

# Trabajo Fin de Master

Linealización del barido de láseres  
sintonizables para caracterización espectral de  
alta resolución de dispositivos fotónicos

Linealization of tunable laser sweeps for the  
high resolution spectral characterization of  
photonics devices

Autor

Jorge Ciudad Real Lacuesta

Director

Carlos David Heras Vila

Master en Ingeniería de Telecomunicaciones



2025

*“Students don’t need a perfect teacher. Students need a happy teacher, who’s gonna make them excited to come to school and grow a love for learning”*

*Richard Phillips Feynman*

# **Índice**

<b>Índice.....</b>	<b>3</b>
<b>1. Resumen-objetivos .....</b>	<b>4</b>
<b>2. Introducción .....</b>	<b>5</b>
<b>3. Estudio teórico .....</b>	<b>7</b>
<b>3. 1 Teoría interferencial .....</b>	<b>7</b>
<b>3.2 Cálculo de la fase del interferograma.....</b>	<b>10</b>
<b>3.3 Cálculo de retardo de grupo.....</b>	<b>11</b>
<b>3.4 OFDR .....</b>	<b>12</b>
<b>4. Sistema de medida .....</b>	<b>14</b>
<b>5. Proceso de medida .....</b>	<b>16</b>
<b>5.1 Corrección de la longitud de onda.....</b>	<b>16</b>
<b>5.2 Calibración de longitud de onda mediante celda de gas .....</b>	<b>20</b>
<b>5.3 Medida del DUT .....</b>	<b>21</b>
<b>6. Medidas .....</b>	<b>22</b>
<b>6.1 Retardo de grupo.....</b>	<b>22</b>
<b>6.2 OFDR .....</b>	<b>25</b>
<b>6.3 Medidas en chip .....</b>	<b>28</b>
<b>7. Conclusiones y líneas futuras.....</b>	<b>30</b>
<b>8. Anexos .....</b>	<b>31</b>
<b>Anexo 1. Velocidad de fase y velocidad de grupo .....</b>	<b>31</b>
<b>Anexo 2. Control SCPI TLS LabJack y DAC NI .....</b>	<b>33</b>
<b>Anexo 3. Corrección de longitud de onda .....</b>	<b>39</b>
<b>Anexo 4. Procesado medida retardo de grupo .....</b>	<b>41</b>
<b>Anexo 5. Programa ODFR .....</b>	<b>46</b>
<b>9. Bibliografía .....</b>	<b>47</b>

## **1. Resumen-objetivos**

Los láseres sintonizables son herramientas esenciales para la caracterización de dispositivos fotónicos, ya que permiten realizar barridos en longitud de onda para analizar su comportamiento espectral en términos de potencia y fase. Sin embargo, su principal limitación radica en la no linealidad en la velocidad de barrido de la longitud de onda con el tiempo, lo que genera errores significativos en aplicaciones que requieren alta precisión.

El objetivo principal de este trabajo es desarrollar un método de linealización de la medida para barridos de láseres sintonizables que permita la caracterización espectral precisa de dispositivos fotónicos con alta resolución. En este trabajo se ha desarrollado y validado un sistema de linealización en longitud de onda basado en la linealización de la fase medida en un interferograma de fibra óptica mediante técnicas de procesado de señal y un algoritmo de corrección de longitud de onda. Este sistema ha sido aplicado y validado en dos medidas de alta precisión: el análisis de retardo de grupo en fibras ópticas y la reflectometría en el dominio de la frecuencia óptica mediante la técnica *OFDR (Optical Frequency Domain Reflectometer)*. Los resultados obtenidos en medidas experimentales sobre componentes de fibra óptica y en dispositivos de óptica integrada demuestran la eficacia del método, logrando una precisión superior a 1 pm en la longitud de onda en todo el rango de barrido.

## 2. Introducción

En lo que se refiere al campo de la óptica y de la fotónica, es imprescindible caracterizar dispositivos ópticos y realizar medidas para conocer el comportamiento de estos y así poder implementarlos en sistemas más complejos. Esta caracterización se ha convertido en un proceso esencial para la evolución de tecnologías basadas en óptica integrada, donde los dispositivos fotónicos juegan un papel fundamental en la mejora de la velocidad y eficiencia de las comunicaciones y otras aplicaciones tecnológicas.

Los láseres sintonizables (*TLS, Tunable Laser Source*) son instrumentos clave para la caracterización de dispositivos fotónicos, ya que permiten realizar barridos en longitud de onda ( $\lambda$ ) para analizar su comportamiento espectral en términos de potencia y fase. Sin embargo, presentan una limitación significativa: la no linealidad en la velocidad de barrido de la longitud de onda de los TLS con el tiempo. En mediciones espetrales, se asume generalmente que el barrido de longitud de onda sigue una relación lineal con el tiempo, lo cual sirve de base para calcular parámetros como la potencia o la fase en función de la longitud de onda. No obstante, en aplicaciones con requisitos de alta precisión, que incluyen técnicas interferométricas como *OFDR (Optical Frequency Domain Reflectometry)* o medidas de retardo de grupo, incluso desviaciones del orden de picómetros en la longitud de onda pueden causar errores significativos e inaceptables en los parámetros ópticos medidos, afectando gravemente la caracterización de dispositivos, incluidos los de óptica integrada.

La no linealidad en el barrido de los *TLS* tiene su origen en las respuestas temporales de los mecanismos internos que regulan el proceso de ajuste de la longitud de onda durante el barrido. Estos mecanismos, encargados de la autocalibración del láser, generan aceleraciones y desaceleraciones en el ajuste de la cavidad resonante, lo que resulta en una tasa de cambio de la longitud de onda que no es constante, con regiones donde varía de forma más rápida y otras más lenta. Esta irregularidad afecta directamente la precisión de las mediciones espetrales cuando se requieren altos niveles de exactitud.

En la literatura se han propuesto diferentes métodos para la corrección posterior, tras el proceso de captura, de la no linealidad del barrido de un *TLS*. Badar et al [1] propone un método para la corrección de la no linealidad del barrido para *OFDR*. Este método integra un pico de referencia dentro del espectro del *OFDR*, generado por una línea de retardo óptico en la configuración experimental. Este pico actúa como un “interferómetro virtual”, permitiendo extraer la información de fase del *TLS* sin necesidad de componentes adicionales. Una vez capturada la señal, realizan distintas simulaciones para modelar el comportamiento del *TLS* con una ecuación ( $f_{opt}(t) = \beta + \alpha \cos(2\pi f_t)$ , donde  $\alpha$  es la amplitud de desviación,  $\beta$  es la tasa promedio de sintonización y  $f_t$  una frecuencia de caracterización). La problemática de este método es que no es genérico, pues depende de las características específicas del *TLS* utilizado. Esto implica que, al cambiar de fuente láser, será necesario recalibrar el sistema para adaptarse a las nuevas no linealidades.

Por otro lado, el artículo de Soller et al [2] presentan una técnica para *OFDR* que utiliza un interferómetro de disparo (*trigger interferometer*) para corregir la no linealidad. Este método realiza una corrección no continua del barrido, ya que solo actúa en puntos concretos de la longitud de onda, específicamente aquellos que coinciden con los momentos en los que se produce el *trigger*. Como resultado, no se lleva a cabo una corrección continua, dejando sin ajustar todos los puntos entre los momentos de disparo, lo que introduce imprecisiones en la linealización del barrido.

Finalmente, en el trabajo de Yu-Kuan Tsai et al [3], se propone una modulación pre-distorsionada para corregir la no linealidad para sistemas FMCW LiDAR. Este método usa un esquema homodino que mide la frecuencia de batido entre dos haces de luz generados por el mismo laser, separados por una línea de retardo óptico. A partir de este batido, se extrae la frecuencia de barrido real mediante la Transformada de Hilbert, la cual se compara con una señal ideal. La modulación del láser se ajusta iterativamente mediante una señal triangular pre-distorsionada, logrando un barrido lineal de alta precisión. Sin embargo, este método sufre limitaciones como la velocidad de adquisición, afectando la detección de velocidades muy bajas (< 5 mm/s).

En este trabajo se ha llevado a cabo la linealización de las medidas de un *TLS*, basada en la linealización de la medida de fase respecto a la frecuencia óptica del interferograma obtenido mediante un interferómetro de fibra óptica. El objetivo principal es alcanzar una precisión superior a 1 pm en todo el rango de longitudes de onda. Este método de linealización garantiza una caracterización precisa de la potencia óptica en función de la longitud de onda, permitiendo obtener medidas de fase de muy alta precisión, necesarias para determinar con exactitud los valores de retardo de grupo y las curvas de reflectometría en dispositivos ópticos.

En el capítulo tres de la memoria se presenta un estudio teórico detallado sobre el fenómeno de interferencia, que permite extraer la fase y analizarla posteriormente en el contexto del estudio del retardo de grupo y del *OFDR*.

En el capítulo cuatro se describe el sistema de medida, destacando la programación del control del *TLS* mediante comandos *SCPI* y la adquisición de datos a través de un *DAQ* utilizando *Matlab*.

Posteriormente, en el capítulo cinco se explica el procesado de señal que se ha aplicado para llevar a cabo la corrección de la longitud de onda con el fin de linealizar la fase. Finalmente, en el capítulo seis se detalla la implementación de esta corrección en distintas mediciones de retardo de grupo, *OFDR* y aplicaciones en chips ópticos.

### 3. Estudio teórico

En este capítulo se lleva a cabo un análisis teórico detallado sobre los fundamentos de la interferencia óptica, comenzando con la descripción del fenómeno interferencial y su relación con el interferograma obtenido en nuestras medidas experimentales. A partir de este interferograma, se explica el proceso de extracción de la fase y su importancia en el análisis de los datos. Posteriormente, se aborda el concepto de retardo de grupo, detallando su cálculo y relevancia en la caracterización de sistemas ópticos. Finalmente, se incluye una sección dedicada a la técnica de *OFDR* destacando su aplicación y utilidad en el estudio del *DUT*.

#### 3. 1 Teoría interferencial

La función de onda  $\psi(r, t)$  que describe el comportamiento de las ondas en el espacio y el tiempo, depende tanto de tiempo  $t$  como de la posición espacial  $r = (x, y, z)$ . Esta expresión se obtiene a partir de las ecuaciones de *Maxwell* y cumple con la ecuación de onda escalar, que se define como [4],[5]:

$$\psi(r, t) = \frac{1}{c^2} \frac{\partial^2 \psi(r, t)}{\partial t^2} \quad [\text{ec 3.1}]$$

Una propiedad de esta función de onda es que es lineal, y por ello el principio de superposición puede aplicarse si hay una interacción de dos o más funciones de onda. Si dos campos tienen un solapamiento, el campo resultante estará dado por la superposición lineal de ambas ondas, siendo:

$$\psi_{total}(r, t) = \psi_1(r_1, t) + \psi_2(r_2, t) \quad [\text{ec 3.2}]$$

Una de las soluciones más simples (la cual usaremos en nuestro desarrollo) para la ecuación de una onda plana monocromática es:

$$\psi(r, t) = E_0 e^{i(\vec{\beta} \cdot \vec{r} - \omega t)} = E_0 e^{i(\beta_x x + \beta_y y + \beta_z z - \omega t)} \quad [\text{ec 3.3}]$$

Donde  $E_0$  es la amplitud del campo,  $\vec{\beta} = \vec{k}n$  el vector propagación y  $\omega = 2\pi f$  la frecuencia angular. Dado que supondremos que solo se propaga en  $z$ , tendremos:

$$\psi(r, t) = E_0 e^{i(\beta z + \omega t)} \quad [\text{ec 3.4}]$$

Es importante aclarar que estamos utilizando notación exponencial, ya que para los cálculos resulta más sencillo trabajar en notación compleja para la descripción de los campos. Sin embargo, hay que señalar que el campo físico que realmente existe es real y queda descrito por la parte real de esta notación, es decir, estará descrito por una función coseno.

El campo eléctrico resultante de la superposición de dos ondas planas monocromáticas copropagantes a la misma frecuencia se puede expresar como:

$$E_{total} = E_1 e^{i(\beta z_1 + \omega t)} + E_2 e^{i(\beta z_2 + \omega t)} \quad [\text{ec 3.5}]$$

Dada la alta frecuencia de la luz óptica ( $\sim 10^{14}$  Hz), la detección de estas ondas electromagnética mediante detectores ópticos conlleva de forma inherente un promediado temporal de la portadora tal y como se indica en la siguiente expresión, ya que los detectores son lentos respecto a la portadora y no siguen las oscilaciones rápidas del campo eléctrico:

$$\langle f(t) \rangle_T = \frac{1}{T} \int_t^{t+T} f(t') dt' \quad [\text{ec 3.6}]$$

Dado que la intensidad es proporcional al valor del promedio temporal del campo según  $I = \frac{1}{2} \epsilon_0 c \langle |E|^2 \rangle$ , que, por simplicidad, aproximaremos a  $I \propto \langle |E^2| \rangle$ , tenemos que:

$$\langle |E|^2 \rangle = \langle (E_1 + E_2)(E_1 + E_2)^* \rangle = \langle E_1^2 \rangle + \langle E_2^2 \rangle + 2\langle |E_1 E_2| \rangle \quad [\text{ec 3.7}]$$

$$\begin{aligned} I_1 &= \langle |E_1^2| \rangle_T \\ I_2 &= \langle |E_2^2| \rangle_T \\ I_{12} &= \langle |E_1 E_2| \rangle_T \end{aligned}$$

Además, dado que la magnitud que se va a observar es real, tenemos:

$$|E_1 E_2| = \Re\{E_1 E_2\} = |E_{01}| \cos(\beta z_1 + \omega t) |E_{02}| \cos(\beta z_2 + \omega t)$$

Aplicando que  $2\cos(A)\cos(B) = \cos(A+B) + \cos(A-B)$ , se obtiene:

$$|E_{01}| |E_{02}| (\cos(\beta z_1 + \omega t + \beta z_2 + \omega t) + \cos(\beta z_1 + \omega t - \beta z_2 - \omega t))$$

Dado que las oscilaciones tan rápidas no las podremos seguir en la detección, el término de la izquierda lo descartaremos, quedándonos con un término interferencial dado por:

$$|E_{01}| |E_{02}| \cos(\beta z_1 + \omega t - \beta z_2 - \omega t) = |E_{01}| |E_{02}| \cos(\beta(z_1 - z_2))$$

siendo  $\beta = \frac{2\pi}{\lambda} n(\lambda)$  el número de ondas por la variación del índice de refracción en la fibra y  $\Delta z = z_1 - z_2$  la diferencia de caminos.

Se tiene que el término de interferencia queda:

$$|E_1 E_2| = |E_{01}| |E_{02}| \cos\left(\frac{2\pi}{\lambda} n(\lambda) \Delta z\right) = \sqrt{I_1 I_2} \cos\left(\frac{2\pi}{\lambda} n(\lambda) \Delta z\right)$$

Quedando finalmente:

$$I_t = I_1 + I_2 + 2\sqrt{I_1 I_2} \cos\left(\frac{2\pi}{\lambda} \Delta n(\lambda) \Delta z\right) \quad [\text{ec 3.8}]$$

Siendo el termino de interferencia dependiente de la diferencia de fases:

$$\Delta\varphi(\lambda) = \varphi_1 - \varphi_2 = \frac{2\pi}{\lambda} n(\lambda) \Delta z = \frac{\omega}{c} \Delta n(\lambda) \Delta z \quad [\text{ec 3.9}]$$

Obteniendo finalmente:

$$I_t = I_1 + I_2 + 2\sqrt{I_1 I_2} \cos(\Delta\varphi(\lambda)) \quad [\text{ec 3.10}]$$

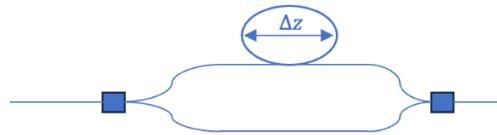
Esta diferencia de fases juega un importante papel dado que es la que determina si la interferencia es constructiva o destructiva, siendo:

- Constructiva si:  $\Delta\varphi = 2n\pi, n \in \mathbb{Z} \rightarrow \cos(\Delta\varphi(\lambda)) = 1$
- Destructiva si:  $\Delta\varphi = (2n - 1)\pi, n \in \mathbb{Z} \rightarrow \cos(\Delta\varphi(\lambda)) = -1$

Donde la intensidad máxima será cuando tengamos interferencia constructiva, quedando  $I_{max} = I_1 + I_2 + 2\sqrt{I_1 I_2}$ , e intensidad mínima cuando tengamos interferencia destructiva, quedando  $I_{min} = I_1 + I_2 - 2\sqrt{I_1 I_2}$ .

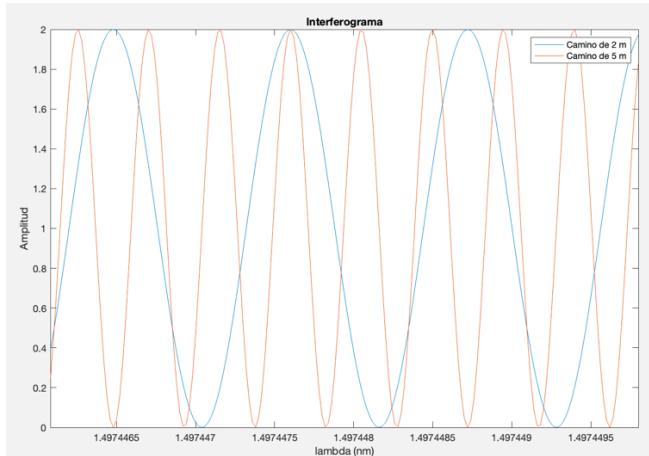
A partir de la **ec 3.8**, se puede extraer que a medida que aumenta la diferencia de caminos  $\Delta z$ , la variación de la fase con la longitud de onda  $\Delta\varphi(\lambda)$  es mayor, lo que incrementa el número de batidos del interferograma.

En nuestro caso, de forma general, los montajes de interferometría son los mostrados en la **Figura 1**, con una diferencia de caminos dada por la diferencia de longitud de dos fibras que forman el interferómetro.



**Figura 1:** Diferencia de caminos entre 2 fibras

A modo de ejemplo de los interferogramas esperados, se muestra en la **Figura 2** una simulación en *Matlab* para una diferencia de caminos de dos y tres metros, pudiéndose observar el incremento de frecuencia producido por el incremento de  $\Delta z$ :



**Figura 2:** En azul, diferencia de batidos para un  $\Delta z$  de 2 m, y en naranja, para un  $\Delta z$  de 5 m

Dado que la medida de potencia óptica implica un promedio temporal de los campos, es necesario que las dos ondas que interfieren mantengan una relación de fase constante, esto es, que las dos ondas sean coherentes, para que este promedio temporal no tienda a cero. En este trabajo se usa un sistema homodino con diferencia de caminos pequeña a efectos de propagación de la luz, hasta unos pocos metros de fibra óptica, y se implementan velocidades de barrido del láser de 0.5 nm/s a 5 nm/s. Así, a efectos prácticos, es buena aproximación considerar que las dos ondas que se superponen a la salida del interferómetro son de la misma longitud de onda dentro del proceso de barrido del TLS.

### 3.2 Cálculo de la fase del interferograma

Como se mencionó en el apartado anterior, tras el desarrollo teórico, se obtiene la siguiente ecuación de interferencia:

$$I_t = I_1 + I_2 + 2\sqrt{I_1 I_2} \cos(\Delta\varphi(\lambda))$$

La fase  $\Delta\varphi(\lambda)$  no es posible calcularla de manera directa del interferograma dado que la intensidad medida  $I_t$  es una magnitud real. Podría calcularse mediante una *Transformada de Hilbert*, pero como las señales que se medirán en las aplicaciones finales pueden ser ruidosas, es preferible extraer  $\Delta\varphi(\lambda)$  mediante un filtrado previo en el espectro de frecuencias, según el proceso mostrado en la siguiente Figura 3:



**Figura 3:** Esquema a seguir para la extracción de la fase

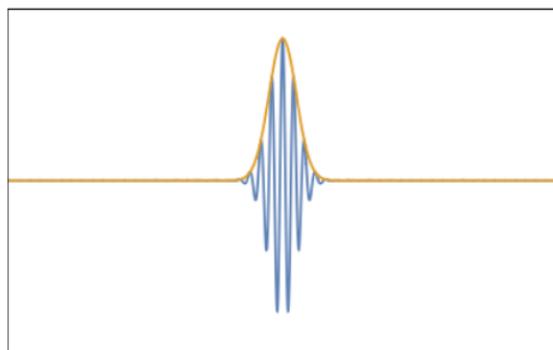
Siguiendo el esquema, y dado que el  $\cos(\Delta\varphi(\lambda)) = \frac{e^{i\Delta\varphi(\lambda)} + e^{-i\Delta\varphi(\lambda)}}{2}$ , aplicando la *Transformada de Fourier* a la señal coseno, se obtiene que [6]:

$$\mathcal{F}\{\cos(\Delta\varphi(\lambda))\} = \pi[\delta(\omega - \omega_0) + \delta(\omega + \omega_0)] \quad [\text{ec 3.11}]$$

De este modo, se obtienen las dos componentes frecuenciales asociadas al coseno, y para obtener la fase basta con quedarse con uno de los dos términos, que en este caso será la parte de frecuencia positiva  $\delta(\omega - \omega_0)$ . Para ello, se aplica un enventanado a esta componente. Una vez enventanada la señal, si se aplica la Transformada de Fourier Inversa, obtenemos la exponencial compleja positiva  $e^{i\Delta\varphi(\lambda)}$ , de la cual se puede extraer directamente  $\Delta\varphi$ , ya que está contenida en la parte imaginaria de esta.

### 3.3 Cálculo de retardo de grupo

Un paquete de ondas [7],[8], como se muestra en la **Figura 4**, es una onda localizada formada por la superposición de muchas ondas individuales, cada una con frecuencias y longitudes de onda ligeramente diferentes. Este fenómeno da lugar a que los diferentes componentes de frecuencia se propaguen a distintas velocidades, lo que provoca que el paquete de ondas se disperse y se extienda a lo largo del tiempo.



**Figura 4:** Representación de un grupo

La dispersión del paquete de ondas se debe a las diferencias en las velocidades de fase y de grupo de sus componentes. La velocidad de fase describe cómo se mueven las crestas y valles individuales de las ondas, es decir, la velocidad de propagación de una onda tal y como la conocemos y que se define por:

$$v_f = \frac{\omega}{k} = \frac{c}{n_{eff}} \quad [\text{ec 3.12}]$$

siendo  $n_{eff}$  el índice de reflexión efectivo. La velocidad de grupo,  $v_g$ , describe la velocidad a la que viaja la envolvente del paquete a través de un medio definiéndose como:

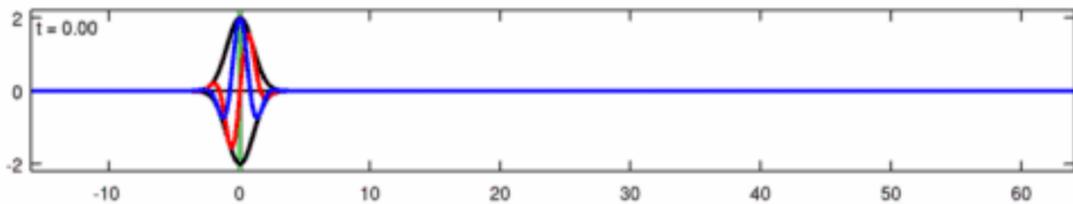
$$v_g = \frac{d\omega}{dk} = \frac{c}{n_{eff} - \lambda \frac{dn_{eff}}{d\lambda}} = \frac{c}{n_g} \quad [\text{ec 3.13}]$$

donde  $n_g$  el índice de grupo. En el **Anexo I** se presenta un desarrollo más detallado para completar las diferencias entre la velocidad de grupo y la velocidad de fase.

La variación de la velocidad de grupo con respecto a  $k$  determina la dispersión cromática, que es el fenómeno que ocurre en medios dispersivos, como en las fibras ópticas, en el cual las diferentes longitudes de onda se propagan a diferentes velocidades debido a la dependencia del índice de refracción del medio con respecto a la longitud de onda,

traduciéndose también en un ensanchamiento temporal del pulso a medida que se propaga tal y como muestra la **Figura 5**, y que se expresa como:

$$D = \frac{d^2\omega}{dk^2} \quad [3. \text{ ec. 14}]$$



**Figura 5:** Representación de la dispersión cromática

Esta variación de la velocidad de grupo afecta directamente al retardo de grupo, que es el tiempo que tarda un paquete de ondas en propagarse. Este retardo está relacionado con la fase  $\Delta\varphi$ , obtenida en el apartado anterior, y se define como:

$$\tau_g = \frac{d\varphi}{d\omega} \approx \frac{\Delta\varphi}{\Delta\omega} \quad [\text{ec 3.15}]$$

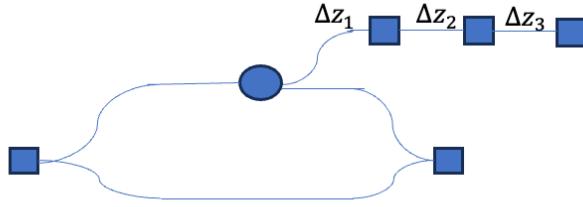
Y relacionando el retardo de grupo con la dispersión cromática, se puede definir como:

$$D = \frac{d\tau_g}{dt} \quad [\text{ec 3.16}]$$

Si no hay dispersión cromática, el retardo de grupo es constante con la longitud de onda, y si el medio es dispersivo, el retardo de grupo no será constante con la longitud de onda.

### 3.4 OFDR

La reflectometría óptica es una técnica fundamental para la caracterización de sistemas ópticos, donde la técnica de medida *OFDR* (*Optical Frequency Domain Reflectometry*) destaca por su alta sensibilidad y resolución. Esta técnica emplea patrones de interferencia, interferogramas, generados en el barrido de un *TLS* para mapear las propiedades del dispositivo a ensayar o *DUT* (*Device Under Test*) donde el *DUT* es colocado en uno de los dos brazos de un interferómetro, trabajando en reflexión, tal y como se muestra en la **Figura 6**. El sistema de medición *OFDR* se centra por tanto en el análisis de las señales reflejadas por un *DUT*. En el esquema mostrado, para mejor comprensión de la técnica *OFDR*, el *DUT* se describe como múltiples puntos de reflexión, que generan múltiples reflexiones a lo largo del recorrido de la onda en su propagación por el *DUT*, cada uno con una distancia diferente  $\Delta z$ .

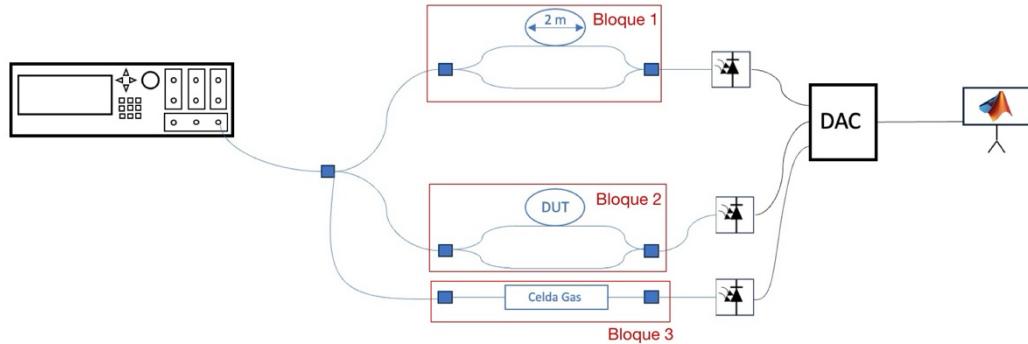


**Figura 6:** Montaje medida OFDR

A diferencia del sistema de interferencia de dos caminos, en el *OFDR* se produce un conjunto de interferencias entre múltiples frentes de onda generados por las distintas reflexiones  $\Delta z$ . Cada camino  $\Delta z$  genera frentes de onda con frecuencias de batido específicas, que dependen directamente de la longitud del camino óptico. Además, como se ha ilustrado en la **Figura 2**, cuanto mayor sea  $\Delta z$ , mayor será la frecuencia de batido asociada. El interferograma resultante puede interpretarse por tanto como la superposición de todos los patrones de interferencia generados. Al aplicar la *Transformada de Fourier* sobre el interferograma obtenido, el espectro frecuencial resultante se manifiesta como un conjunto de *deltas de Dirac*, cada una correspondiente a una frecuencia de batido asociada a una  $\Delta z$ . Esto permite identificar con alta precisión las reflexiones en el *DUT* y caracterizar sus propiedades ópticas con gran resolución espacial.

## 4. Sistema de medida

El esquema general que se usa para las medidas es el que se muestra en la **Figura 7**:



**Figura 7:** Esquema general de medida

El sistema se divide en 3 bloques de medida bien diferenciados:

- El bloque 1, el interferómetro de referencia, que se utiliza para obtener la corrección en longitud de onda que luego se aplica al resto de bloques.
- El bloque 2, el interferómetro en el que se encuentra el DUT, que puede ser cualquier dispositivo que se quiera caracterizar y que trabajará en transmisión en el caso de medir el retardo de grupo o en reflexión si se quiere obtener la reflectometría por OFDR.
- El bloque 3, formado directamente por la celda de gas para poder realizar una corrección de longitud de onda absoluta

En el montaje se utiliza un Láser Sintonizable o *TLS* modelo *Agilent 8164B* [9], el cual incluye un módulo láser *81600B* y de tres detectores, modelos *81637B*. Estos detectores se encargan de convertir la señal óptica a eléctrica, los cuales se conectan a la adquisidora de datos *DAQ* mediante cable coaxial. La *DAQ* empleada es de *National Instruments* modelo *NI USB-6281* [10], con una tasa de muestreo  $R_b = 625 \text{ Kb/s}$ . Este valor es importante, ya que la tasa de muestreo determina la calidad de la señal en función del número de canales  $N$  utilizados, cumpliendo la relación  $f_s = \frac{R_b}{N}$ . Las fibras utilizadas son tipo estándar monomodo SMF G.652 de *Corning* y los divisores ópticos son 50/50 [11].

En el bloque 2, en la rama del interferómetro indicada como *DUT*, se conecta el dispositivo que se desea caracterizar, que puede ser simplemente una fibra para realizar pruebas de validación o de calibración, un chip óptico a caracterizar o cualquier elemento óptico de interés.

En el bloque 3 se integra una celda de gas que permite verificar la corrección absoluta de la longitud de onda. La celda de gas que se utiliza es el modelo QUAD-20/5/150/150 de *Wavelength References* [12]. El modelo *QUAD* indica que el número de gases que contiene es de 4, siendo  $H_2O + ^{12}C_2H_2 + ^{12}CO + ^{13}CO$  entre las longitudes de onda de 1350-1633nm.

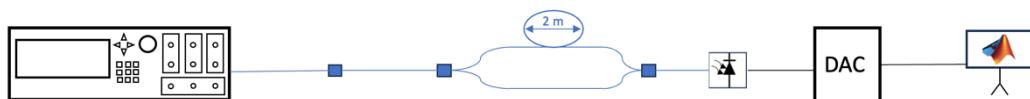
Tanto el control automático del láser sintonizable mediante comandos *SCPI* como la programación para la adquisición de datos a través de la *DAQ* se implementan utilizando *Matlab* y se detallan en el **ANEXO 2**.

## 5. Proceso de medida

En este capítulo se detalla el proceso de corrección de longitud de onda, tanto relativa como absoluta, y se describe la metodología empleada para la medición del *DUT* según la aplicación correspondiente. Los códigos implementados para llevar a cabo las diferentes medidas se recogen en los **Anexos**: el código para la corrección de longitud de onda está en el **Anexo 3**, el código para la medida del retardo de grupo en el **Anexo 4**, y el código para la realización de las pruebas de OFDR en el **Anexo 5**.

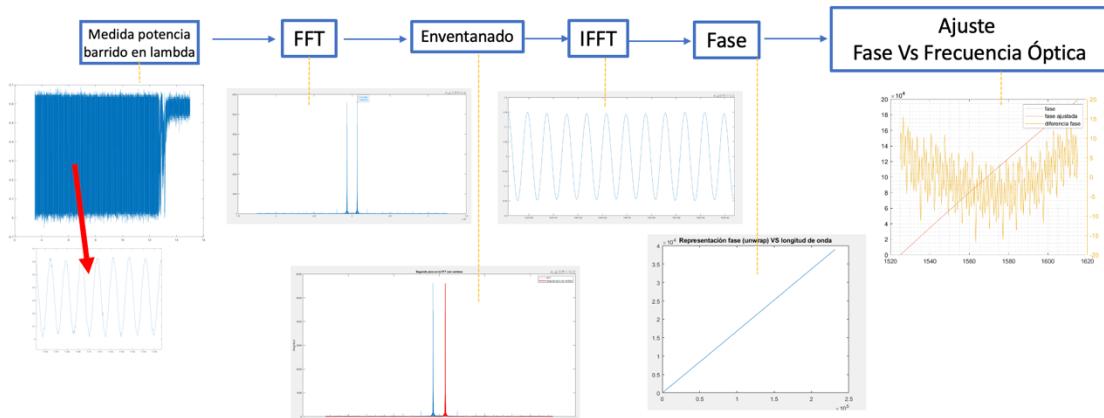
### 5.1 Corrección de la longitud de onda

Para la corrección de la longitud de onda se utiliza la rama del montaje Bloque 1 de la **Figura 7**, esto es, en un interferómetro de fibra óptica con una diferencia de caminos de 2 metros de fibra:



**Figura 8:** Esquema del Bloque 1 para corrección de lambda

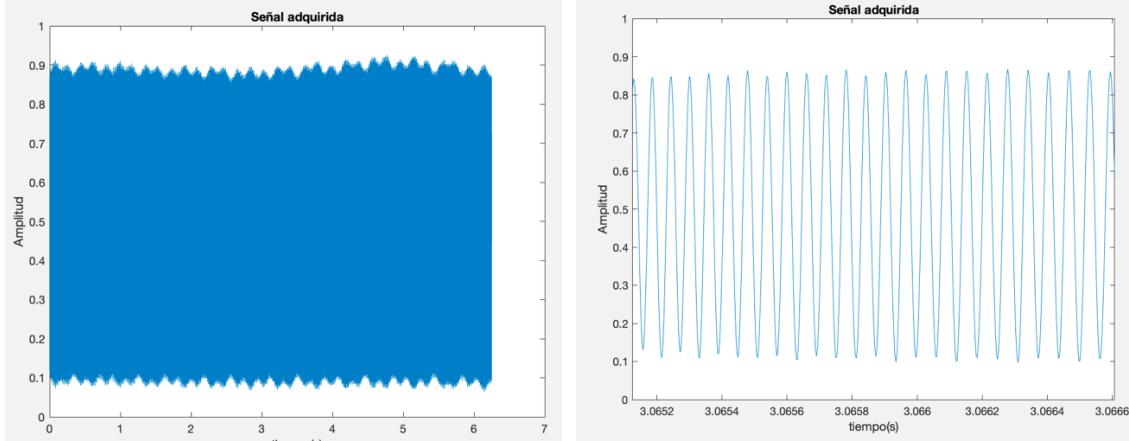
En el proceso de corrección de longitud de onda se siguen los pasos descritos en el siguiente esquema:



**Figura 9:** Esquema del Bloque 1 para corrección de lambda

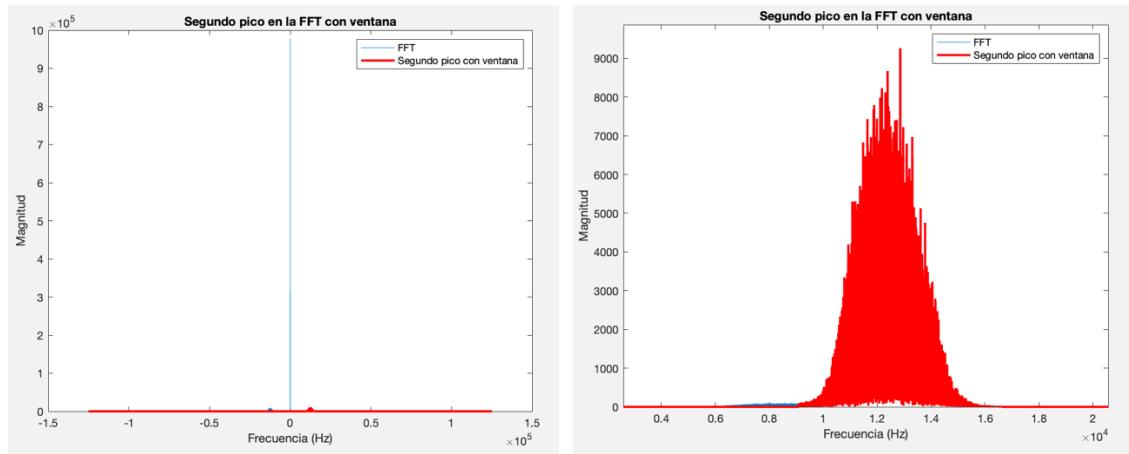
En primer lugar, se realiza la captura de los datos del interferograma *potencia\_medida* en un proceso de barrido que da como resultado la medida mostrada, a modo de ejemplo, en la **Figura 10**. En cada proceso de barrido, el *trigger* del equipo determina los tiempos inicial y final de captura de la *DAQ*, de forma que el número de datos capturados en el vector *potencia\_medida* viene determinado por la tasa de adquisición de la *DAQ* y por el tiempo de duración de dicha adquisición, a su vez, determinado por la velocidad de barrido y las longitudes de onda inicial y final del barrido. A cada uno de los valores registrados en el vector *potencia\_medida* se le asigna inicialmente una longitud de onda, *lambda\_asignada*, mediante la función *linspace*, de forma que se construye un *array* *[lambda\_asignada, potencia\_medida]* en el que *lambda\_asignada* contiene valores de

longitud de onda equiespaciados. Notar que esta asignación solo sería correcta en el caso que el barrido del *TLS* fuese perfectamente lineal en el tiempo, y es por ello que debe realizarse una corrección a esta asignación.



**Figura 10:** A la izquierda, señal interferente completa, a la derecha, zoom de la señal interferente

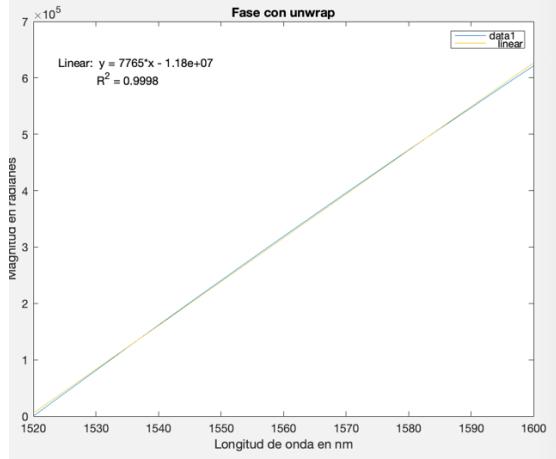
Dado que el *array* de datos contiene errores en cada una de las longitudes de onda asignadas, la cuantificación de estos se realiza en términos de la fase del interferograma para cada longitud de onda. Por ello, mediante el proceso descrito en el **apartado 3.2**, se calcula la *FFT* del interferograma, comprobando que se obtienen espectros estrechos como el que se muestra en la **Figura 11**, idealmente las *deltas de Dirac* obtenidas en la **ec 3.11**. Tras enventanar la señal, se aplica la *IFFT* y se realiza el cálculo de la fase, obteniendo un *array* [*lambda\_asignada* , *fase\_medida*].



**Figura 11:** A la izquierda, en azul la *FFT* del interferograma y en rojo la misma señal enventanada. A la derecha, detalle en zoom de los espectros en la zona enventanada

Teniendo en cuenta que la variación de la fase del interferograma es inversamente lineal con la longitud de onda y directamente lineal con la frecuencia óptica tal y como se describía en la **ec 3.9**, resulta conveniente por sencillez trabajar en el espacio de frecuencias ópticas. Para ello, si  $\omega = 2\pi f_{opt}$ , mediante  $f_{opt} = \frac{2\pi c}{\lambda_s}$  se construye el *array* [*frec\_optica\_asignada* , *fase\_medida*]. En este contexto, la fase medida  $\varphi_m$  se procesa

con el comando *unwrap*, que elimina los saltos de  $2\pi$  y genera una representación continua de la fase. Posteriormente, se utiliza la herramienta *basic fitting* para ajustar la fase a una función lineal como se ilustra en la **Figura 12**:



**Figura 12:** Ajuste de la fase (azul) respecto a una función lineal (amarillo)

Notar que si el barrido del *TLS* fuese perfectamente lineal, los datos representados [*frec\_optica\_asignada* , *fase\_medida*] se ajustarían perfectamente a la tendencia de la línea recta, representada por la fase ajustada  $\varphi_o$ . Sin embargo, es evidente que la fase medida  $\varphi_m$  no presenta un comportamiento perfectamente lineal, lo que hace necesario aplicar un ajuste para linealizarla.

Esta fase ajustada  $\varphi_o$  se obtiene mediante el comando *polyfit*, que como se ha mostrado en la **Figura 12**, proporciona el vector lineal que describe el comportamiento esperado de la fase con respecto a la frecuencia óptica. Una vez calculada la fase ajustada  $\varphi_o$ , las frecuencias ópticas corregidas se convierten nuevamente a longitud de onda, permitiendo aplicar la corrección a la fase medida para calcular el vector de longitudes de onda corregidas  $\lambda_c$  según el algoritmo detallado a continuación:

Para  $\varphi_m(i) > \varphi_o(i)$ :

$$\lambda_c(i) = \lambda_s(i) + \frac{\lambda_s(i+1) - \lambda_s(i)}{\varphi_o(i+1) - \varphi_o(i)} \cdot (\varphi_m(i) - \varphi_o(i))$$

Para  $\varphi_m(i) < \varphi_o(i)$ :

$$\lambda_c(i) = \lambda_s(i) - \frac{\lambda_s(i) - \lambda_s(i-1)}{\varphi_o(i) - \varphi_o(i-1)} \cdot (\varphi_o(i) - \varphi_m(i))$$

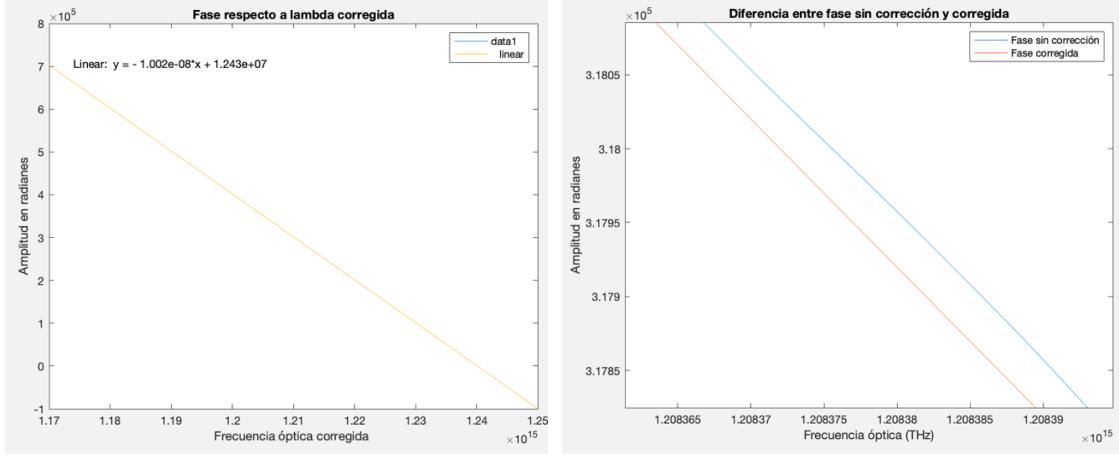
Para  $\varphi_m(i) = \varphi_o(i)$ :

$$\lambda_c(i) = \lambda_s(i)$$

donde  $\lambda_c$  como la longitud de onda corregida y  $\lambda_s$  como la longitud de onda original.

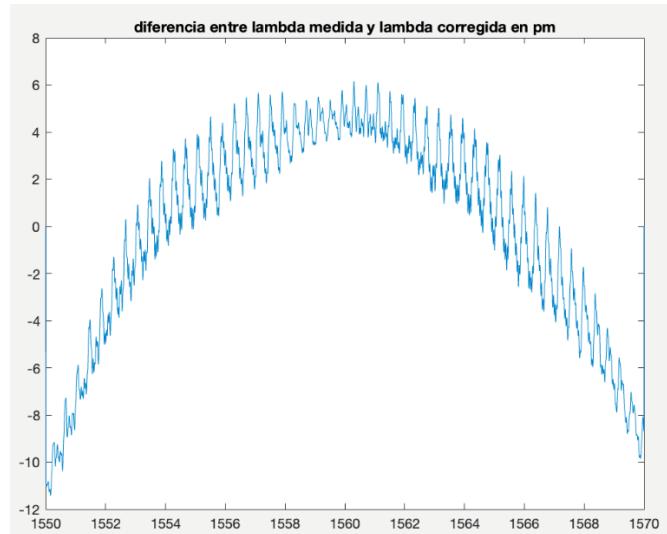
Este método de ajuste se basa en una interpolación lineal entre los valores de  $\lambda$  y la fase, donde  $\varphi_m(i) > \varphi_o(i)$  indica que el *TLS* estaba adelantado en longitud de onda con respecto a la situación de barrido lineal, y  $\varphi_m(i) < \varphi_o(i)$  que estaba retrasado. Por esta razón, la longitud de onda se corrige de manera proporcional a la diferencia de fases entre la medida y el ajuste. De esta forma se corrigen las longitudes de onda

medidas a las longitudes de onda corregidas para obtener el array [*lambda\_corregida*, *potencia\_medida*]. Este vector *lambda\_corregida* es el resultado final del proceso de corrección de *lambda*, y se usará para asignar las longitudes de onda corregidas a las medidas obtenidas del *DUT* en cada uno de los barridos



**Figura 13:** A la izquierda, resultado de la corrección de la fase (azul) y tendencia lineal teórica (amarillo). A la derecha, detalle con un zoom de las fases corregidas y sin corregir.

En la **Figura 14** se muestra, a modo de ejemplo, el resultado final de un proceso de corrección de longitud de onda en un rango de barrido de 1550 nm a 1570 nm a 5nm/s de velocidad de barrido, indicando las diferencias obtenidas entre la longitud de onda medida y la longitud de onda corregida.



**Figura 14:** Diferencia entre *lambda medida* y *corregida* en picómetros

En la curva de la figura se observa que el rango de la corrección oscila entre -10 pm y 6 pm, lo que destaca la alta precisión del método implementado. Además, esta gráfica evidencia la tendencia del comportamiento intrínseco de autocorrección del *TLS* a lo largo del barrido, alternando entre fases de aceleración y desaceleración. Estas variaciones son corregidas de manera efectiva con nuestro método, lo que quedará

demonstrado en los resultados obtenidos al aplicar la corrección de longitud de onda en los procesos de medida de retardo de grupo y de *OFDR* mostrados en el **Capítulo 6** de la memoria.

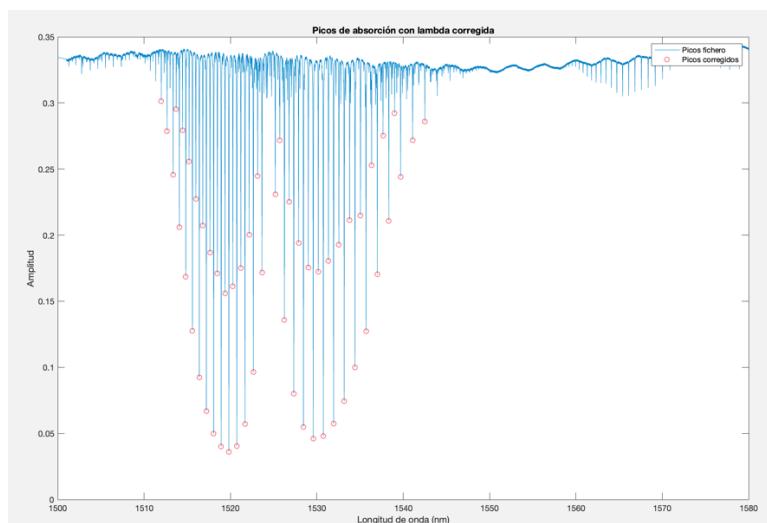
## 5.2 Calibración de longitud de onda mediante celda de gas

Siguiendo el esquema del Bloque 3, detallado en la **Figura 15**, se implementa la medida directa en transmisión de una celda de gas, lo que permite realizar la corrección absoluta de la longitud de onda del *TLS*.



**Figura 15:** Esquema del Bloque 3, para calibración de *lambda* absoluta

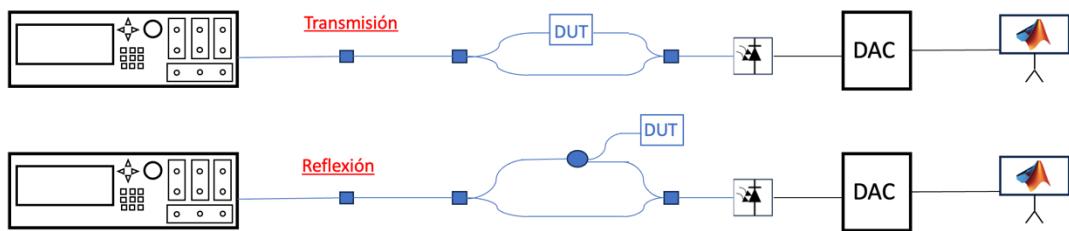
A diferencia de la corrección relativa descrita previamente, esta metodología se centra en un enfoque basado en referencias absolutas, utilizando los picos de absorción conocidos bien definidos de una mezcla de gases que en este caso son de cuatro distintos:  $H_2O + ^{12}C_2H_2 + ^{12}CO + ^{13}CO$ , los cuales presentan líneas de absorción características que permiten alinear de forma precisa la longitud de onda a valores absolutos. Si se realiza una medida de la potencia de la señal transmitida a través de la celda de gas durante un barrido del *TLS*, es posible ajustar la corrección de longitud de onda relativa, realizada previamente en el Bloque 1, a una calibración en longitud de onda absoluta. La calibración obtenida se muestra en la **Figura 16**, donde los picos de absorción están representados por la línea azul y los puntos de referencia de longitud de onda absoluta adquiridos están marcados con círculos rojos. Los valores reales de los picos de absorción están definidos en el fichero de calibración proporcionado por el fabricante de la celda de gas.



**Figura 16:** En azul, picos de absorción del fichero del fabricante, en rojo, los valores corregidos del barrido realizado

### 5.3 Medida del DUT

La medición en el *DUT* (*Device Under Test*) se lleva a cabo en el Bloque 2, como se ilustra en la **Figura 7**. Este bloque implementa un interferómetro, y la disposición del *DUT* depende del parámetro a medir. Para las mediciones de retardo de grupo, el *DUT* se posiciona en configuración de transmisión, permitiendo que la señal óptica lo atravesie. Por otro lado, para las mediciones mediante *OFDR*, el *DUT* se coloca en configuración de reflexión, utilizando un circulador para redirigir la señal reflejada hacia el detector. Esto permite analizar con precisión las propiedades de reflexión del dispositivo.



**Figura 17:** Montaje en transmisión y reflexión para la medida del DUT

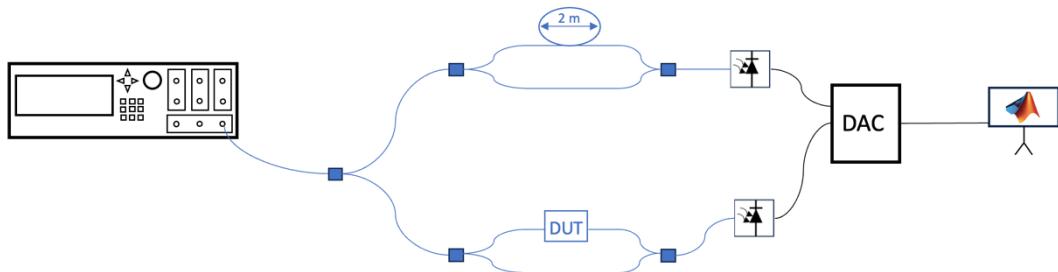
Los resultados de la caracterización de diferentes *DUT* se muestran en el **Capítulo 6**.

## 6. Medidas

En este capítulo se describen en detalle las medidas realizadas de retardo de grupo y de OFDR sobre distintos *DUT*. En cada una de las medidas realizadas, se realiza la corrección de la longitud de onda relativa y absoluta descritas en el **apartado 5.1 y 5.2**. Para cada medida realizada, se muestra el impacto que tiene la corrección de longitud de onda en la calidad y precisión de las medidas.

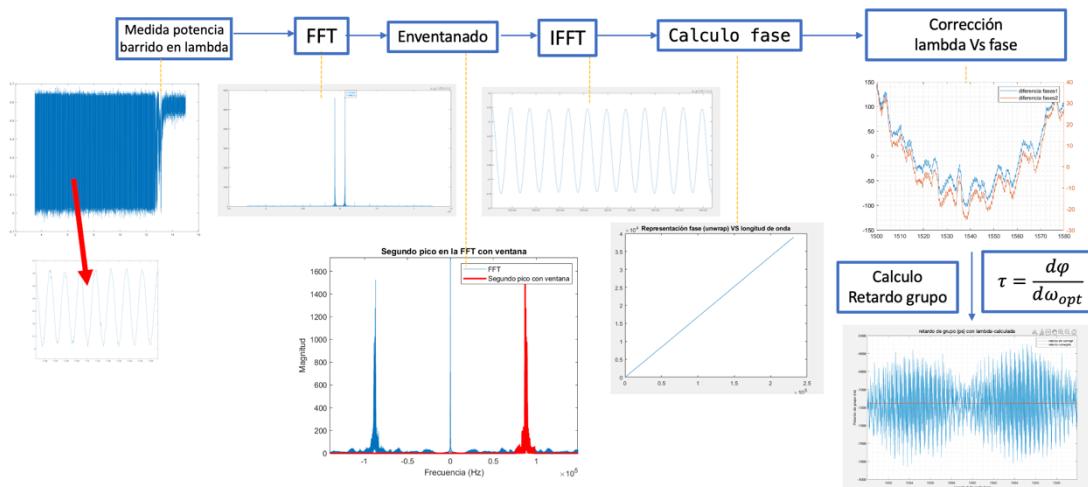
### 6.1 Retardo de grupo

El montaje que se utiliza para la medida es el siguiente:



**Figura 18:** Sistema de montaje para medida del retardo de grupo

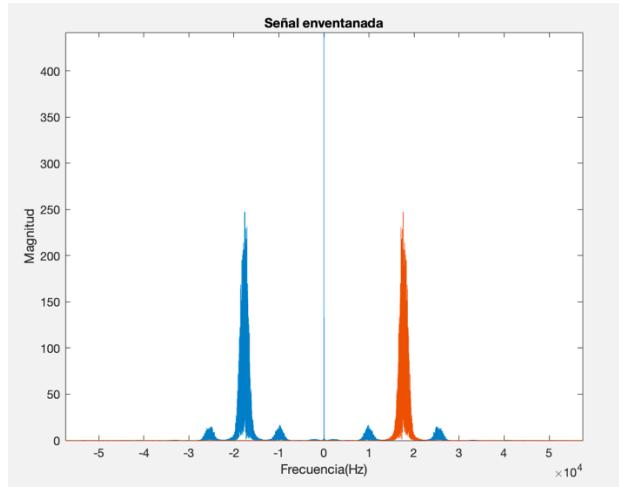
Y el esquema del proceso de medida que se sigue para aplicar la corrección en longitud de onda y la obtención del retardo de grupo corregido es el siguiente:



**Figura 19:** Esquema del proceso de corrección de la medida de retardo de grupo

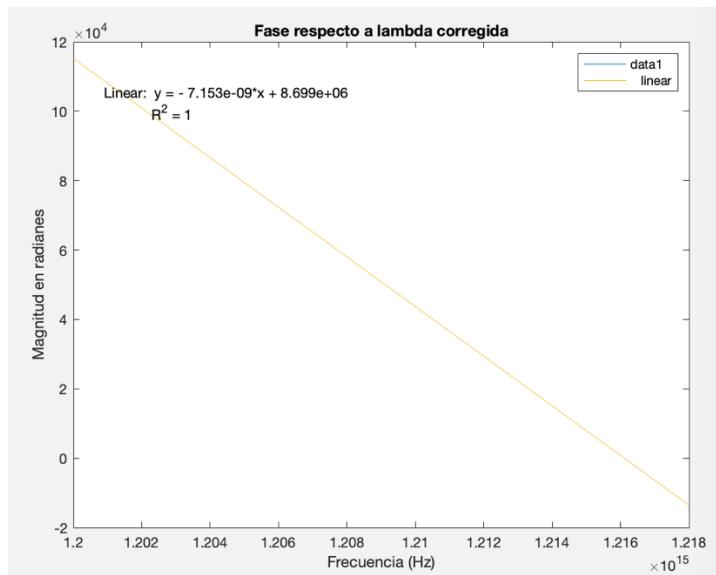
Para la validación del esquema y del proceso de medida de retardo de grupo, se utiliza como *DUT* una fibra óptica de unos pocos metros. Este es debido a que el comportamiento de la fibra óptica es bien conocido, y el valor de retardo de grupo resultante debe ser un valor prácticamente constante con la longitud de onda, con una pendiente muy pequeña, por debajo de la resolución de medida, debido a la dispersión cromática de la fibra.

Siguiendo el esquema del proceso de medida, por un lado, las medidas obtenidas en los bloques 1 y 3 son utilizadas para obtener el vector de longitudes de onda corregido para cada barrido implementado o medida realizada. Al mismo tiempo, en el bloque 2 se realiza la medida del *DUT*. El interferograma adquirido en el bloque 2, en el que el *DUT* se coloca en transmisión en uno de los brazos del interferómetro, es procesado según el esquema de la **Figura 19**. Se aplica la *FFT*, se enventana la parte positiva del espectro (**Figura 20**), se aplica la *IFFT* y se obtiene la fase a cada longitud de onda.



**Figura 20:** Señal enventanada del segundo camino

Dado que ya se ha corregido el vector de longitudes de onda mediante el método explicado en el **apartado 5.1**, esta corrección se aplica directamente a la representación de la fase medida, resultando en una curva de fase que es prácticamente lineal con la frecuencia óptica, **Figura 21**, tal y como se espera teóricamente.

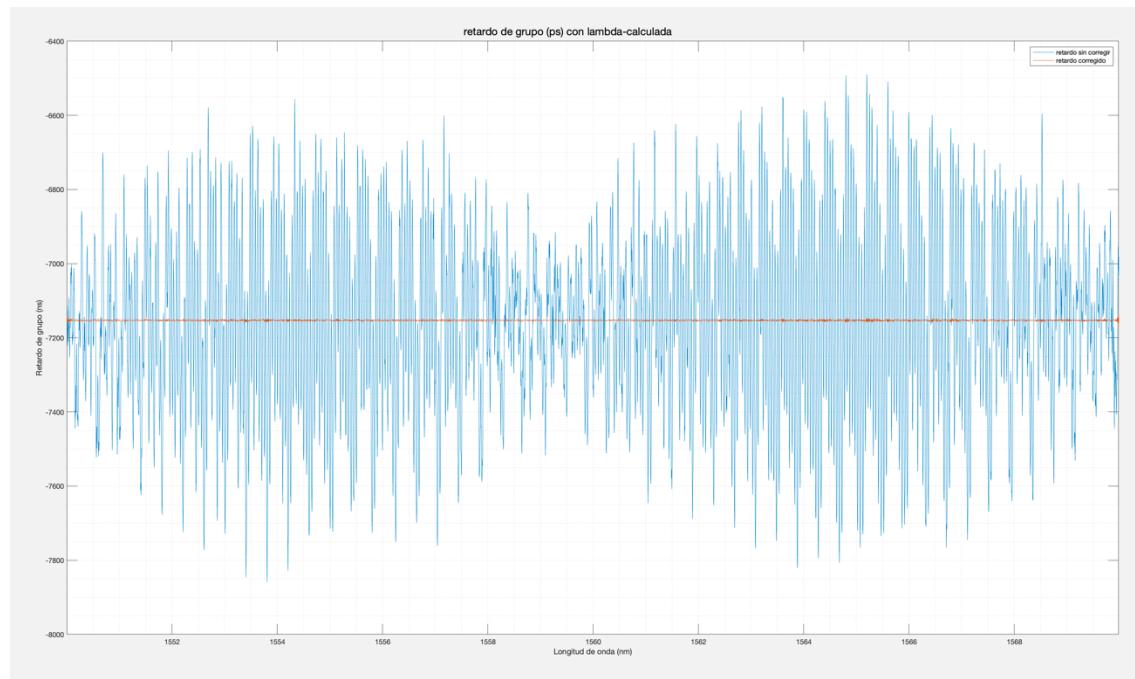


**Figura 21:** Fase respecto a lambda corregida

Se puede observar en la imagen que la ecuación de ajuste lineal presenta un valor muy bajo en el término de la pendiente, reflejando la alta precisión alcanzada en la corrección

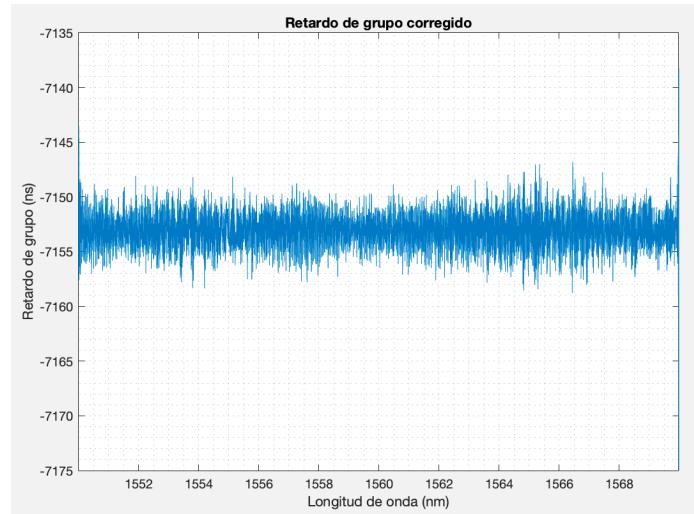
de la longitud de onda. Además, el coeficiente  $R^2 = 1$  confirma que el ajuste es perfecto, validando así la fiabilidad del método aplicado para los datos analizados.

A partir de estos valores de fase se calcula el retardo de grupo con la **ec 3.15**. Para apreciar la mejora en la precisión de la medida que se obtiene al aplicar la corrección en longitud de onda, en la siguiente gráfica se muestran los resultados obtenidos en la medida de retardo de grupo sin aplicar la corrección y aplicando la corrección.



**Figura 22:** Diferencia entre retardo de grupo sin corregir (azul) y corregido (naranja)

En la **Figura 23** se muestra en detalle ampliado los valores obtenidos de retardo de grupo con corrección de longitud de onda del barrido.



**Figura 23:** Retardo de grupo corregido

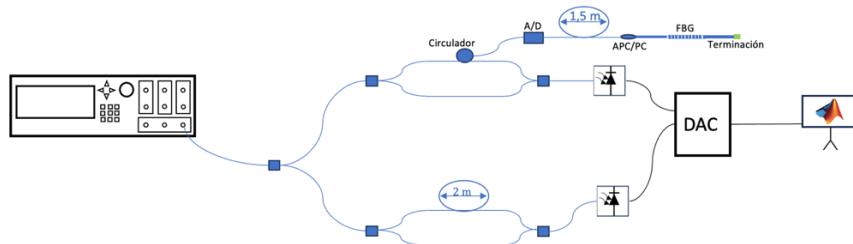
Es evidente que la precisión de la medida mejora en prácticamente dos órdenes de magnitud (desviación estándar 238.24, donde la variación pico a pico es de 1368.98 ns para los valores de retardo sin corregir, y 9.85 con una variación pico a pico de 14.14 ns

para los valores corregidos). Estos resultados demuestran que el método implementado elimina los errores sistemáticos del barrido, garantizando resultados más consistentes y fiables.

## 6.2 OFDR

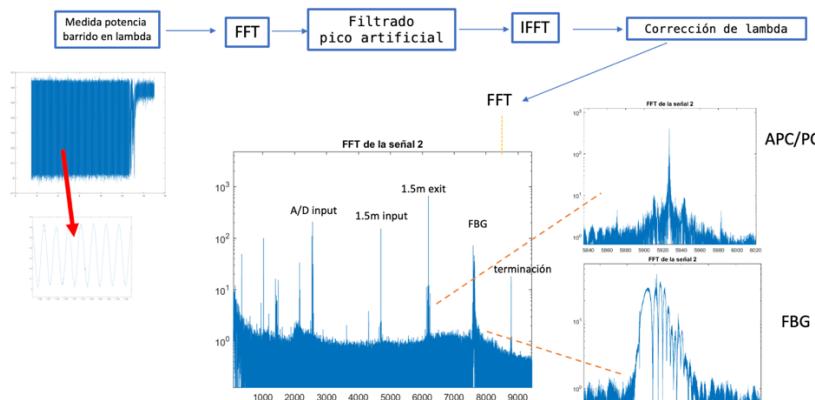
Como se ha explicado previamente, las reflexiones dentro del *DUT* generan múltiples frentes de onda con diferentes frecuencias de batido asociadas a las distancias ópticas de cada reflexión. Una correcta linealización de la medida del *TLS* asegura que la *Transformada de Fourier* del interferograma obtenido proporcione un espectro preciso, donde cada frecuencia de batido pueda asociarse directamente a una reflexión específica.

El montaje que se utiliza en la medida de *OFDR* del *DUT* es:



**Figura 24:** Sistema de montaje para la medida de OFDR

siendo el esquema del método de medida implementado el mostrado en la **Figura 25**:

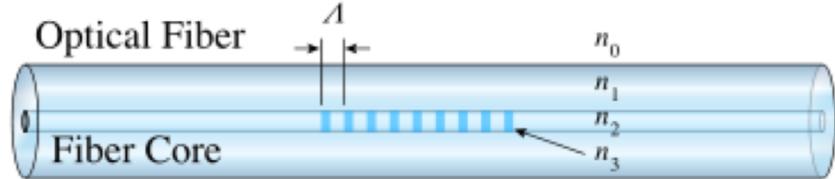


**Figura 25:** Esquema del proceso de corrección de la medida de OFDR

En esta ocasión, el montaje incluye un circulador óptico para posibilitar redirigir la señal reflejada por el *DUT* hacia el detector.

Para mostrar la validez del método de corrección en longitud de onda en la técnica *OFDR*, en esta ocasión el *DUT* está formado por un acoplador divisor (*A/D*) al cual se conecta una fibra de 1,5 metros de largo, terminada en un conector *APC/PC*, al que se le acopla un filtro en fibra *FBG* (*Fiber Bragg Grating*). Una *FBG* es un dispositivo que utiliza una fibra óptica con un patrón de variación periódica del índice de refracción

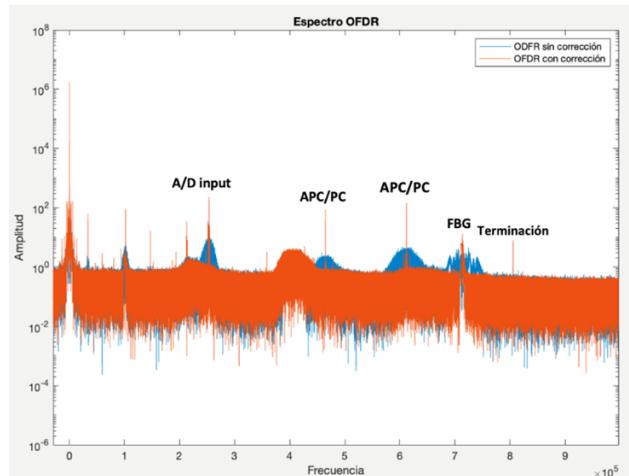
grabado en el núcleo. Este patrón actúa como un filtro que refleja selectivamente una longitud de onda (patrón periódico) o un conjunto de longitudes de onda específicas (patrón no periódico), mientras que el resto de longitudes de onda se transmiten [9].



**Figura 26:** Sistema de montaje para la medida de OFDR

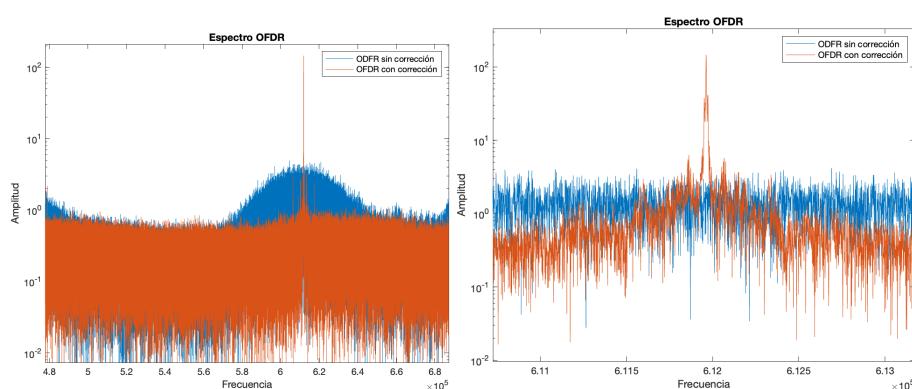
Siguiendo el proceso de medida, se realiza con el *TLS* un barrido en longitud de onda, se captura el interferograma en el bloque 2 para *DUT* en reflexión y se aplica la *FFT*. En el proceso, se corrige la longitud de onda del barrido según el bloque 1.

Para mostrar el impacto de la corrección en longitud de onda en la medida de *OFDR*, en la siguiente imagen se comparan las curvas *OFDR* obtenidas sin corrección de longitud de onda (azul) y con corrección de longitud de onda (naranja).



**Figura 27:** En azul, espectro OFDR sin corrección, en naranja, espectro OFDR con corrección

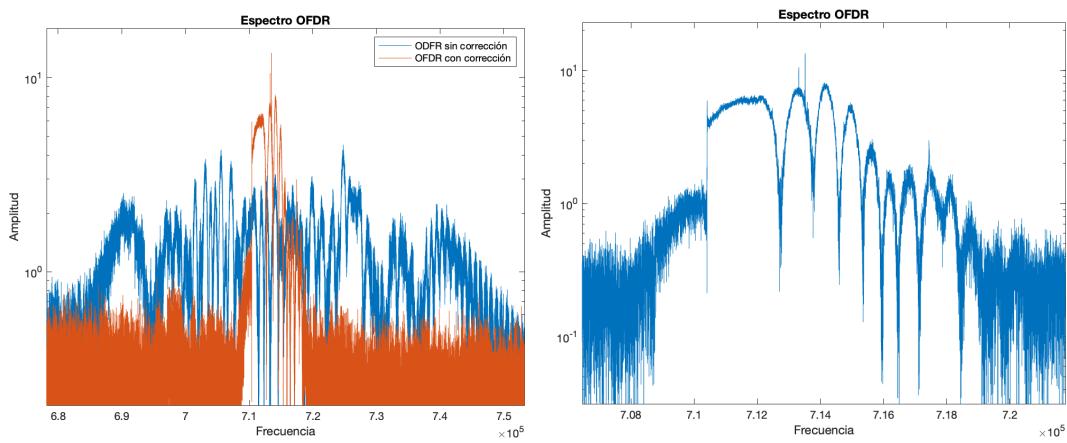
En las siguientes imágenes se pueden apreciar los componentes mencionados antes, como en la **Figura 28**, que se puede observar los espectros asociados a los conectores que unen la fibra de 1,5 metros, tanto al acoplador-divisor como al conector APC/PC:



**Figura 28:** A la izquierda, espectro del conector APC/PC sin corrección (azul) y con corrección(naranja). A la derecha, detalle ampliado

Se observa que, en el caso sin corrección, el espectro es significativamente más ancho y ruidoso, dificultando la distinción de los puntos donde se encuentran los conectores. En cambio, con la corrección aplicada, se logra detectar claramente los picos correspondientes al inicio y final de la fibra, lo que permite identificar con precisión las posiciones de los conectores.

Por otro lado, en la **Figura 29** se muestra el comportamiento de la *FBG*. Sin corrección, el espectro aparece muy ensanchado y prácticamente indistinguible, sin embargo, al aplicar la corrección, se define con claridad el espectro de la *FBG*, evidenciando la eficacia del método.



Finalmente, el espectro correspondiente al pico de terminación de la fibra también destaca la importancia de la corrección, dado que, sin esta, este directamente no aparecería en el espectro, mientras que, con la corrección, se puede identificar con precisión el final del sistema:

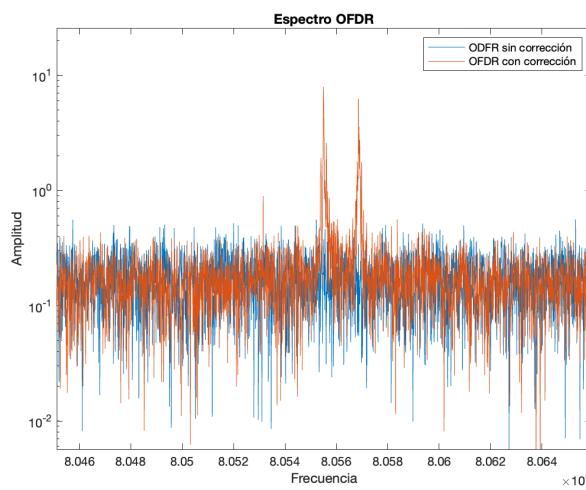
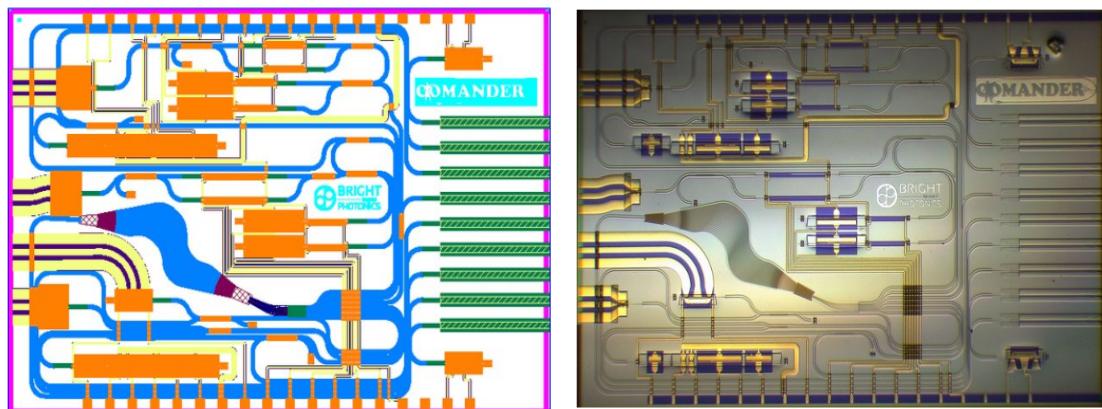


Figura 30: Espectro de la terminación sin corrección (azul) y con corrección(naranja).

## 6.3 Medidas en chip

La óptica integrada representa una de las tecnologías más innovadoras en el campo de la fotónica, pues permite la integración de múltiples dispositivos ópticos en un único chip ofreciendo sistemas más compactos y funcionales.

Esta tecnología posibilita la implementación de elementos fotónicos como interferómetros, guías de onda, *PBS* (*Polarizing Beam Splitter*), moduladores, *AWG* (*Arrayed Waveguide Grating*), filtros y acopladores en un mismo *PIC* (*Photonic Integrated Circuit*).



**Figura 31:** A la izquierda, diseño de un *PIC* en Nazca para mandar al foundry, a la derecha, versión final fabricada en chip

En cuanto a la tecnología de fabricación, existen múltiples materiales de desarrollo como el fosfuro de indio (*InP*), que además permite integrar fuentes de luz como láseres directamente en el chip. Sin embargo, la tendencia actual se dirige hacia la integración basada en silicio (*Silicon Photonics*), gracias a su compatibilidad con los procesos de fabricación de semiconductores, reduciendo significativamente los costes y ofreciendo ventajas en términos de velocidad y densidad de integración.

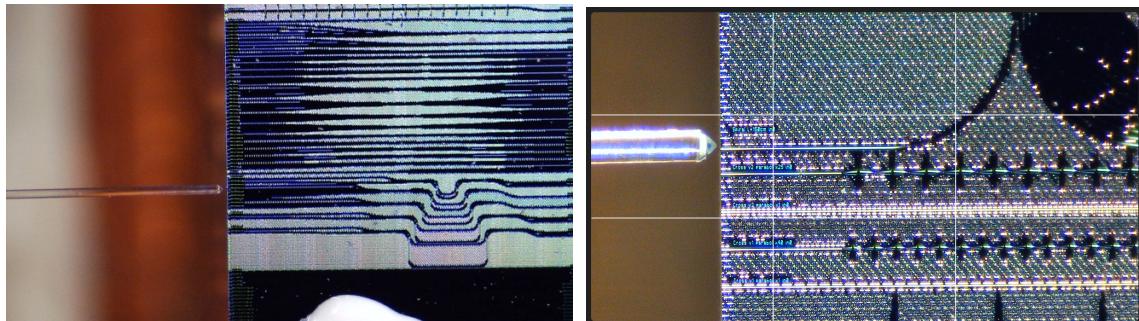
Por ello, para realizar una implementación del sistema de *OFDR* en un contexto de tecnología actualizada, se ha probado la realización de una medida en un *PIC* fabricado en *InP*.



**Figura 32:** Imagen del sistema de montaje de caracterización del chip

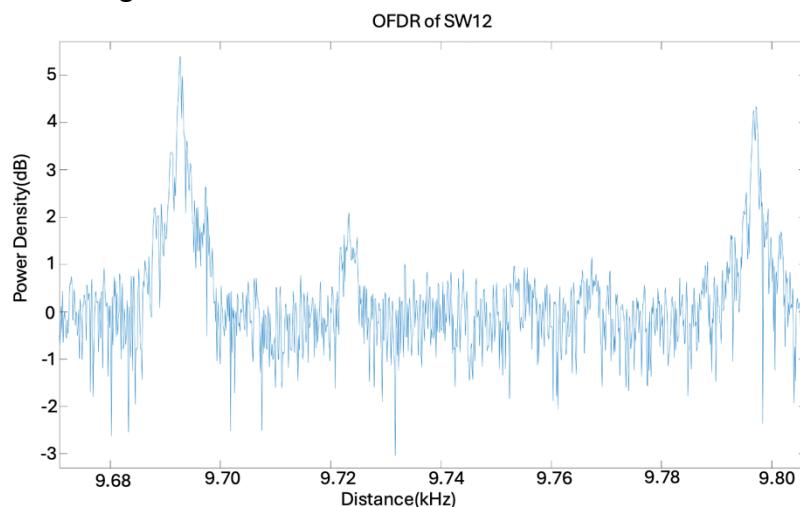
El paso más crítico para realizar esta medida en chip es introducir la luz en la guía del chip, dado que es el principal foco de pérdidas de potencia.

Para ello se hace uso de posicionadores de 6 ejes (donde los 3 primeros son X,Y,Z y los otros 3 corresponden alabeo, cabeceo y guiñada). Además, se hace uso de un microscopio para poder alinear manualmente la fibra con el chip.



**Figura 33:** Imagen de microscopio que muestra el alineamiento de la fibra con la guía del chip

Para la prueba de medida de *OFDR*, se ha implementado la corrección de longitud de onda utilizando un interferómetro externo. Posteriormente, se ha aplicado el proceso de medida de *OFDR* explicada con anterioridad a una de las partes del chip, específicamente una guía recta.



**Figura 34:** Medida de *OFDR* en una guía recta del chip

Se observa claramente en reflexión los picos correspondientes a la entrada y la salida de la guía. Estos picos representan los extremos de la guía recta y su separación, medida en el dominio frecuencial, corresponde a la longitud física de la guía.

## 7. Conclusiones y líneas futuras

En este trabajo fin de máster se ha propuesto y validado un sistema de linealización en longitud de onda del barrido de un láser *TLS*. El sistema, basado en la linealización de la fase medida en un interferograma de fibra óptica, utiliza un proceso que combina la *transformada de Fourier* y el filtrado de señales para la obtención de la fase, junto con un algoritmo de corrección de longitud de onda para garantizar la linealización de las medidas obtenidas con la frecuencia óptica del *TLS*. Este proceso ha sido aplicado y validado en dos medidas diferentes que requieren alta precisión en longitud de onda: la medida de retardo de grupo en componentes ópticos (utilizando unos metros de fibra óptica por simplicidad) y la medida de reflectometría mediante la técnica de *OFDR* (realizada sobre conectores ópticos y un filtro de *red de Bragg*). Los resultados obtenidos demuestran la eficacia y validez del sistema y del método implementado.

Este trabajo ha implicado el desarrollo teórico de conceptos clave como la interferencia, la obtención de la fase, el retardo de grupo y el *OFDR*, evidenciando la concordancia entre los datos experimentales y los fundamentos teóricos. Además, ha requerido un profundo conocimiento de la interferometría en fibra óptica, así como de métodos de procesado de señales. También ha sido necesario diseñar y programar un sistema completo de control, adquisición y procesamiento de datos, junto con la implementación de herramientas para el manejo y control del equipamiento óptico de laboratorio.

El uso de una celda de gas como complemento ha permitido realizar una corrección y linealización precisa del vector de longitudes de onda, alcanzando una precisión superior a un picómetro. Esta corrección se ha implementado con éxito en diversas medidas, verificando su eficacia tanto en el análisis del retardo de grupo como en la técnica *OFDR*. Finalmente, se ha realizado una medida de *OFDR* en un chip de óptica integrada, validando el correcto funcionamiento del sistema en un dispositivo fotónico avanzado.

Como líneas futuras, se pueden nombrar las siguientes:

- Mejorar la adquisición de datos, tanto en el incremento de la tasa de la *DAQ* como la implementación de un filtro antialiasing a la entrada, permitiendo así una mayor precisión en la medida.
- Optimizar la longitud del latiguillo del interferómetro de corrección de longitud de onda, ajustándola a cada tipo de dispositivo a caracterizar en número de batidos de la interferencia, permitiendo también mayor precisión y resolución en la medida.
- Llevar a cabo una campaña de medidas destinada a la caracterización de chips de óptica integrada.
- Estudiar el origen de los artefactos que aparecen en ciertas medidas *OFDR* para poder eliminarlos adecuadamente.

## 8. Anexos

### Anexo 1. Velocidad de fase y velocidad de grupo

Para demostrar la relación  $v_g = \frac{\partial \omega}{\partial \tau}$  lo haremos de la forma más general posible [14]:

$$u(x, t) = \int a(k) e^{i(\omega k - kx)} dk$$

donde  $\omega = \omega(k)$  se debe considerar como una función del número de onda  $k$ .

Ahora asumiremos que el grupo es muy estrecho, de modo que en la integral solo existirán aquellas ondas de amplitud finita cuyos números de onda difieren del número de onda promedio  $k_0$  por una cantidad muy pequeña. Si ponemos  $k = k_0 + k_1$ ,  $\omega(k) = \omega_0 + \omega_1(k)$  y  $a(k_0 + k_1) = b(k_1)$ , el grupo de ondas puede escribirse como:

$$u(x, t) = A(x, t) e^{i(\omega_0 t - k_0 x)}$$

donde

$$A(x, t) = \int a(k_0 + k_1) e^{i(\omega_1 t - k_1 x)} dk_1 = \int b(k_1) e^{i(\omega_1 t - k_1 x)} dk_1$$

Y por tanto:

$$\begin{aligned} u(x, t) &= \int a(k_0 + k_1) e^{i(\omega_1 t - k_1 x)} e^{i(\omega_0 t - k_0 x)} dk_1 \\ &= \int a(k_0 + k_1) e^{i((\omega_0 + \omega_1)t - (k_0 + k_1)x)} dk_1 \end{aligned}$$

Por lo tanto, en el grupo de ondas puede considerarse como una onda única de frecuencia  $\omega_0$ , número de ondas  $k_0$  y amplitud  $A(x, t)$ , que varía de un punto a otro. Esto es válido ya que hemos supuesto que  $A(x, t)$  varía lentamente en comparación con  $e^{i(\omega_0 t - k_0 x)}$ . En primera aproximación, la variación debida al batido de  $\omega_1$  es muy lenta con respecto a  $\omega_0$ .

La velocidad con la que una amplitud definida  $A(x, t)$ , por ejemplo, en su máximo, avanza con la onda es lo que sería nuestra velocidad de grupo  $v_g$ .

La derivada total de esta amplitud varía por tanto en función de  $x$  y  $t$ .

Si la amplitud fuera constante, es decir, sin variación en el máximo de la envolvente, su derivada sería 0, por ello:

$$\frac{\partial A}{\partial x} \frac{dx}{dt} + \frac{\partial A}{\partial t} = 0$$

Donde  $\frac{\partial A}{\partial x}$  es la variación espacial,  $\frac{\partial A}{\partial t}$  la variación temporal y  $\frac{dA}{dt}$  la velocidad de grupo:

$$v_g = \left( \frac{dx}{dt} \right)_{A=cte} = - \frac{\frac{\partial A}{\partial t}}{\frac{\partial A}{\partial x}}$$

Si ahora derivamos:

$$\frac{\partial A}{\partial t} = i\omega_1 \int b(k_1) e^{i(\omega_1 t - k_1 x)} dk_1$$

$$\frac{\partial A}{\partial x} = -ik_1 \int b(k_1) e^{i(\omega_1 t - k_1 x)} dk_1$$

Como hemos supuesto que el grupo está confinado a un rango muy estrecho de longitudes de onda, podemos expandir por Taylor  $\omega_1(k_1)$  en potencias de  $k_1$ :

$$\omega_1(k_1) = \omega(k) - \omega_0 = \left( \frac{d\omega}{dk} \right)_0 k_1 + \dots$$

Por ello:

$$\frac{\partial A}{\partial t} = -\frac{\partial A}{\partial x} \frac{dx}{dt} = -\frac{\partial A}{\partial x} \left( \frac{d\omega}{dk} \right)_0$$

Teniendo por tanto que la velocidad de grupo es  $v_g = \frac{d\omega}{dk}$

Para comprobarlo:

$$v_g = -\frac{\frac{\partial A}{\partial t}}{\frac{\partial A}{\partial x}} = -\frac{i\omega_1 \int b(k_1) e^{i(\omega_1 t - k_1 x)} dk_1}{-ik_1 \int b(k_1) e^{i(\omega_1 t - k_1 x)} dk_1} = \frac{\omega_1}{k_1}$$

$$v_g = \frac{\omega_1}{k_1} = \frac{\left( \frac{d\omega}{dk} \right)_0 k_1}{k_1} = \frac{d\omega}{dk}$$

Y la velocidad de fase:

$$v_f = \frac{\omega}{k}$$

## Anexo 2. Control SCPI TLS LabJack y DAC NI

```
function tls_and_labj(long_ini,long_fin,velocidad,scanRate,tiempo,n_barridos,num_channels)
%Programa para capturar datos del TLS con LabJack
% tls_and_labj(long_ini,long_fin,velocidad,scanRate,tiempo)
%Parámetros de entrada:
%    - long_ini: longitud de onda inicial en nm
%    - long_fin: longitud de onda final en nm
%    - velocidad: velocidad de barrido en nm/s
%    - scanRate: Tasa de muestreo en muestras/s (para 1 canal,
%                50000 muestras/s (máximo), para 2 canales, 25000 muestras/s, etc...)
%    - tiempo: calculado como la diferencia de longitudes de onda/velocidad
%              + 4 segundos de margen

numScans = tiempo*scanRate;
obj2 = instrfind('Type', 'gpib', 'BoardIndex', 7, 'PrimaryAddress', 5, 'Tag', '');
% Create the GPIB object if it does not exist
% otherwise use the object that was found.
if isempty(obj2)
    obj2 = gpib('AGILENT', 7, 5); %Laser sintonizable
else
    fclose(obj2);
    obj2 = obj2(1);
end
fopen(obj2);
%Despues de poner contraseña en laser sintonizable
data1=query(obj2,'outp0:chan1?');
valor=str2double(data1);

if valor==0
    fwrite(obj2,'outp0:chan1 1');
end

for i=1:n_barridos
    nombre_carpetasprintf('barrido%d',i);
    if ~exist(nombre_carpetasprintf('barrido%d',i));
        mkdir(nombre_carpetasprintf('barrido%d',i));
    end

[datos,trigger,data_ref,tiempo_vec,long_ini,long_fin]=tls_process_and_labjack(obj2,scanRate,tiempo,long_ini,long_fin,velocidad,num_channels);
    nombre_archivo1=fullfile(nombre_carpetasprintf('barrido%d',i),'data_captura%d.mat',i));
    if(num_channels==3)
        save(nombre_archivo1,"datos","trigger","data_ref","tiempo_vec","long_ini","long_fin");
    %descomentar para guardar matriz
    else
        save(nombre_archivo1,"datos","trigger","tiempo_vec","long_ini","long_fin");
    %descomentar para guardar matriz
    end
    pause(3)
end
end
```

```

function [datos,trigger,data_ref,tiempo_vec,long_ini,long_fin] =
tls_process_and_labjack(obj2,scanRate, tiempo,long_ini,long_fin,velocidad,num_channels)
%% Parámetros GPIB

% Esta función hace el barrido del TLS mientras la labjack captura
fwrite(obj2,['sour0:chan1:wav:swe:star ' num2str(long_ini) 'nm']);
fwrite(obj2,['sour0:chan1:wav:swe:stop ' num2str(long_fin) 'nm']);

fwrite(obj2,['sour0:chan1:wav:swe:spe ' num2str(velocidad) 'nm/s']);
fwrite(obj2,'sour0:chan1:wav:swe:cycl 1'); % ciclos

fwrite(obj2,'sense2:chan1:func:stat logg'); %registro de datos de medida
if(num_channels==3)
    fwrite(obj2,'sense4:chan1:func:stat logg'); %registro de datos de medida
end
fwrite(obj2,'sour0:chan1:wav:swe:mode cont'); %realiza un barrido continuo de longitudes de onda.
fwrite(obj2,'sour0:chan1:wav:swe:step 0.001nm'); %establece el paso de 1nm para el barrido continuo

fwrite(obj2,'sour0:chan1:wav:swe star'); % inicio del barrido

if(num_channels==3)
    [datos,trigger,data_ref,tiempo_vec]=labjack_process2_pruebamia(scanRate,tiempo,num_channels);
else
    [datos,trigger,tiempo_vec]=labjack_process(scanRate,tiempo);
    data_ref=0;
end
end

```

```

function [datos,trigger,data_ref,tiempo_vec] = labjack_process2_pruebamia(scanRate,
tiempo,num_channels)
%Este programa está hecho para 2 canales, AIN0 y AIN1 y AIN2
%tiempo, depende del barrido, poner por ejemplo 5 seg
%scanRate, para dos canales poner 25000kHz
% Make the UD .NET assembly visible in MATLAB.
ljasm = NET.addAssembly('LJUDDotNet');
ljudObj = LabJack.LabJackUD.LJUD;

i = 0;

k = 0;
ioType = 0;
channel = 0;
dblValue = 0;
dblCommBacklog = 0;
dblUDBacklog = 0;
%scanRate = 25000;% 25000
%tiempo=5;%segundos
numScans = tiempo*scanRate;
tiempo_vec=(0:numScans-1)*(1/scanRate);
%%
numScansRequested = 0;
loopAmount = 1; % Number of times to loop and read stream data
% Variables to satisfy certain method signatures
dummyInt = 0;
dummyDouble = 0;
dummyDoubleArray = [0];

try
    % Read and display the UD version.
    disp(['UD Driver Version = ' num2str(ljudObj.GetDriverVersion())])

    % Open the first found LabJack U3.
    [ljudObj, lchandle] = ljudObj.OpenLabJackS('LJ_dtU3', 'LJ_ctUSB', '0', true, 0);

    % Stop any previous stream.
    try
        ljudObj.eGet(lchandle, 'LJ_ioSTOP_STREAM', 0, 0, 0);
    catch
    end

    % Read and display the hardware version of this U3.
    [ljudObj, dblValue] = ljudObj.eGetSS(lchandle, 'LJ_ioGET_CONFIG', 'LJ_chHARDWARE_VERSION',
0, 0);
    disp(['U3 Hardware Version = ' num2str(dblValue)])

    % Read and display the firmware version of this U3.
    [ljudObj, dblValue] = ljudObj.eGetSS(lchandle, 'LJ_ioGET_CONFIG', 'LJ_chFIRMWARE_VERSION',
0, 0);
    disp(['U3 Firmware Version = ' num2str(dblValue)])

    % Start by using the pin_configuration_reset IOType so that all
    % pin assignments are in the factory default condition.
    ljudObj.ePutS(lchandle, 'LJ_ioPIN_CONFIGURATION_RESET', 0, 0, 0);

    % Configure FI00 and FI01 as analog, all else as digital. That means we
    % will start from channel 0 and update all 16 flexible bits. We will
    % pass a value of b00000000000001 or d3.
    % For the last parameter we are forcing the value to an int32 to ensure
    % MATLAB uses the proper function overload.
    %ljudObj.ePutS(lchandle, 'LJ_ioPUT_ANALOG_ENABLE_PORT', 0, 3, 16);
    ljudObj.ePutS(lchandle, 'LJ_ioPUT_ANALOG_ENABLE_PORT', 0, 7, int32(16));

    % Configure the stream:
    % Set tasa Rate
    ljudObj.AddRequestSS(lchandle, 'LJ_ioPUT_CONFIG', 'LJ_chSTREAM_SCAN_FREQUENCY', scanRate,
0, 0);

    % Give the driver a 5 second buffer (scanRate * 2 channels * 5 seconds).
    ljudObj.AddRequestSS(lchandle, 'LJ_ioPUT_CONFIG', 'LJ_chSTREAM_BUFFER_SIZE',
numScans*num_channels*5, 0, 0);

    % Configure reads to retrieve whatever data is available with waiting.
    LJ_swSLEEP = ljudObj.StringToConstant('LJ_swSLEEP');
    ljudObj.AddRequestSS(lchandle, 'LJ_ioPUT_CONFIG', 'LJ_chSTREAM_WAIT_MODE', LJ_swSLEEP, 0,
0);

    % Define the scan list as AIN0 then AIN1.
    ljudObj.AddRequestS(lchandle, 'LJ_ioCLEAR_STREAM_CHANNELS', 0, 0, 0, 0);

```

```

ljudObj.AddRequestS(ljhandle, 'LJ_ioADD_STREAM_CHANNEL', 0, 0, 0, 0); %channel 0, AIN0
ljudObj.AddRequestS(ljhandle, 'LJ_ioADD_STREAM_CHANNEL', 1, 0, 0, 0); %channel 1, AIN1
trigger
    ljudObj.AddRequestS(ljhandle, 'LJ_ioADD_STREAM_CHANNEL', 2, 0, 0, 0); %channel 2, AIN2
data_ref

% Execute the list of requests.
ljudObj.GoOne(ljhandle);

% Get all the results just to check for errors.
LJE_NO_MORE_DATA_AVAILABLE = ljudObj.StringToConstant('LJE_NO_MORE_DATA_AVAILABLE');
finished = false;
while finished == false
    try
        [ljerror, ioType, channel, dblValue, dummyInt, dummyDbl] =
ljudObj.GetNextResult(ljhandle, 0, 0, 0, 0);
    catch e
        if(isa(e, 'NET.NetException'))
            eNet = e.ExceptionObject;
            if(isa(eNet, 'LabJack.LabJackUD.LabJackUDException'))
                % If we get an error, report it. If the error is
                % LJE_NO_MORE_DATA_AVAILABLE we are done.
                if(int32(eNet.LJUDError) == LJE_NO_MORE_DATA_AVAILABLE)
                    finished = true;
                end
            end
        end
    end
    % Report non NO_MORE_DATA_AVAILABLE error.
    if(finished == false)
        throw(e)
    end
end
end

% Start the stream.
[ljerror, dblValue] = ljudObj.eGetS(ljhandle, 'LJ_ioSTART_STREAM', 0, 0, 0);

% The actual scan rate is dependent on how the desired scan rate divides
% into the LabJack clock. The actual scan rate is returned in the value
% parameter from the start stream command.
disp(['Actual Scan Rate = ' num2str(dblValue)])
disp(['Actual Sample Rate = ' num2str(num_channels*dblValue) newline]) % # channels *
scan rate

% Get the enums for LJ_ioGET_STREAM_DATA and LJ_chALL_CHANNELS which we use
% in the read stream data loop.
typeIO0 = ljasm.AssemblyHandle.GetType('LabJack.LabJackUD.LJUD+IO');
LJ_ioGET_STREAM_DATA = typeIO0.GetEnumValues.Get(22); % Use enum index for GET_STREAM_DATA
typeCHANNEL = ljasm.AssemblyHandle.GetType('LabJack.LabJackUD.LJUD+CHANNEL');
LJ_chALL_CHANNELS = typeCHANNEL.GetEnumValues.Get(99); % Use the enum index for
ALL_CHANNELS

% Read stream data
for i=1:loopAmount
    % Loop will run the number of times specified by loopAmount variable
    % Since we are using wait mode LJ_swSLEEP, the stream read waits for a
    % certain number of scans and control how fast the program loops.

    % Init array to store data.
    adblData = NET.createArray('System.Double', num_channels*numScans); % Max buffer size
(#channels*numScansRequested)
    %adblData = NET.createArray('System.Double', 104857600); % Max buffer size
(#channels*numScansRequested)

    % Read the data. The array we pass must be sized to hold enough SAMPLES,
    % and the Value we pass specifies the number of SCANS to read.
    numScansRequested = numScans;
    % Use eGetPtr when reading arrays in 64-bit MATLAB. Also compatible with
    % 32-bits.
    [ljerror, numScansRequested] = ljudObj.eGetPtr(ljhandle, LJ_ioGET_STREAM_DATA,
LJ_chALL_CHANNELS, numScansRequested, adblData);

    % Display the number of scans that were actually read.
    disp(['Iteration # = ' num2str(i)])
    disp(['Number read = ' num2str(numScansRequested)])

```

```

% Display the first scan.
disp(['First scan = ' num2str(adblData(1)) ', ' num2str(adblData(2))])

%mat_datos(1,i)=adblData(1);
%mat_datos(2,i)=adblData(2);

% Retrieve the current backlog. The UD driver retrieves stream data
% from the U3 in the background, but if the computer or code is too slow
% the driver might not be able to read the data as fast as the U3 is
% acquiring it, and thus there will be data left over in the U3 buffer.
[ljerror, dblCommBacklog] = ljudObj.eGetSS(ljhandle, 'LJ_ioGET_CONFIG',
'LJ_chSTREAM_BACKLOG_COMM', dblCommBacklog, dummyDoubleArray);
disp(['Comm Backlog = ' num2str(dblCommBacklog)])

[ljerror, dblUDBacklog] = ljudObj.eGetSS(ljhandle, 'LJ_ioGET_CONFIG',
'LJ_chSTREAM_BACKLOG_UD', dblUDBacklog, dummyDoubleArray);
disp(['UD Backlog = ' num2str(dblUDBacklog) newline])
end
datos=[];
trigger=[];
data_ref=[];
for i = 1:3:(numScansRequested*3)
    datos(end+1) = adblData(i);
    trigger(end+1) = adblData(i+1);
    data_ref(end+1) = adblData(i+2);
end

% Stop the stream
ljudObj.eGetS(ljhandle, 'LJ_ioSTOP_STREAM', 0, 0, 0);

disp('Done')
catch e
    %showErrorMessage(e)
    errorrr=e;
end
end

```

```

function [datos,trigger,data_ref,tiempo_vec,long_ini,long_fin] =
tls_process_and_NI(obj2,scanRate, tiempo,long_ini,long_fin,velocidad,num_channels)
%% Parámetros GPIB
% Esta función hace el barrido del TLS mientras la labjack captura
fwrite(obj2,['sour0:chan1:wav:swe:star ' num2str(long_ini) 'nm']);
fwrite(obj2,['sour0:chan1:wav:swe:stop ' num2str(long_fin) 'nm']);

fwrite(obj2,['sour0:chan1:wav:swe:spe ' num2str(velocidad) 'nm/s']);
fwrite(obj2,'sour0:chan1:wav:swe:cycl 1'); % ciclos

fwrite(obj2,'sense2:chan1:func:stat logg'); %registro de datos de medida
if(num_channels==3)
    fwrite(obj2,'sense4:chan1:func:stat logg'); %registro de datos de medida
end
fwrite(obj2,'so111111111111111111110ur0:chan1:wav:swe:mode cont'); %realiza un barrido continuo
de longitudes de onda.
fwrite(obj2,'sour0:chan1:wav:swe:step 0.001nm'); %establece el paso de 1nm para el barrido
continuo

fwrite(obj2,'sour0:chan1:wav:swe star'); % inicio del barrido

dq = daq("ni");
dq.Rate = scanRate;
ch1=addinput(dq, "Dev2", "ai0", "Voltage");
ch1.Range=[-2,2];
ch1.TerminalConfig='SingleEnded';

ch2=addinput(dq, "Dev2", "ai1", "Voltage");
ch2.Range=[-2,2];
ch2.TerminalConfig='SingleEnded';
t = addtrigger(dq,"Digital","StartTrigger","External","Dev2/PFI0")

data = read(dq, seconds(tiempo));
datos=data.Dev2_ai0;
hold on
data_ref=data.Dev2_ai1;
trigger=data.Dev2_ai1;

tiempo_vec_temp=0:1/scanRate:length(datos)/scanRate;
tiempo_vec=tiempo_vec_temp(1:end-1);
%%
stop(dq);

end

```

### Anexo 3. Corrección de longitud de onda

```
function [val_corte_fase]=ajuste_landa_CH()

close all;
dir_path='/Users/Jorge/Desktop/programas_TFM_casa/barrido';
full_path = [dir_path];%, num2str(i)];
dir_info = dir(full_path);
num_files = sum(~[dir_info.isdir]);
    val_corte_fase=1;

load('datos_fase.mat');

for barrido = 1:num_files
    fase_medida=fase_acumulada(:,barrido);
    lands_short=lands;
    frec_optica=(2*pi*3E17)./(lands_short);

    p=polyfit(frec_optica,fase_medida,1);
    fase_ajustada=polyval(p,frec_optica);
    dif_fase=fase_medida-fase_ajustada';

    lambda_corregida = zeros(size(fase_medida));

    if fase_medida(1) > fase_ajustada(1)
        lambda_corregida(1) = lands_short(1) + ((lands_short(2) - lands_short(1)) /
        (fase_ajustada(2) - fase_ajustada(1))) * (fase_medida(1) - fase_ajustada(1));
    elseif fase_medida(1) < fase_ajustada(1)
        lambda_corregida(1) = lands_short(1) - ((lands_short(1) - lands_short(2)) /
        (fase_ajustada(1) - fase_ajustada(2))) * (fase_ajustada(1) - fase_medida(1)); % Esto no tiene
        mucho sentido aquí, podrías ajustarlo si es necesario
    else
        lambda_corregida(1) = lands_short(1);
    end

    % Bucle principal: de 2 hasta el penúltimo elemento
    for i = 2:length(fase_ajustada)-1
        if fase_medida(i) > fase_ajustada(i)
            lambda_corregida(i) = lands_short(i) + ((lands_short(i+1) - lands_short(i)) /
            (fase_ajustada(i+1) - fase_ajustada(i))) * (fase_medida(i) - fase_ajustada(i));
        elseif fase_medida(i) < fase_ajustada(i)
            lambda_corregida(i) = lands_short(i) - ((lands_short(i) - lands_short(i-1)) /
            (fase_ajustada(i) - fase_ajustada(i-1))) * (fase_ajustada(i) - fase_medida(i));
        else
            lambda_corregida(i) = lands_short(i);
        end
    end

    % Manejo del último elemento (i = length(fase_ajustada))
    n = length(fase_ajustada);
    if fase_medida(n) > fase_ajustada(n)
        lambda_corregida(n) = lands_short(n) + ((lands_short(n) - lands_short(n-1)) /
        (fase_ajustada(n) - fase_ajustada(n-1))) * (fase_medida(n) - fase_ajustada(n));
    elseif fase_medida(n) < fase_ajustada(n)
        lambda_corregida(n) = lands_short(n) - ((lands_short(n) - lands_short(n-1)) /
        (fase_ajustada(n) - fase_ajustada(n-1))) * (fase_ajustada(n) - fase_medida(n));
    else
        lambda_corregida(n) = lands_short(n);
    end

    figure()
    plot(lambda_corregida(2:end),fase_medida(2:end));
    title('Fase respecto a lambda corregida');
    xlabel('lambda (nm) corregida ');
    ylabel('Amplitud en radianes');

    frec_optica_vieja=(2*pi*3E17)./(lands_short);
    frec_optica_nueva=(2*pi*3E17)./(lambda_corregida);
    figure()
    plot(frec_optica_vieja,fase_medida);
    title('Fase respecto a frecuencia óptica sin corregir');
    xlabel('Frecuencia óptica (THz)');
    ylabel('Amplitud en radianes');

    figure()
    plot(frec_optica_nueva,fase_medida);
    title('Fase respecto a frecuencia óptica corregida');
    xlabel('Frecuencia óptica');
```

```

    ylabel('Amplitud en radianes');

val_corte_fase=1;

lambda_corregida=lambda_corregida(val_corte_fase:end);
fase_medida=fase_medida(val_corte_fase:end);

frec_optica=(2*pi*3E17)./(lambda_corregida);
figure()
plot(frec_optica,fase_medida);
title('Fase respecto a lambda corregida');
xlabel('Frecuencia óptica corregida ');
ylabel('Amplitud en radianes');

figure()
plot(frec_optica,fase_medida);
hold on
plot(frec_optica,fase_ajustada);
yyaxis right
legend('fase medida','fase ajustada');
diferencia=lambda_corregida'-lands_short;

figure()
plot(lands_short,diferencia*10^3);
title('diferencia entre lambda medida y corregida');

res = diff(fase_medida)./ diff(frec_optica);

frec_optica_sincorr=(2*pi*3E17)./(lands_short);
res_sincorr = diff(fase_medida)./ diff(frec_optica_sincorr');

res(1)=res(2);
res(length(res))=res(length(res)-1);

figure()
plot(frec_optica_sincorr,fase_medida);
hold on
plot(frec_optica,fase_medida);
legend('Fase sin corrección','Fase corregida');
xlabel('Frecuencia óptica (THz)');
ylabel('Amplitud en radianes');
title('Diferencia entre fase sin corrección y corregida');

figure()
plot(lands_short(1:end-1), 1e12*res_sincorr);
hold on;
plot(lambda_corregida(1:end-1), 1e12*res);
grid minor
sgtitle('Representacion del retardo de grupo (ps) con lambda-calculada ')
ylabel('Retardo de grupo (ns)')
xlabel('Longitud de onda (nm)')
legend('retardo sin corregir','retardo corregido');
res1=1e12*res;
lambda_corregida(1:length(lambda_corregida),barrido)= lambda_corregida;
save("landas_calculadas_iter1",'lambda_corregida','lands_short');%, 'fase_medida','res');
end

end

```

## Anexo 4. Procesado medida retardo de grupo

```
% Programa general
clc
close all
clear all

addpath('C:\Users\usuario\Desktop\Laser_Sintonizable\captura_y_procesado\programas_main\programas_captura');%
addpath 'C:\Users\usuario\Desktop\Laser_Sintonizable\captura_y_procesado\programas_main'
long_ini=1520;%1500;
long_fin=1600;%1580;
velocidad=10;%40;
scanRate=20*16000;%25000; % Valor de tasa de scan, para 1 canal, 50000(máximo), 2 canales,
50000/2, etc..;
tiempo=(long_fin-long_ini)/velocidad;
n_barridos=1;
num_canales=3;
% tls_and_labj(long_ini,long_fin,velocidad,scanRate,tiempo,n_barridos,num_canales);
% tls_and_NI(long_ini,long_fin,velocidad,scanRate,tiempo,n_barridos,num_canales);

tlsSantec_and_NI(long_ini,long_fin,velocidad,scanRate,tiempo,n_barridos,num_canales);

%close all
% clearvars -except scanRate
%%
clc
close all
clear all

%scanRate=20*16000;
scanRate=250000
fs=scanRate;
num_canales=3;
[val_corte1]=captura_procesado1_CH(fs,num_canales)
%%
close all

clc
[val_corte_fase]=ajuste_landa_CH();

%%
clc
%close all
ajuste_y_procesado_camino_2(fs);
```

```

function [val_corte1]=captura_procesado1(fs,num_canales)
%Este programa se encarga de tomar los datos obtenidos y coger la señal
%El siguiente paso es cortar la señal (mirar visualmente el último valor
%en el que tenemos señal correcta
%para eso mirar última gráfica
close all;
clear all;

dir_path='/Users/Jorge/Desktop/programas_TFM_casa/barrido';
fs = 250000;
index = 1;
divider = zeros(1,3);
ini_sweep = zeros(1,3);
v=[];
t=[];
trig=[];

full_path = [dir_path];%, num2str(i)];
dir_info = dir(full_path);
num_files = sum(~[dir_info.isdir]);
str = ['Abriendo el directorio ',full_path,' ...'];
disp(str);

for i = 1:num_files % dentro de cada carpeta de datos existe una serie de subcarpetas, por lo
que se
    % hace una iteracion para cada subcarpeta, cada una
    % contienen un barrido independiente del mismo experimento

    str = ['Lectura del fichero ',dir_info(i+2).name,' ...'];
    disp(str);
    file_path = [full_path,'/',dir_info(i+2).name];
    load(file_path);

    ini_sweep(i)=1;
    divider(i)=length(datos);
    %ini_sweep(2)=ini_sweep(1);
    a=(datos(ini_sweep(i):divider(i)));
    a_n=(data_ref(ini_sweep(i):divider(i)));

    referencia(1:length(a),i)= a(:,1);
    senal_medida(1:length(a_n),i)= a_n(:,1);
    trigger_b(1:length(a),i)=trigger(:,1);

    b=tiempo_vec(ini_sweep(i):divider(i));
    time(1:length(b),i) = b(:,1);

    c=trigger(ini_sweep(i):divider(i));
    trig(1:length(c),i) =c(:,1);
end

save('data_logxx_full.mat','datos','data_ref','tiempo_vec','divider','ini_sweep','trigger','lo
ng_ini','long_fin','referencia');
save('data_pru.mat','referencia','senal_medida','long_ini','long_fin');
load('data_pru.mat');
for barrido = 1:num_files
    referencia=referencia(:,barrido)';
    medida=senal_medida(:,barrido)';
    val_corte1=length(medida);
    corteinicio = 1;
    t1=tiempo_vec(1:length(medida));
    tdatos=tiempo_vec(1:length(medida));

    end
    disp('datos raw completo')

for barrido = 1:num_files
    disp(['Se está analizando la interferencia del barrido', barrido]);
    senal=referencia;
    lands = linspace(long_ini,long_fin,length(senal));
    senal_t_ventana= Filtrar_FFT(senal,fs,3000);
    tiempo_ventana = linspace(0, length(senal_t_ventana)/fs, length(senal_t_ventana));

    fase=unwrap(angle(senal_t_ventana));
    fase_acumulada(1:length(fase),barrido)= fase; % fase_b es la fase acumulada, unwrapped
    fase_instantanea=angle(senal_t_ventana); % fase_normal es la fase instantánea
    retardo_grupo=unwrap(diff(unwrap(angle(senal_t_ventana))));


```

```

retardo_grupo_acumulado(1:length(retardo_grupo),barrido)= retardo_grupo;

frec_optica=(2*pi*3E17)./(lands);
res = diff(fase)./ diff(frec_optica);
res(1)=res(2);
res(length(res))=res(length(res)-1);

res_prom=res;
promediado=2000;
for x= 1 : length(res)-promediado
    res_prom(x) = mean(res(x:x+promediado));
end

figure()
subplot(2,1,1)
plot(lands(1:end-1),1e12*res);%ps
title('Retardo de grupo');
subplot(2,1,2)
plot(lands(1:end-1),1e12*res_prom);
title('Retardo de grupo promediado');

grid minor
sgtitle('Representacion del retardo de grupo')
ylabel('Retardo de grupo (ps)')
xlabel('Longitud de onda (nm)')
end

save('datos_fase.mat','fase_acumulada','fase_instantanea','lands'); %fase_b es la fase
acumulada, unwrapped. fase_normal es la fase instantanea
end

```

```

function ajuste_y_procesado_camino_2(fs)

dir_path='/Users/Jorge/Desktop/programas_TFM_modificar_casa/barrido';
load('data_pru.mat');
load('landas_calculadas_iter1.mat');

rangoFrec=450;
senal_medida_envent=Filtrar_FFT(senal_medida,fs,rangoFrec);
tiempo_ventana=linspace(0,length(senal_medida_envent)/fs,length(senal_medida_envent));
fase=unwrap(angle(senal_medida_envent));

figure()
plot(lambda_corregida,fase);

frec_optica=(2*pi*3E17)./(lambda_corregida);

res = diff(fase)./ diff(frec_optica);
res(1)=res(2);
res(length(res))=res(length(res)-1);

res_prom=res;
promediado=2000;
for x= 1 : length(res)-promediado
    res_prom(x) = mean(res(x:x+promediado));
end
frec_optica_sin_corr=(2*pi*3E17)./(lands_short);

res_sin_corr=diff(fase)./ diff(frec_optica_sin_corr');

figure()
subplot(2,1,1)
plot(lambda_corregida(1:end-1),1e12*res);%ps
title('Retardo de grupo');
subplot(2,1,2)
plot(lambda_corregida(1:end-1),1e12*res_prom);
title('Retardo de grupo promediado');

grid minor
sgtitle('Representacion del retardo de grupo')
ylabel('Retardo de grupo (ps)')
xlabel('Longitud de onda (nm)')

figure()
plot(lambda_corregida(1:end-1),1e12*res);%ps
hold on
plot(lands_short(1:end-1),res_sin_corr*1e12);
hold off
legend('Con corrección', 'Sin corrección')
title('Retardo de grupo')

end

```

```

function [senal_t_ventana]=Filtrar_FFT(senal,fs, rangoFrec)
sig_fft=fft(senal);
tf = linspace(-fs/2,fs/2,length(sig_fft));
o1_log=fftshift((sig_fft));

figure()
plot(tf,abs(o1_log));
title ('Espectro de la señal')
xlabel('Frecuencia (Hz)');
ylabel('Amplitud');

[~, index_primer_pico] = max(abs(o1_log));
frecuencia_primer_pico = tf(index_primer_pico);

% Elimina la región que rodea al primer pico para encontrar el segundo pico máximo
indice_ini_segundo_pico = index_primer_pico + 100; % Empieza después del primer pico

% Encuentra el segundo pico máximo
[~, index_segundo_pico] = max(abs(o1_log(indice_ini_segundo_pico:end)));
index_segundo_pico = index_segundo_pico + indice_ini_segundo_pico - 1; % Ajuste del índice al espacio original

margen_ventana = rangoFrec; %30000; %Margen alrededor del pico
indice_inicio_ventana = max(1, index_segundo_pico - margen_ventana);
indice_fin_ventana = min(length(o1_log), index_segundo_pico + margen_ventana);
if rangoFrec==0 %si estoy filtrando la celula de gas
    indice_inicio_ventana = index_primer_pico + 10;
    indice_fin_ventana = index_primer_pico + 50000;
end

% Creamos la ventana para el segundo pico
ventana_segundo_pico = zeros(size(o1_log));
ventana_segundo_pico(indice_inicio_ventana:indice_fin_ventana) = 1;

senal_f_ventana=fftshift(sig_fft).*ventana_segundo_pico;

% Ploteamos la señal original con la ventana del segundo pico
frec_ventana=tf;
figure()
plot(frec_ventana, abs(o1_log));
hold on;
plot(frec_ventana, (abs(senal_f_ventana)), 'r', 'LineWidth', 2);
xlabel('Frecuencia (Hz)');
ylabel('Magnitud');
title('Segundo pico en la FFT con ventana');
legend('FFT', 'Segundo pico con ventana');
hold off;

senal_t_ventana = ifft(fftshift(senal_f_ventana));
tiempo_ventana = linspace(0, length(senal_t_ventana)/fs, length(senal_t_ventana));
end

```

## Anexo 5. Programa ODFR

```
function Reflectometria_Camino(valor_corte,val_corte_fase,fs)

fs = 320000;
close all;

load('data_pru_.mat');
v1_n = values_n;
corte=length(v1_n);
corteinicio = 1;

v1_n_shorted=v1_n(corteinicio:corte);

figure()
plot(v1_n_shorted);
title('Valores cortados de la señal')

senal=v1_n_shorted;

sig_fft=fft(senal);
frec=linspace(0,fs,length(sig_fft));
ylim([0,800]);

tf = linspace(-fs/2,fs/2,length(sig_fft));
o1_log=fftshift((sig_fft));
o1_log_filtrado = o1_log;
figure()
plot(abs(o1_log));
%ylim([0,800]);
title('sig');

muestras_a_ajustar1 =1689600:1689990;% 1689805:1689840;
muestras_a_ajustar2 =874100:874900,% 874270:874300;

o1_log_filtrado(muestras_a_ajustar1)=0.001;
o1_log_filtrado(muestras_a_ajustar2)=0.001;

plot(abs(o1_log_filtrado));
title('sig');
senal=ifft(ifftshift(o1_log_filtrado));

landa = linspace(long_ini,long_fin,length(senal));

load('landas_calculadas_iter1_CH.mat');
lambda_calculada=lambda_calculada_b(corteinicio:corte);

frecuencia = 3e17 ./ lambda_calculada;
% Definir una malla uniforme de frecuencias
frecuencia_uniforme = linspace(min(frecuencia), max(frecuencia), length(frecuencia));
% Interpolamos la señal a la malla uniforme de frecuencias
senal_uniforme = interp1(frecuencia, senal, frecuencia_uniforme, 'linear');
landa_uniforme = 3e17./frecuencia_uniforme;

sig_fft_correg=fft(senal_uniforme);
tf_correg = linspace(-fs/2,fs/2,length(sig_fft_correg));
o1_log_correg=fftshift((sig_fft_correg));

figure()
semilogy(tf(tf>0)/12.3374,o1_log((tf>0)));
hold on

figure()
semilogy(tf(tf>0)/(12.3374),abs(o1_log_correg(tf>0)));
title ('FFT de la señal 2')
end
```

## **9. Bibliografía**

- [1]. Mudabbir Badar, Ping Lu et al. Integrated Auxiliary Interferometer for Self-Correction of Nonlinear Tuning in Optical Frequency Domain Reflectometry
- [2]. Brian J.Soller, Dawn K. Gifford, et al. (2020). High Resolution Optical Frequency Domain Reflectometry for Characterization of Components and Assemblies. Luna Technologies Incorporated.
- [3]. Yu-Kuan Tsai, Zheng-Xiand Liao et al. Linearization of wavelength sweeping lasers for the construction of 4-D FMCW LiDAR images of slow-moving objects using baseband beat note signals
- [4]. Mark A.Heald, Jerry B. Marion. Classical Electromagnetic Radiation. Third Edition
- [5]. Justiniano Casas. Óptica.
- [6]. Alan V. Oppenheim, Alan S. Willsky and S. Hamid Nawab. Señales y Sistemas. Segunda edición
- [7]. Richard P. Feynman. The Feynman Lectures on Physics, Volume 1: Mainly Mechanics, Radiation and Heat. The New Millennium Edition.
- [8]. Léon Brillouin. Wave Propagation and Group Velocity. First Edition
- [9]. Manual TLS modelo 8164B. User Guide:  
[https://www.artisantg.com/info/Keysight\\_Agilent\\_8164B\\_Manual\\_20201221026.pdf?  
srsltid=AfmBOoomeRHQvaWMrJKIXILbVobwnlax2E5WhWUsTgF9vLtf5eDmQgSI](https://www.artisantg.com/info/Keysight_Agilent_8164B_Manual_20201221026.pdf?srsltid=AfmBOoomeRHQvaWMrJKIXILbVobwnlax2E5WhWUsTgF9vLtf5eDmQgSI)
- [10]. National Instruments, DAC modelo NI6281:  
[https://www.ni.com/es-es/shop/model/usb-  
6281.html?srsltid=AfmBOoqB2Jrn\\_ZfK0xev6-7kmNO05WQG4VXKv6zA1ERAr-  
P9I4Mvl77](https://www.ni.com/es-es/shop/model/usb-6281.html?srsltid=AfmBOoqB2Jrn_ZfK0xev6-7kmNO05WQG4VXKv6zA1ERAr-P9I4Mvl77)
- [11]. Fibras SMF Thorlabs:  
[https://www.thorlabs.com/newgroupage9.cfm?objectgroup\\_id=949](https://www.thorlabs.com/newgroupage9.cfm?objectgroup_id=949)
- [12]. Wavelength References, modelo QUAD-20/5/150/150  
<https://www.wavelengthreferences.com/wp-content/uploads/MultiGas.pdf>
- [13]. Andreas Othonos. Fiber Bragg Gratings. Fundamentals and Applications in Telecommunications and Sensing
- [14]. Max Born. Atomic Physics. Sixth Edition

**[15].** Zhicheng Zhong, Lin Zhang, et al. (2020). Improved Fourier Transformation Based Method for Accurate Phase and Amplitude Retrieval in Spectral Interferometry. *Journal of Optics*.

**[16].** J.P. Dakin. Optical Fiber Sensors: Systems and Applications, Volume 2, Chapter 15: Distributed Optical Fiber Sensor Systems.

**[17].** Utkarsh Sharma, Xing Wei and Jin U. Kang (editor). Fiber optic sensing and Imaging, Chapter 2. Fiber Optic Interferometric Devices

**[18].** Bahaa E.A. Saleh, Malvin Carl Teich. Fundamentals of Photonics. Third Edition

**[19].** Zhaobing Tian, Jack Barnes et al. (2008). Refractive Index Sensing With Mach-Zender Interferometer Based on Concatenating Two Single-Mode Fiber Tapers. *IEEE Photonics Technology Letters*.

**[20].** I.H. Malitson (1965). Interspecimen Comparison of the Refractive Index of Fused Silica. *Journal of The Optical Society of America*.

**[21].** Wenlong Yang, Rui Pan et al. (2020). A High Sensitivity Asymmetric Double Tapered Fiber Interference Sensor. *International Journal for Light and Electron Optics*.

**[22].** William S. Rodney and Robert J. Spindler (1954). Index of Refraction of Fused-Quartz Glass for Ultraviolet Visible, and Infrared Wavelengths. *Journal of Research of the National Bureau of Standards*.