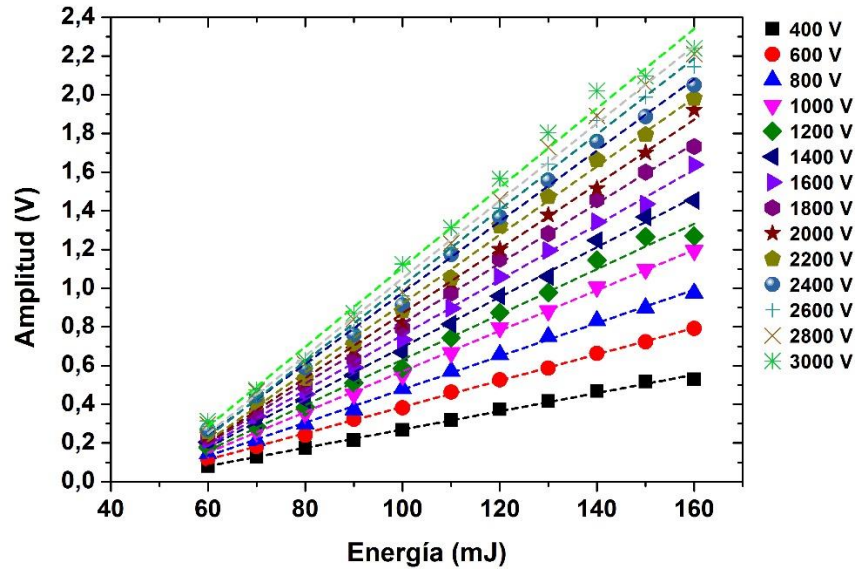
En la sección 3.3.4 se graficó la amplitud del primer pico de la señal eléctrica con respecto a la energía de los pulsos del láser, para seis tensiones, 400 V, 1000 V, 1400, 2000 V, 2400 V y 3000 V. En el siguiente gráfico se muestra el conjunto completo de tensiones, partiendo desde 400 V a 3000 V (Figura 7.11 a)). De igual forma se muestra los gráficos de la amplitud con respecto a las energías. El siguiente gráfico (Figura 7.11 b) representa el conjunto completo de los datos obtenidos del experimento de la sección 2.2.2.2.

Se hallaron los tres valores de las constantes asociadas a la ecuación **(3-1)** para cada uno de los catorce voltajes. Para hallar cada uno de los valores se usó el método de mínimos cuadrados. En la siguiente tabla se muestra el valor, el error y el coeficiente de determinación ajustado para cada uno de los valores, m y b (Tabla 7.3).

Voltaje	Valor b	Error	Valor m	Error	R_A^2
400	-0,202	0,012	4,71E-03	1,0E-04	0,994
600	-0,293	0,006	6,80E-03	6E-05	0,999
800	-0,381	0,014	8,58E-03	1,2E-04	0,998
1000	-0,478	0,015	1,05E-02	1,3E-04	0,998
1200	-0,541	0,039	1,17E-02	3,4E-04	0,990
1400	-0,598	0,025	1,29E-02	2,2E-04	0,997
1600	-0,670	0,025	1,43E-02	2,2E-04	0,997
1800	-0,731	0,023	1,55E-02	2,0E-04	0,998
2000	-0,801	0,033	1,67E-02	2,9E-04	0,996
2200	-0,853	0,040	1,77E-02	3,5E-04	0,995
2400	-0,863	0,039	1,84E-02	3,4E-04	0,996
2600	-0,941	0,047	1,96E-02	4,1E-04	0,995
2800	-0,951	0,050	2,00E-02	4,3E-04	0,995
3000	-0,946	0,069	2,06E-02	6E-04	0,990

Tabla 7.3. Se muestra los valores y errores de los coeficientes m y b .

a)



b)

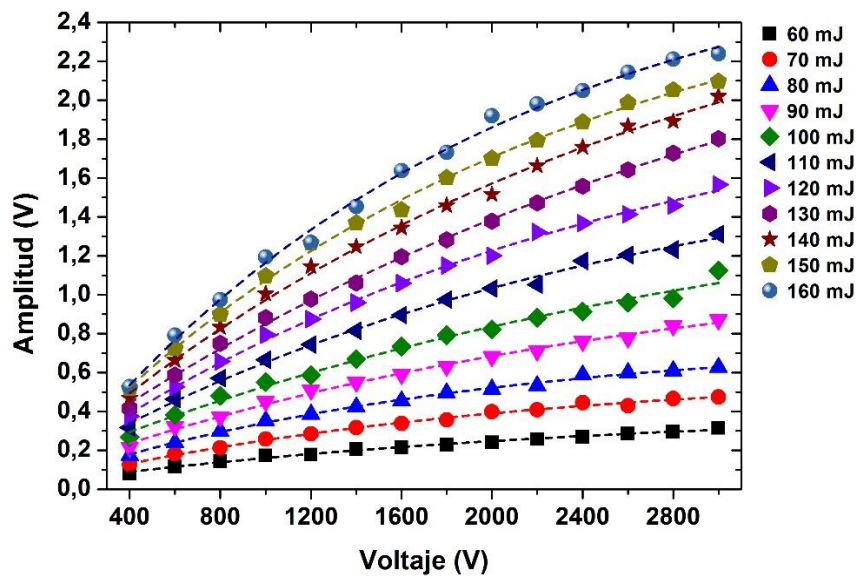


Figura 7.11. Gráficos de la amplitud en función de la energía y el voltaje para una distancia focal de 100 mm. Se muestra las curvas de ajuste.