



UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA

DEPARTAMENTO DE ELECTRÓNICA

Detección de peak espectral en sensores de fibra óptica basados en rejillas de Bragg a través de correlación cruzada implementada con transformada de Fourier.

Memoria de titulación para optar al título de
Ingeniero Civil Electrónico

Autor:

Sebastián Hernández C.

Profesor guía:

Dr. Marcelo Soto H.

Mayo - 2026

CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

Tipo de monografía (marcar una opción): ☐ Memoria o trabajo de título ☐ Tesis de Postgrado

Título del trabajo: **Detección de peak espectral en sensores de fibra óptica basados en rejillas de Bragg a través de correlación cruzada implementada con transformada de Fourier.**

Nombre del candidato(a): **Sebastián Hernández Contreras**

Carrera / Grado: **Ingeniería Civil Electrónica**

Campus: **Casa Central** Departamento: **Electrónica**

2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, Marcelo Soto, en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente

DEJO CONSTANCIA que:

- He revisado esta versión del documento y corresponde a la versión final aprobada del trabajo.
- El trabajo cumple con los requisitos académicos y de formato establecidos por la institución.

3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL (marcar una opción)

☒ El trabajo **NO contiene** información que amerite confidencialidad y puede ser publicado de inmediato en repositorio con acceso abierto.

☐ El trabajo **CONTIENE** información con potenciales implicancias de propiedad industrial o intelectual y requiere un periodo de confidencialidad (**embargo**) por (**marcar una opción**):

☐ 6 meses ☐ 12 meses ☐ 2 años ☐ 3 años ☐ 5 años ☐ 10 años

Fundamentación de la necesidad de confidencialidad (obligatorio si se solicita embargo):

4.- FIRMAS

Profesor(a) guía o director(a) de memoria o tesis:

Fecha: 21.07.2026

Firma: 

Estudiante o Candidato(a):

Fecha: 21. 07. 2026

Firma: 

Este formulario debe ser insertado como página 2 de la memoria o tesis, completado y firmado por estudiante y profesor(a) antes de la entrega en portal PRISMA de Biblioteca USM.

Agradecimientos

Quiero agradecer a mis padres por darme la oportunidad de realizar esta carrera y por su constante apoyo a lo largo de todos estos años.

A mis amigos, tanto del colegio como de la universidad, por escucharme y animarme en los momentos buenos y en los difíciles.

A mi pareja Ignacia, por su apoyo en los momentos finales y por motivarme para concluir este trabajo.

Gracias a mi profesor guía, Marcelo Soto, por su paciencia, orientación y enseñanzas.

Un agradecimiento especial a Sebastián San Martín, por su invaluable guía, sus conocimientos compartidos y por ser un punto de apoyo fundamental en el análisis y las conclusiones de este trabajo. Quisiera agradecer también a los proyectos que hicieron posible la realización de este trabajo:

- FONDECYT Regular 1241085, ANID.
- FONDEQUIP EQM180226, ANID.
- FONDEQUIP EQM220113, ANID.
- Centro AC3E CIA250006, ANID.

Resumen

Esta memoria de titulación presenta el desarrollo, implementación y validación de un algoritmo para la detección del peak espectral en sensores de fibra óptica basados en rejillas de Bragg (FBG). El objetivo principal consiste en estimar con alta precisión el desplazamiento de la longitud de onda de Bragg ($\Delta\lambda_B$), el cual está directamente relacionado con variaciones de magnitudes físicas como la temperatura. Se realizó un análisis comparativo de diversos algoritmos de detección, seleccionándose la correlación cruzada implementada mediante la Transformada Rápida de Fourier (FFT) debido a su equilibrio óptimo entre precisión, robustez al ruido y eficiencia computacional ($O(N \log N)$).

En la etapa experimental, se implementaron cinco variantes del algoritmo en MATLAB, destacando el método de correlación cruzada por FFT con ajuste polinomial cuadrático. Este método demostró un error cuadrático medio (RMS) inferior a 0.8°C y una alta estabilidad incluso bajo condiciones adversas de atenuación óptica (hasta 20 dB), validando su robustez frente a relaciones señal-ruido bajas. Posteriormente, con el fin de habilitar su despliegue en sistemas embebidos, se migró el algoritmo desde MATLAB a lenguaje C. Se compararon dos enfoques: generación automática de código mediante MATLAB Embedded Coder y una implementación manual optimizada utilizando la librería FFTW. Los resultados evidenciaron que la implementación manual en C logra una mejora en el tiempo de ejecución de 4.3 veces respecto al entorno original de MATLAB, manteniendo la precisión de la detección. Se concluye que la metodología propuesta es viable para aplicaciones de monitoreo estructural y sensores distribuidos que requieren procesamiento en tiempo real.

Palabras clave: Sensores de Fibra Óptica, Rejillas de Bragg (FBG), Correlación Cruzada, Transformada de Fourier (FFT), Implementación Embebida, Detección de Peak Espectral.

Abstract

This graduation thesis presents the development, implementation, and validation of an algorithm for spectral peak detection in fiber optic sensors based on Fiber Bragg Gratings (FBG). The main objective is to accurately estimate the Bragg wavelength shift ($\Delta\lambda_B$), which is directly related to variations in physical magnitudes such as temperature. A comparative analysis of various detection algorithms was conducted, selecting cross-correlation implemented via the Fast Fourier Transform (FFT) due to its optimal balance between precision, noise robustness, and computational efficiency ($O(N \log N)$).

In the experimental stage, five variants of the algorithm were implemented in MATLAB, highlighting the cross-correlation method by FFT with quadratic polynomial fitting. This method demonstrated a Root Mean Square (RMS) error below 0.8°C and high stability even under adverse optical attenuation conditions (up to 20 dB), validating its robustness against low signal-to-noise ratios. Subsequently, to enable deployment in embedded systems, the algorithm was migrated from MATLAB to C language. Two approaches were compared: automatic code generation using MATLAB Embedded Coder and a manual optimized implementation using the FFTW library. Results evidenced that the manual C implementation achieves a 4.3-fold improvement in execution time compared to the original MATLAB environment, maintaining detection precision. It is concluded that the proposed methodology is viable for structural monitoring applications and distributed sensors requiring real-time processing.

Keywords: Fiber Optic Sensors, Fiber Bragg Gratings (FBG), Cross-Correlation, Fourier Transform (FFT), Embedded Implementation, Spectral Peak Detection.

Índice general

Índice general	4
Índice de figuras	6
1 Introducción	8
1.1 Motivación	8
1.2 Objetivos	9
1.3 Alcances y Limitaciones	9
1.4 Estructura de la Tesis	9
2 Marco teórico	10
2.1 Fundamento de las rejillas de Bragg	10
2.1.1 Fundamentos básicos	10
2.1.2 Resultados de un FBG uniforme	11
2.1.3 Sensores Ópticos utilizando FBG	11
2.1.4 Clasificación de FBG	12
2.1.5 Inscripción de FBGs	15
2.1.6 Interrogación de FBGs	16
2.2 Algoritmos de detección de peak	18
2.2.1 Métodos Directos	18
2.2.2 Métodos por ajuste de curva	19
2.2.3 Métodos basados en transformadas	20
2.2.4 Métodos basados en optimización	22
2.2.5 Algoritmos basados en correlación	23
2.2.6 Comparación y elección del método	25
3 Implementación y Resultados	28
3.1 Obtención de datos e implementación del algoritmo	28
3.1.1 Implementación de los algoritmos de correlación cruzada	28
3.1.2 Entorno de ejecución	33
3.1.3 Adquisición de datos	34
3.1.4 Calibración del sensor FBG	40
3.2 Análisis y resultados.	42
3.2.1 Metodología de análisis	42
3.2.2 Resultados	43
3.2.3 Análisis general de resultados	48

4	Implementación de Algoritmos en Lenguaje C	50
4.1	Generación de Código Mediante MATLAB Embedded Coder	51
4.2	Implementación de Código Propio en C	52
4.3	Análisis y Comparación de Resultados de Implementación	53
4.3.1	Comparación de la Precisión de Detección	53
4.3.2	Análisis de Rendimiento Temporal	55
4.4	Análisis de las Implementaciones	55
5	Conclusiones	57
	Bibliografía	58

Índice de figuras

2.1	Esquema de una FBG uniforme [2]	11
2.2	Reflectividad espectral típica de un FBG uniforme, considerando $\lambda_B = 1550$ nm [2].	12
2.3	Esquema de una FBG <i>chirpeada</i> [2]	14
2.4	Espectro de reflectividad de una FBG <i>chirpeada</i> para diferentes experimentos de <i>chirp</i> [2]	14
2.5	Perfil de modulación del índice de refracción en una FBG apodizada [2]	14
2.6	Esquema de inscripción por interferometría [2]	15
2.7	Esquema de inscripción mediante <i>phase-mask</i> [2]	16
2.8	Esquema de inscripción <i>point-by-point</i> [2]	16
2.9	Esquema de interrogación de FBGs utilizando luz espectralmente grande.(basado en el documento [3])	17
2.10	Esquema de interrogación de FBGs utilizando barrido de luz.(basado en el documento [3])	18
2.11	Detección del <i>peak</i> espectral mediante el método X-dB [3]	19
2.12	Estimación de λ_B mediante el método del centroide [3]	20
2.13	Implementación del método de correlación cruzada. A la izquierda: espectro de referencia y espectro desplazado en 5 pm. A la derecha: función de correlación mostrando un corrimiento detectado de 45 pm [3].	23
2.14	Esquema de implementación del método <i>Correlation Polynomial Fit</i> , comparado con el método <i>Wavelength-Shifted</i> [3].	24
3.1	Esquema del montaje experimental para la adquisición de datos. Incluye interrogador Luna S155, VOA, FBG en vaso con agua caliente.	34
3.2	Interfaz del software ENLIGHT utilizado para la adquisición y visualización de espectros FBG. Permite exportar datos en formato .txt.	35
3.3	Medición de potencia óptica en la entrada del VOA. Se registra el nivel de señal antes de la atenuación.(a)Medición de potencia óptica en la salida del VOA con atenuación mínima. Muestra la pérdida base de 0.80 dB del dispositivo. (b)	35
3.4	Comparación del espectro reflejado por la FBG antes y después de la conversión de escala. (a) Espectro original en escala logarítmica. (b) Espectro en escala lineal . . .	36
3.5	Comparación del espectro reflejado por la FBG antes y después de la conversión de escala. (a) Escala logarítmica. (b) Escala lineal.	37
3.6	Evolución de la temperatura medida con termómetro de referencia durante el Experimento 1. Los puntos representan mediciones reales cada 50 s.	38
3.7	Ejemplo del espectro de la FBG original y sus versiones submuestreadas obtenidas mediante <i>downsampling</i> con diferentes factores n	38

3.8	Experimento 2 el cual presenta una atenuación de 10dB. (a) Escala logarítmica. (b) Escala lineal.	39
3.9	Comparación de los espectros reflejados en los Experimentos 1 y 2. A pesar de la atenuación de 10 dB. (a) Escala logarítmica. (b) Escala lineal.	40
3.10	Potencia máxima reflejada para cada una de las 12 mediciones del Experimento 3, según el nivel de atenuación total aplicada por el VOA.(a) Escala logarítmica. (b) Escala Lineal.	41
3.11	Comparación entre la temperatura estimada a partir del desplazamiento espectral detectado por el algoritmo de FFT (<i>puntos azules</i>) y la temperatura medida por el termómetro (<i>puntos rojos</i>) durante el Experimento 1. El rango térmico va desde 73°C hasta 48°C. La alta concordancia entre ambas señales valida la precisión del método propuesto.	43
3.12	Evolución del error RMS y desviación estándar en función de la temperatura para los cuatro algoritmos.(a) offset RMS. (b) Desviación estándar.	45
3.13	Desempeño de los métodos para el experimento con atenuación de 20 dB.(a) Comparación entre el desempeño del algoritmo de FFT con y sin atenuación de 20 dB.(b)	46
3.14	Evolución del error RMS y desviación estándar en función de la atenuación total para los cinco algoritmos.(a) offset RMS. (b) Desviación estándar.	47
3.15	Evolución del error RMS y desviación estándar en función de la atenuación total para los cinco algoritmos.(a) offset RMS. (b) Desviación estándar.	48
4.1	Interfaz de MATLAB Coder mostrando el análisis del algoritmo antes de la generación de código C. Permite verificar compatibilidad, modificar parámetros y pre-visualización de salidas.	51
4.2	Comparación de la temperaturas teóricas medidas junto a las estimadas a partir de los métodos. (a)Implementación inicial de Matlab.(b)Implementación en Coder (MEX).(c) Implementación C	53
4.3	Desempeño de las distintas implementaciones de los parámetros de estudio en función de la temperatura.(a) offset RMS. (b) Desviación estándar	54
4.4	Desempeño de las distintas implementaciones de los parámetros de estudio en función de la temperatura.(a) offset RMS. (b) Desviación estándar	54

Capítulo 1

Introducción

1.1 Motivación

En las últimas décadas, los sensores de fibra óptica basados en rejillas de Bragg (FBG, *Fiber Bragg Gratings*) se han consolidado como una tecnología clave para aplicaciones de monitoreo estructural, sensado distribuido y control de procesos industriales [1]. Sus ventajas inherentes —inmunidad a interferencias electromagnéticas, capacidad de multiplexación en longitud de onda (WDM), pequeñas dimensiones y alta sensibilidad a temperatura y deformación [2, 3]— los hacen especialmente atractivos en entornos donde los sensores eléctricos convencionales presentan limitaciones críticas.

El principio de funcionamiento de una FBG se fundamenta en una modulación periódica del índice de refracción en el núcleo de una fibra óptica, la cual actúa como un filtro reflectivo selectivo [2]. La longitud de onda central reflejada, conocida como longitud de onda de Bragg (λ_B), está determinada por la condición de Bragg:

$$\lambda_B = 2n_{\text{eff}}\Lambda, \quad (1.1)$$

donde n_{eff} es el índice de refracción efectivo del modo guiado y Λ es el período espacial de la modulación [4]. Cualquier perturbación externa, como cambios de temperatura (ΔT) o deformación axial ($\Delta\epsilon$), altera estos parámetros, provocando un corrimiento espectral $\Delta\lambda_B$ que codifica la información física medida [5, 6]:

$$\Delta\lambda_B = k_\epsilon\Delta\epsilon + k_T\Delta T, \quad (1.2)$$

donde $k_\epsilon \approx 1 \text{ pm}/\mu\epsilon$ y $k_T \approx 10 \text{ pm}/^\circ\text{C}$ representan las sensibilidades típicas para una FBG en fibra de sílice [7, 8]. La utilidad práctica de estos sensores depende, por tanto, de la capacidad del sistema de interrogación para detectar con precisión el desplazamiento del peak espectral ($\Delta\lambda_B$), el cual constituye el parámetro fundamental para inferir la magnitud física de interés [3].

Sin embargo, la medición directa del peak espectral presenta desafíos significativos, especialmente en condiciones de baja relación señal-ruido (SNR), resolución espectral limitada o cuando se requieren altas tasas de muestreo para aplicaciones en tiempo real [3, 9]. Existen diversos algoritmos para la detección del peak espectral, los cuales pueden clasificarse en métodos directos, ajustes de curva, métodos basados en transformadas y técnicas de optimización [3, 9]. Los métodos directos, como la detección de máximo simple o el centroide, son computacionalmente eficientes pero sensibles al ruido [10]. Por otro lado, los métodos de ajuste de curva (polinomial o gaussiano) ofrecen mayor precisión sub-muestral pero incrementan la carga computacional [3]. En este escenario, los métodos basados en correlación cruzada implementados mediante la Transformada Rápida de Fourier (FFT) han demostrado un equilibrio óptimo entre precisión y eficiencia, reduciendo la complejidad algorítmica de $\mathcal{O}(N^2)$ a $\mathcal{O}(N \log N)$ [11, 12]. Esta eficiencia es crucial para habilitar el procesamiento en sistemas con restricciones computacionales.

1.2 Objetivos

El objetivo principal de esta memoria es desarrollar, implementar y validar un algoritmo de detección de peak espectral basado en correlación cruzada por FFT para sensores FBG. Específicamente, se busca:

- Implementar y comparar cinco variantes del algoritmo de correlación cruzada en MATLAB: `xcorr`, FFT estándar, FFT con *zero-padding*, FFT con ajuste polinomial y correlación directa.
- Validar experimentalmente la precisión y robustez de los algoritmos mediante adquisición de datos con un interrogador comercial Luna S155, bajo variaciones controladas de temperatura y atenuación óptica.
- Migrar el algoritmo seleccionado a lenguaje C mediante dos enfoques: generación automática con MATLAB Embedded Coder y una implementación manual optimizada utilizando la librería FFTW [13].
- Evaluar el desempeño de ambas implementaciones en C en términos de precisión de detección y tiempo de ejecución, comparándolas con la versión original en MATLAB.

1.3 Alcances y Limitaciones

Es importante destacar que el presente trabajo se limita a la etapa de implementación algorítmica en lenguaje C. Si bien el código generado está diseñado para ser portable a sistemas embebidos, **no se realiza una implementación física en hardware** (microcontrolador, FPGA o plataforma embebida). La validación del código en C se lleva a cabo en entorno de escritorio, comparando su precisión y eficiencia frente a la implementación original en MATLAB. La migración a una plataforma embebida específica (por ejemplo, STM32 [14] o ESP32 [15]) queda como trabajo futuro, una vez validada la portabilidad y el desempeño del algoritmo en lenguaje de bajo nivel.

1.4 Estructura de la Tesis

La memoria se organiza de la siguiente manera: El **Capítulo 2** presenta el marco teórico, detallando los fundamentos de las FBG, sus métodos de interrogación y una revisión exhaustiva de los algoritmos de detección de peak espectral reportados en la literatura, con énfasis en los métodos basados en correlación cruzada y FFT [2, 4, 3, 11]. El **Capítulo 3** describe la implementación de las cinco variantes del algoritmo en MATLAB, el montaje experimental con interrogador Luna S155, el preprocesamiento de los espectros y el análisis de resultados en términos de precisión (RMS), robustez al ruido y eficiencia computacional. Finalmente, el **Capítulo 4** aborda la migración del algoritmo seleccionado a lenguaje C, comparando la generación automática de código mediante MATLAB Embedded Coder con una implementación manual optimizada utilizando la librería FFTW [13]. Se evalúan ambas versiones en términos de precisión de detección y tiempo de ejecución, validando su viabilidad para futuras aplicaciones en tiempo real. Con esta organización, la tesis busca aportar una metodología reproducible y eficiente para la interrogación de sensores FBG, sentando las bases para su posterior despliegue en sistemas embebidos de bajo consumo y alta confiabilidad.

Capítulo 2

Marco teórico

2.1 Fundamento de las rejillas de Bragg

2.1.1 Fundamentos básicos

Considerando la definición de una *fiber Bragg grating* (FBG), esta consiste en una modulación periódica del índice de refracción en el núcleo de una fibra óptica [2]. Esta estructura actúa como un filtro óptico altamente selectivo, reflejando únicamente aquellas longitudes de onda que satisfacen la condición de Bragg, mientras transmite el resto del espectro. Dicha condición surge de la conservación del momento y de la energía en la interacción entre la onda óptica y la estructura periódica de la rejilla. Para conservar la energía, las ondas incidente y reflejada deben tener la misma frecuencia; por otro lado, la conservación del momento exige que la suma del vector de onda incidente k_i y el vector de retroceso de la rejilla K sea igual al vector de onda reflejado k_f , como se muestra en la siguiente ecuación [2]:

$$k_i + K = k_f \quad (2.1)$$

El vector de onda de la rejilla K es perpendicular a los planos de modulación del índice de refracción y tiene una magnitud $|K| = 2\pi/\Lambda$. Dado que los vectores de onda incidente y reflejado tienen la misma magnitud pero direcciones opuestas, se puede simplificar la ecuación de conservación del momento, obteniendo la condición de Bragg en su forma escalar [2],[4]:

$$2 \left(\frac{2\pi\eta_{\text{eff}}}{\lambda_B} \right) = \frac{2\pi}{\Lambda} \quad (2.2)$$

$$\lambda_B = 2\eta_{\text{eff}}\Lambda \quad (2.3)$$

donde η_{eff} es el índice de refracción efectivo del modo guiado en la fibra, Λ es el período espacial de la modulación del índice de refracción, y λ_B es la longitud de onda de Bragg, es decir, la longitud de onda central reflejada en espacio libre. Aunque λ_B es la única longitud de onda que satisface exactamente la condición de interferencia constructiva completa, la FBG refleja un rango finito de longitudes de onda alrededor de λ_B , como se observa en la figura 2.1. Este comportamiento se debe a la naturaleza distribuida de la reflexión, originada por el espaciamiento coherente de la luz a lo largo de la rejilla. Las longitudes de onda cercanas a λ_B presentan una superposición constructiva, generando una banda espectral reflejada con forma de perfil de Bragg. Por el contrario, las longitudes de onda alejadas de λ_B sufren interferencia destructiva debido a la acumulación de desfases a lo largo de la rejilla, resultando en una atenuación efectiva de su amplitud reflejada.

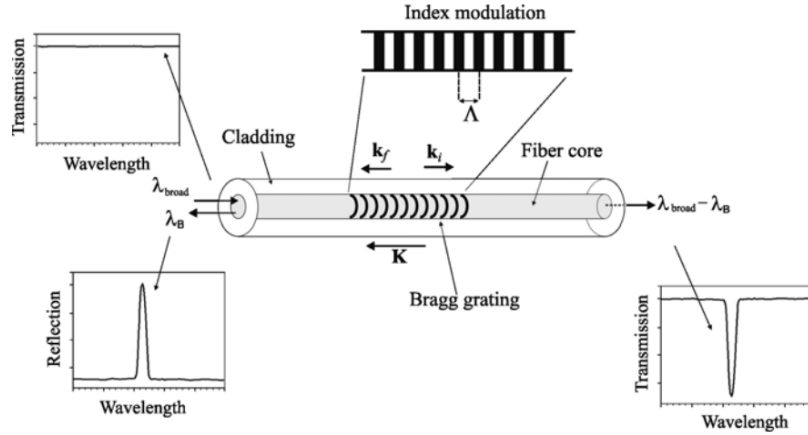


Figure 2.1: Esquema de una FBG uniforme [2]

2.1.2 Resultados de un FBG uniforme

La teoría de acoplamiento de modos (coupled-mode theory) permite describir analíticamente el comportamiento espectral de una FBG uniforme [2],[3]. Bajo esta aproximación, la reflectividad $R(\lambda)$ de una FBG con modulación periódica y constante del índice de refracción en su longitud total L viene dada por:

$$R(\lambda) = \frac{\sinh^2 \left(L \sqrt{k^2 - \sigma(\lambda)^2} \right)}{\cosh^2 \left(L \sqrt{k^2 - \sigma(\lambda)^2} \right) - \frac{\sigma(\lambda)^2}{k^2}} \quad (2.4)$$

donde $R(\lambda)$ es la reflectividad espectral, λ la longitud de onda en espacio libre, L la longitud total de la rejilla, y k el coeficiente de acoplamiento, un parámetro adimensional que caracteriza el contraste del índice de refracción de la rejilla. Este coeficiente está relacionado con la amplitud de modulación del índice de refracción $\delta_{\eta_{\text{eff}}}$ mediante $k = \frac{\pi}{\lambda_B} \delta_{\eta_{\text{eff}}}$, y determina tanto el ancho de banda (FWHM) como el valor máximo de reflectividad de la rejilla.

El término $\sigma(\lambda)$, conocido como corrimiento espectral generalizado (o *detuning*), se define como:

$$\sigma(\lambda) = \frac{\pi}{\lambda} \delta_{\eta_{\text{eff}}} + 2\pi \eta_{\text{eff}} \left(\frac{1}{\lambda} - \frac{1}{\lambda_B} \right) \quad (2.5)$$

donde $\delta_{\eta_{\text{eff}}}$ es la amplitud de la modulación del índice de refracción efectivo. Cuando $\lambda = \lambda_B$, el corrimiento espectral $\sigma(\lambda_B) = \frac{\pi}{\lambda_B} \delta_{\eta_{\text{eff}}}$, y la reflectividad alcanza su valor máximo, dado por:

$$R_{\text{max}} = \tanh^2(kL) \quad (2.6)$$

Este comportamiento se ilustra en la figura 2.2, donde se muestra un espectro de reflectividad típico de una FBG uniforme, centrado en $\lambda_B = 1550$ nm [2]. La forma simétrica y de tipo lorentziano del perfil reflejado es característica de rejillas con modulación constante, y su ancho de banda y profundidad dependen directamente de los parámetros k y L .

2.1.3 Sensores Ópticos utilizando FBG

Las FBG son ampliamente utilizadas como sensores ópticos para la detección de variaciones de temperatura (ΔT) y deformación axial ($\Delta \epsilon$), debido a que ambos parámetros afectan tanto el índice

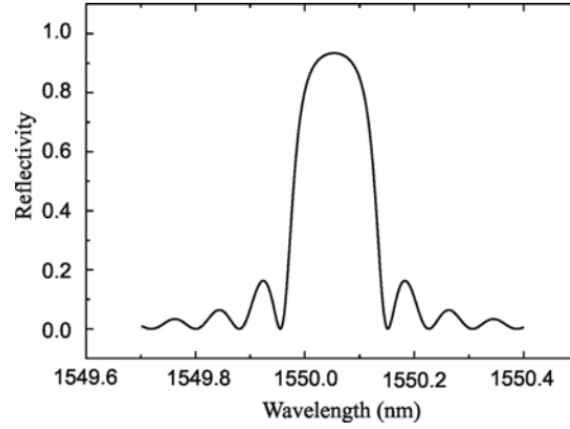


Figure 2.2: Reflectividad espectral típica de un FBG uniforme, considerando $\lambda_B = 1550$ nm [2].

de refracción efectivo (η_{eff}) como el período espacial de la rejilla (Λ). A partir de la ecuación (2.3), $\lambda_B = 2\eta_{\text{eff}}\Lambda$, la variación en la longitud de onda de Bragg puede expresarse como [2],[5],[1]:

$$\Delta\lambda_B = 2 \left[\Lambda \frac{\partial \eta_{\text{eff}}}{\partial \epsilon} + \eta_{\text{eff}} \frac{\partial \Lambda}{\partial \epsilon} \right] \Delta\epsilon + 2 \left[\Lambda \frac{\partial \eta_{\text{eff}}}{\partial T} + \eta_{\text{eff}} \frac{\partial \Lambda}{\partial T} \right] \Delta T \quad (2.7)$$

$$\Delta\lambda_B = k_\epsilon \Delta\epsilon + k_T \Delta T \quad (2.8)$$

donde k_ϵ y k_T representan, respectivamente, la sensibilidad de la FBG frente a cambios de deformación (strain) y temperatura. Estos coeficientes integran efectos térmicos y mecánicos sobre la estructura de la rejilla, incluyendo la expansión térmica del material, el cambio del índice de refracción con la temperatura (termo-óptico) y el efecto fotoelástico inducido por la deformación. Los valores típicos de sensibilidad para una FBG en fibra de sílice son aproximadamente $k_\epsilon \approx 1$ pm/ $\mu\epsilon$ y $k_T \approx 10$ pm/ $^\circ\text{C}$ [7]. Sin embargo, es importante destacar que la medición del desplazamiento de λ_B no permite distinguir directamente entre un cambio inducido por temperatura o por deformación, ya que ambos provocan un corrimiento espectral idéntico. Esta falta de selectividad inherente requiere el uso de técnicas de acondicionamiento, multiplexión espacial o configuraciones diferenciales (como una FBG de referencia aislada térmicamente) para desacoplar ambos efectos en aplicaciones prácticas.

2.1.4 Clasificación de FBG

- **Clasificación según las condiciones de inscripción**

Las FBG pueden clasificarse en cuatro tipos principales según las condiciones de inscripción, definidas por la intensidad y duración de la exposición a luz ultravioleta (UV), lo que influye directamente en su estabilidad térmica, profundidad de modulación del índice de refracción y régimen de operación. Además, el tipo de láser (por ejemplo, láser de excímero o interferométrico) y la dopación de la fibra óptica (como Ge-dopada) son factores determinantes en la formación de cada tipo.

- *TIPO I*: Este es el tipo más común en aplicaciones prácticas, debido a su proceso de inscripción sencillo, que utiliza luz UV de baja intensidad y fibras estándar sin sensibilización previa [2],[6]. Presentan una modulación moderada del índice de refracción y son adecuadas para operar en rangos de temperatura entre -40°C y 80°C , típicos en

telecomunicaciones y sensores ambientales. Sin embargo, su principal limitación es la degradación térmica a temperaturas superiores a 200°C.

- *TIPO IA*: Estas FBG se inscriben mediante exposición prolongada a luz UV de baja intensidad en fibras sensibilizadas con hidrógeno [2],[16]. Se caracterizan por una modulación negativa del índice de refracción (decremento respecto al valor base) y una baja sensibilidad térmica, lo que las hace útiles en aplicaciones donde se requiere estabilidad frente a variaciones de temperatura.
- *TIPO II*: Su fabricación requiere pulsos de alta intensidad (típicamente mediante láser UV de femtosegundos), que inducen daño estructural permanente en el núcleo de la fibra [2],[17]. Estas rejillas presentan una alta reflectividad y una excelente estabilidad térmica, soportando temperaturas superiores a 800°C durante horas, lo que las hace ideales para sensores en ambientes extremos.
- *TIPO IIA*: Formadas mediante exposición prolongada a luz UV de intensidad moderada, también en fibras sensibilizadas [2],[18]. A diferencia de las TIPO I, presentan una mayor estabilidad térmica, con degradación alrededor de 500°C. Poseen una modulación del índice de refracción más profunda que las TIPO I, pero menor que las TIPO II.

• Clasificación según la modulación del índice de refracción

La función de modulación espacial del índice de refracción $\eta(z)$ define el comportamiento espectral de una FBG. En una FBG uniforme, esta modulación es constante y está dada por:

$$\eta(z) = \eta_0 + \delta_\eta \cos\left(\frac{2\pi z}{\Lambda}\right)$$

donde η_0 es el índice de refracción promedio, δ_η la amplitud de modulación, z la posición longitudinal, y Λ el período espacial constante [2].

Una variante importante es la FBG con modulación *chirpeada*, en la que el período Λ varía a lo largo del eje longitudinal de la fibra. Esta estructura es ampliamente utilizada en comunicaciones ópticas para la compensación de dispersión cromática y el acondicionamiento de pulsos [2]. El cambio en $\Lambda(z)$ implica que la longitud de onda de Bragg local también varía con la posición, según:

$$\lambda_B(z) = 2\eta_{\text{eff}}(z)\Lambda(z)$$

En una FBG *chirpeada* linealmente, el período sigue una dependencia del tipo:

$$\Lambda(z) = \Lambda_0 + \Lambda_1 z$$

donde Λ_0 es el período inicial y Λ_1 representa la tasa de variación (gradiente de *chirp*). Como resultado, las secciones con menor Λ reflejan longitudes de onda más cortas, mientras que las de mayor Λ reflejan longitudes más largas, generando un espectro de reflectividad de ancho de banda ampliado, como se muestra en la figura 2.4.

Otra modificación clave es la apodización, que consiste en variar gradualmente la amplitud de modulación $\delta_\eta(z)$ a lo largo de la rejilla, con el fin de suprimir los lóbulos laterales en el espectro reflejado (ver figuras 2.2 y 2.4). Las FBG apodizadas son esenciales en sistemas de alta precisión donde se requiere una respuesta espectral limpia y mínima dispersión no deseada [2],[19]. En la figura 2.5 se ilustra un esquema de la modulación del índice de refracción en una FBG apodizada.

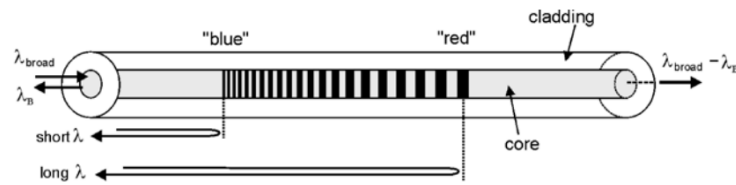


Figure 2.3: Esquema de una FBG *chirpeada* [2]

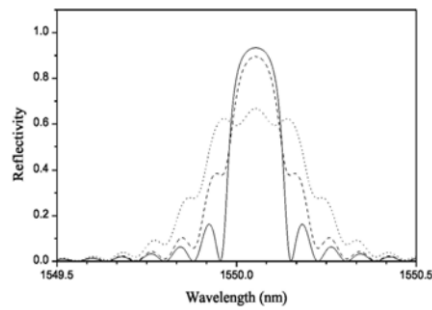


Figure 2.4: Espectro de reflectividad de una FBG *chirpeada* para diferentes experimentos de *chirp* [2]

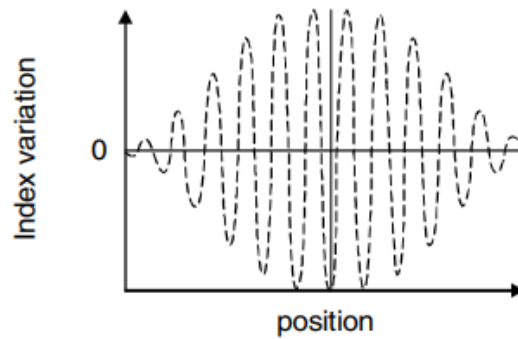


Figure 2.5: Perfil de modulación del índice de refracción en una FBG apodizada [2]

2.1.5 Inscripción de FBGs

La inscripción de una FBG en una fibra óptica consiste en modificar localmente el índice de refracción del núcleo de la fibra mediante exposición controlada a luz ultravioleta (UV), generando una estructura periódica capaz de reflejar selectivamente una longitud de onda específica. Este proceso requiere el uso de fibras fotosensibles, generalmente dopadas con germanio, para que el índice de refracción cambie de forma permanente tras la irradiación.

Los primeros métodos empleados correspondían a la técnica de inscripción interna [2],[20], en la que un láser de alta intensidad irradiaba el núcleo de la fibra a través de uno de sus extremos. Debido a que la luz se propagaba longitudinalmente, la modulación del índice de refracción resultaba no uniforme, requiriendo longitudes de rejilla considerablemente grandes para alcanzar una reflectividad significativa. Por esta razón, y debido a su baja reproducibilidad, este método ha quedado en desuso.

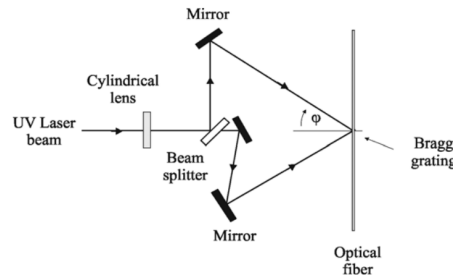


Figure 2.6: Esquema de inscripción por interferometría [2]

Un enfoque más avanzado es la inscripción por interferometría mostrado en la figura 2.6, que consiste en exponer una fibra fotosensible a un patrón de interferencia generado por dos haces coherentes de luz UV (típicamente de ~ 244 nm o 255 nm) [2],[21]. El método más común es el interferómetro divisor de amplitud, en el que un haz UV se divide y dirige hacia la fibra con el mismo ángulo respecto al eje longitudinal. La superposición de estos haces crea un patrón de interferencia periódico sobre la fibra, cuyo período Λ determina la longitud de onda de Bragg de la rejilla inscrita. Este período depende de la longitud de onda del láser λ_w y del ángulo de incidencia θ , según la siguiente relación:

$$\Lambda = \frac{\lambda_w}{2 \sin \theta} \quad \Rightarrow \quad \lambda_B = \frac{2\eta_{\text{eff}}\Lambda}{1} = \frac{\eta_{\text{eff}}\lambda_w}{\sin \theta} \quad (2.9)$$

donde λ_w es la longitud de onda de la luz UV en el aire, θ es el ángulo entre cada haz y el eje de la fibra, y η_{eff} es el índice de refracción efectivo del modo guiado. Una ventaja clave de este método es la flexibilidad para ajustar λ_B mediante el control preciso de θ . Además, el largo de la rejilla puede definirse mediante el tiempo de exposición o la máscara de iluminación, permitiendo diseñar FBGs con diferentes anchos de banda y niveles de reflectividad.

Sin embargo, este método es altamente sensible a perturbaciones mecánicas y térmicas, ya que cualquier desplazamiento en los espejos o variación en el índice del aire afecta el patrón de interferencia. Asimismo, requiere una fuente UV con alta coherencia espacial y temporal, así como estabilidad térmica y de potencia, lo que incrementa el costo y complejidad del sistema.

La figura 2.7 muestra otra técnica ampliamente utilizada, es la inscripción mediante *phase-mask*, que emplea una máscara de fase de sílice con una estructura periódica grabada en su superficie [2],[22]. Al incidir un haz UV plano sobre esta máscara, esta difracta la luz principalmente en los órdenes -1 y $+1$, cuya interferencia sobre la fibra genera un patrón periódico con un período igual a la mitad del

período de la máscara. Este método es robusto y estable, ya que solo requiere un componente óptico principal (la máscara), reduciendo significativamente la sensibilidad a vibraciones y alineaciones. Además, permite una alta reproducibilidad en la fabricación de FBGs uniformes. No obstante, la intensidad del patrón de interferencia disminuye con la distancia entre la máscara y la fibra, por lo que se debe mantener un acoplamiento cercano (típicamente en contacto o con separación mínima). Esto implica un riesgo de daño físico o contaminación de la fibra, especialmente bajo irradiación prolongada con alta densidad de potencia UV.

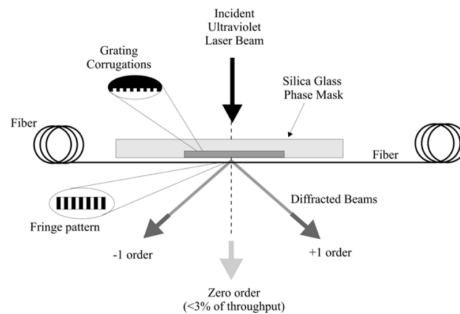


Figure 2.7: Esquema de inscripcón mediante *phase-mask* [2]

Finalmente la figura 2.8 muestra el método *point-by-point* el cual consiste en escribir cada plano de la rejilla de forma individual, utilizando un pulso láser focalizado y moviendo precisamente la fibra (o el haz) entre cada exposición [2],[23]. Este enfoque permite diseñar FBGs con perfiles de modulacón arbitrarios (por ejemplo, apodizados, chirpeados o multifrecuencia), ya que cada punto puede controlarse independientemente.

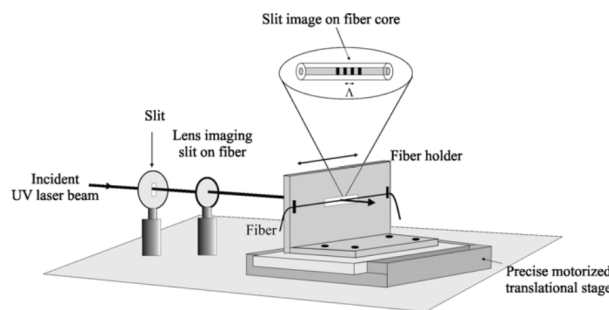


Figure 2.8: Esquema de inscripcón *point-by-point* [2]

La principal desventaja de este método es su baja velocidad y la extrema precisión necesaria en el posicionamiento (del orden de nanómetros), lo que lo hace susceptible a errores por vibraciones, deriva térmica o imperfecciones en el sistema de desplazamiento. A pesar de esto, es especialmente útil para la fabricación de rejillas altamente personalizadas, como FBGs para multiplexión densa o sensores inteligentes.

2.1.6 Interrogación de FBGs

La interrogación de una *fiber Bragg grating* (FBG) puede realizarse mediante dos enfoques principales, dependiendo del tipo de fuente óptica utilizada: “interrogación con fuente de luz de ancho espectral grande” (Luz blanca) y “interrogación con láser sintonizable”.

El esquema de la Figura 2.9 muestra el enfoque que utiliza una fuente de luz con un ancho espectral amplio, como un diodo emisor de luz superluminiscente (SLED) o el ruido de emisión espontánea amplificada (ASE) generado por un amplificador de fibra dopada con erbio (EDFA) [4, 3]. La luz blanca se acopla a la fibra que contiene la FBG mediante un acoplador óptico (típicamente 50/50), y la porción reflejada por la rejilla regresa al acoplador, que la dirige hacia un sistema de detección espectral, como un Analizador de Espectro Óptico (OSA, *Optical Spectrum Analyzer*). El OSA mide la intensidad reflejada como función de la longitud de onda, permitiendo identificar la longitud de onda de Bragg (λ_B) a partir del pico de reflectividad con alta resolución espectral [3]. Este método permite la interrogación simultánea de múltiples FBGs con diferentes λ_B , lo que facilita la multiplexación en el dominio de la longitud de onda (WDM, *Wavelength Division Multiplexing*) [4, 3]. Además, al combinar este esquema con un conmutador óptico, es posible implementar redes de sensores cuasi-distribuidos para monitoreo estructural a gran escala [3].

El espectrómetro mide la intensidad reflejada como función de la longitud de onda, permitiendo identificar la longitud de onda de Bragg (λ_B) a partir del pico de reflectividad. Este método permite la interrogación simultánea de múltiples FBGs con diferentes λ_B , lo que facilita la multiplexación en el dominio de la longitud de onda (WDM). Además, al combinar este esquema con un *switch* óptico, es posible implementar redes de sensores cuasi-distribuidos.

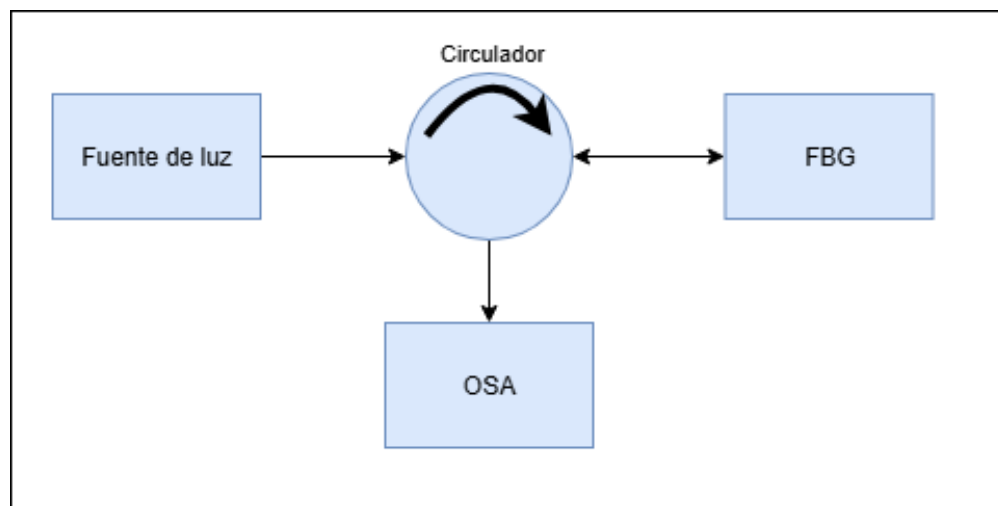


Figure 2.9: Esquema de interrogación de FBGs utilizando luz espectralmente grande.(basado en el documento [3])

El segundo enfoque se basa en un láser sintonizable de baja potencia, cuya longitud de onda se varía de forma controlada mediante una señal de barrido (rampa o diente de sierra) generada por un oscilador o generador de funciones. Este esquema, mostrado en la Figura 2.10, permite escanear un rango espectral específico y detectar la reflexión de la FBG cuando la longitud de onda del láser coincide con λ_B .

La luz reflejada se dirige, mediante acopladores ópticos, hacia uno o varios fotodetectores, cuya señal se digitaliza y procesa para determinar el instante de máxima reflexión. Este tiempo o fase se correlaciona directamente con λ_B .

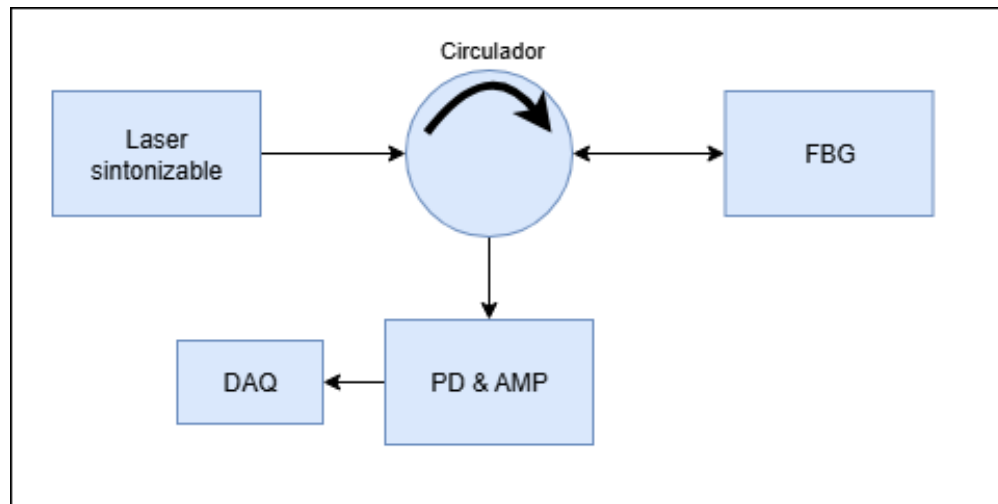


Figure 2.10: Esquema de interrogación de FBGs utilizando barrido de luz.(basado en el documento [3])

2.2 Algoritmos de detección de peak

La detección del *peak* espectral en una FBG constituye uno de los aspectos más críticos en su uso como sensor, ya que a partir de este valor se determina el desplazamiento de la longitud de onda de Bragg ($\Delta\lambda_B$), el cual está directamente relacionado con cambios en temperatura (ΔT) o deformación ($\Delta\epsilon$) [3]. Una estimación precisa y robusta de la posición del *peak* es, por tanto, fundamental para garantizar la exactitud del sistema de medición.

Diversos algoritmos han sido desarrollados para extraer esta información del espectro reflejado, y pueden clasificarse según el enfoque utilizado para identificar el máximo espectral.

2.2.1 Métodos Directos

Los métodos directos se basan únicamente en el espectro de reflectividad medido para estimar la posición del *peak* espectral, sin realizar modelado ni ajuste funcional. Entre los más utilizados se encuentran:

- *X-dB bandwidth*

Este método, ampliamente empleado en sistemas de interrogación de FBG, consiste en definir un umbral de reflectividad R_{th} ubicado a X dB por debajo del valor máximo R_{max} , es decir:

$$R_{th} = R_{max} - X \text{ [dB]}$$

Como se ilustra en la figura 2.11, las longitudes de onda cuya reflectividad supera este umbral definen un intervalo espectral alrededor del *peak*. La longitud de onda de Bragg puede entonces estimarse como el centro de este intervalo:

$$\lambda_B = \frac{\lambda_{izq} + \lambda_{der}}{2}$$

donde λ_{izq} y λ_{der} son las longitudes de onda en los bordes del umbral. Este método es simple y rápido, pero sensible a ruido y asimetrías en el perfil espectral [9].

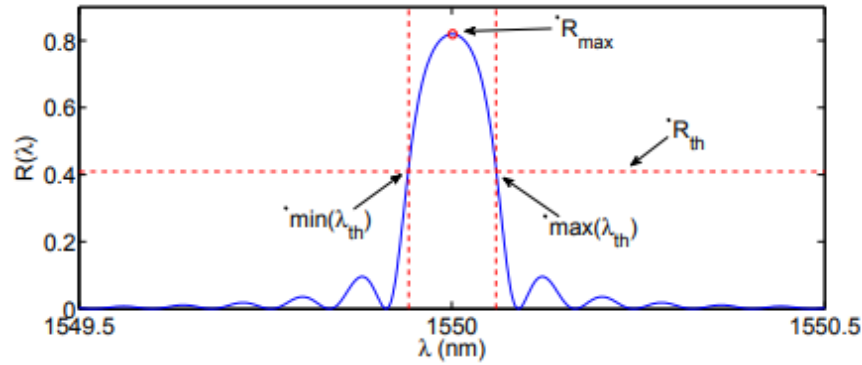


Figure 2.11: Detección del *peak* espectral mediante el método X-dB [3]

- *Centroide*

Este método calcula el centroide espectral del perfil de reflectividad, análogo al centro de masa, como una media ponderada por la amplitud del espectro:

$$\lambda_B = \frac{\sum_n \lambda_n R(\lambda_n)}{\sum_n R(\lambda_n)} \quad (2.10)$$

donde n recorre todas las muestras espectrales, λ_n es la longitud de onda de la muestra n -ésima, y $R(\lambda_n)$ es el valor de reflectividad correspondiente [9],[10]. A diferencia del método X-dB, el centroide utiliza toda la información espectral, lo que lo hace más robusto frente a ruido aleatorio, aunque puede sesgarse ante distorsiones asimétricas o fondo elevado. La figura 2.12 ilustra el principio de este método.

2.2.2 Métodos por ajuste de curva

Los métodos de ajuste de curva (o *curve fitting*) estiman la posición del *peak* espectral mediante la aproximación de un modelo funcional al entorno del máximo del espectro medido. Estos métodos mejoran la resolución respecto al muestreo discreto, siempre que el modelo elegido represente adecuadamente la forma espectral de la FBG [3].

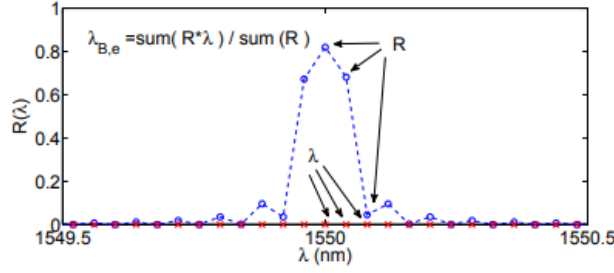


Figure 2.12: Estimación de λ_B mediante el método del centroide [3]

- *Ajuste Polinomial*

Este método consiste en ajustar un polinomio de grado reducido (generalmente cuadrático) a los puntos espectrales cercanos al máximo detectado. El caso más común es el ajuste parabólico, dado por [3]:

$$R(\lambda) = a_2\lambda^2 + a_1\lambda + a_0 \quad (2.11)$$

donde los coeficientes a_0 , a_1 y a_2 se determinan mediante regresión lineal sobre un conjunto de muestras espectrales alrededor del *peak* inicial. La longitud de onda de Bragg se estima entonces como el máximo de la parábola ajustada [9]:

$$\lambda_B = -\frac{a_1}{2a_2} \quad (2.12)$$

Este método es computacionalmente eficiente y adecuado para espectros con forma simétrica, aunque puede introducir errores si el perfil espectral presenta asimetría o ruido elevado [9].

- *Ajuste Gaussiano*

Este enfoque modela el perfil espectral de la FBG como una función gaussiana, adecuada para rejillas con respuesta simétrica y lóbulos laterales bajos. La función de ajuste es [3]:

$$R(\lambda) \approx A \cdot \exp\left(-\frac{(\lambda - \lambda_0)^2}{2\sigma^2}\right) \quad (2.13)$$

donde A es la amplitud, λ_0 la longitud de onda central (estimada como λ_B), y σ el ancho efectivo del *peak*. Debido a su naturaleza no lineal, los parámetros se estiman mediante algoritmos de regresión no lineal, como el de mínimos cuadrados iterativos (por ejemplo, Levenberg-Marquardt). Aunque más costoso computacionalmente, este método ofrece buena precisión cuando el espectro se aproxima a una forma gaussiana [9].

2.2.3 Métodos basados en transformadas

Los métodos basados en transformadas aprovechan representaciones en dominios alternativos, como el frecuencial o el de componentes ortogonales, para mejorar la detección del *peak* espectral en condiciones de bajo muestreo, ruido o resolución limitada. Estos enfoques permiten estimar desplazamientos sub-muestrales mediante propiedades matemáticas de las transformadas, sin requerir ajuste funcional ni interpolación explícita.

- *Fast Phase Correlation* (basado en comparación de fases)

Este método se fundamenta en la propiedad de desplazamiento de la transformada de Fourier, la cual establece que un corrimiento $\Delta\lambda$ en el dominio de la longitud de onda induce un desfase lineal proporcional en el dominio de la frecuencia espacial. En lugar de calcular la correlación cruzada completa mediante una transformada inversa, este enfoque compara directamente las fases de la transformada rápida de Fourier (FFT) de un espectro medido $R(\lambda)$ y uno de referencia $R_{\text{ref}}(\lambda)$ [24, 11].

Sean $F[k] = \mathcal{F}\{R(\lambda)\}$ y $F_{\text{ref}}[k] = \mathcal{F}\{R_{\text{ref}}(\lambda)\}$ las representaciones complejas de los espectros en el dominio frecuencial, donde k denota el índice del bin de frecuencia ($k = 0, 1, \dots, N-1$). Cada coeficiente $F[k]$ contiene información de magnitud y fase, expresable como $F[k] = |F[k]|e^{j\angle F[k]}$. Si ambos espectros difieren únicamente en un desplazamiento espectral $\Delta\lambda$, la teoría de Fourier dicta que la diferencia de fase entre sus componentes frecuenciales correspondientes es lineal respecto al índice k , dada por:

$$\angle F[k] - \angle F_{\text{ref}}[k] = -2\pi k \cdot \frac{\Delta\lambda}{N\delta\lambda} \quad (2.14)$$

donde $\angle F[k]$ y $\angle F_{\text{ref}}[k]$ representan la fase instantánea de la componente k -ésima para la medición y la referencia, respectivamente; N corresponde al número total de puntos en la FFT; $\delta\lambda$ es el espaciado espectral entre muestras adyacentes en el dominio de la longitud de onda; y $\Delta\lambda$ es el desplazamiento del *peak* a estimar.

Reordenando la ecuación (2.14), se obtiene una expresión directa para el desplazamiento espectral en función de la diferencia de fase:

$$\Delta\lambda = -\frac{\delta\lambda}{2\pi k} (\angle F[k] - \angle F_{\text{ref}}[k]) N \quad (2.15)$$

En la práctica, esta estimación no se realiza sobre un único bin de frecuencia, sino que se calcula sobre un rango de índices k donde la magnitud de la FFT es significativa (es decir, donde la relación señal-ruido es alta), promediando los resultados para mejorar la robustez frente al ruido instrumental [24]. Este método permite una estimación precisa de $\Delta\lambda$ con resolución sub-muestral sin necesidad de realizar la correlación cruzada inversa (IFFT), reduciendo la carga computacional. Por esta razón, es especialmente útil en sistemas de alta velocidad donde se requiere bajo costo computacional y baja latencia [11].

- *Transformada de Karhunen-Loève* (Análisis de Componentes Principales)

Esta técnica es particularmente útil en sistemas con bajo muestreo espectral o ruido elevado [25]. Consiste en descomponer un conjunto de espectros de FBG en una base ortogonal de funciones (modos de Karhunen-Loève), que capturan la mayor varianza posible con un número reducido de componentes. Aunque no se basa directamente en la FFT, el proceso puede incluir un paso de correlación espectral.

Dado un vector de muestra espectral $\mathbf{r}$, se construye una matriz de autocorrelación $\mathbf{M} = \mathbb{E}[\mathbf{r}\mathbf{r}^H]$, que puede tener estructura de Toeplitz si el proceso es estacionario. Esta matriz se descompone mediante:

$$\mathbf{M} = \mathbf{V}\mathbf{D}\mathbf{V}^{-1} \quad (2.16)$$

donde $\mathbf{D}$ es una matriz diagonal con los autovalores (que representan la energía de cada modo), y $\mathbf{V}$ contiene los autovectores (los modos espaciales). El desplazamiento del *peak* se infiere a partir del modo dominante o mediante proyección del espectro sobre los componentes principales. Este enfoque permite reconstruir y estimar λ_B incluso con señales submuestreadas o de baja relación señal-ruido.

2.2.4 Métodos basados en optimización

Estos métodos consisten en entrenar un modelo que optimice una función de costo para estimar la longitud de onda de Bragg (λ_B), utilizando un conjunto de espectros de referencia o datos de entrenamiento previamente caracterizados. La estimación se realiza mediante la minimización de un funcional que mide la discrepancia entre el espectro medido y una representación paramétrica o aprendida.

- *Redes Neuronales* (Aproximación con Funciones de Base Radial)

Para estimar λ_B mediante una red neuronal con funciones de base radial (RBF), el espectro reflejado $R(\lambda)$ se muestrea en N puntos y se aproxima mediante una combinación lineal de P funciones gaussianas [3, 26]:

$$R(\lambda) \approx F(\lambda) = \sum_{p=1}^P \omega_p \varphi(\lambda_p) \quad (2.17)$$

donde cada función de base radial está definida como:

$$\varphi(\lambda_p) = \exp \left(-\frac{(\lambda - \lambda_p^{(0)})^2}{2\sigma_p^2} \right)$$

con $\lambda_p^{(0)}$ la posición central y σ_p el ancho de la p -ésima función. Los pesos ω_p se determinan minimizando una función de costo regularizada [9, 26]:

$$J = \sum_{n=1}^N [F(\lambda_n) - R(\lambda_n)]^2 + \gamma \sum_{p=1}^P \omega_p^2 \quad (2.18)$$

donde el primer término mide el error de ajuste sobre las N muestras espectrales, y el segundo introduce regularización tipo Tikhonov (o L_2) mediante el parámetro $\gamma > 0$, que evita sobreajustes. Una vez entrenada la red, la posición del *peak* puede inferirse a partir de la combinación resultante o mediante una capa de salida adicional que predice directamente λ_B .

- *Gaussian Process Regression* (GPR)

La regresión basada en procesos gaussianos (GPR) es un método no paramétrico de aprendizaje supervisado ampliamente utilizado en tareas de regresión con cuantificación de incertidumbre [27]. En lugar de ajustar una única función al conjunto de datos, GPR modela la distribución sobre funciones que podrían explicar las observaciones, asumiendo que cualquier conjunto finito de muestras sigue una distribución gaussiana multivariada.

Dado un conjunto de espectros de entrenamiento $\{R_i(\lambda), \lambda_{B,i}\}$, GPR aprende una relación probabilística entre el perfil espectral y λ_B . Para un nuevo espectro $R_{\text{nuevo}}(\lambda)$, el modelo predice no solo el valor esperado de λ_B , sino también su varianza (incertidumbre). Esto lo hace particularmente útil en aplicaciones donde se requiere conocer la confiabilidad de la estimación.

La predicción se basa en una función de covarianza (o kernel), como el kernel RBF:

$$k(\lambda, \lambda') = \sigma_f^2 \exp \left(-\frac{(\lambda - \lambda')^2}{2\ell^2} \right)$$

que define la similitud entre longitudes de onda. GPR no requiere un modelo explícito del espectro y puede generalizar bien incluso con pocos datos de entrenamiento. Sin embargo, su costo computacional presenta dos componentes diferenciadas [27]:

- **Entrenamiento:** Requiere la inversión de la matriz de covarianza de tamaño $N \times N$, con complejidad $O(N^3)$, donde N es el número de muestras de entrenamiento.
- **Evaluación/Predicción:** Para inferir λ_B en un nuevo espectro, la complejidad es $O(N^2)$ para el cálculo de la media predictiva, y $O(N^3)$ si se requiere también la varianza completa.

Esta escalabilidad cúbica en la etapa de entrenamiento limita su aplicabilidad en sistemas de alta velocidad o con grandes volúmenes de datos, a menos que se empleen aproximaciones como procesos gaussianos dispersos (*sparse GPs*) [27].

2.2.5 Algoritmos basados en correlación

Los métodos basados en correlación cruzada estiman el desplazamiento del *peak* espectral entre un espectro de referencia $R_{\text{ref}}(\lambda)$ y uno medido $R(\lambda)$ mediante el cálculo de su similitud como función del corrimiento. El desplazamiento que maximiza esta correlación corresponde al cambio en la longitud de onda de Bragg ($\Delta\lambda_B$), lo cual permite inferir variaciones de temperatura o deformación.

- Wavelength-Shifted Mutual Correlation

Este método calcula la correlación entre un espectro de referencia $R_{\text{ref}}(\lambda)$ (obtenido en condiciones conocidas de temperatura y deformación) y una versión desplazada del espectro medido, evaluada en $\lambda - m \cdot \delta\lambda$. La correlación se define como[28]:

$$P_m = \sum_{\lambda} R_{\text{ref}}(\lambda) \cdot R(\lambda - m \cdot \delta\lambda) \quad (2.19)$$

donde m es un índice entero que indica el número de pasos de desplazamiento, y $\delta\lambda$ es el espaciado espectral entre muestras adyacentes del espectrómetro. El cálculo se repite para distintos valores de m , generando una función de correlación discreta P_m . El desplazamiento estimado corresponde al valor que maximiza P_m :

$$\Delta\lambda_B = \delta\lambda \cdot \arg \max_m P_m \quad (2.20)$$

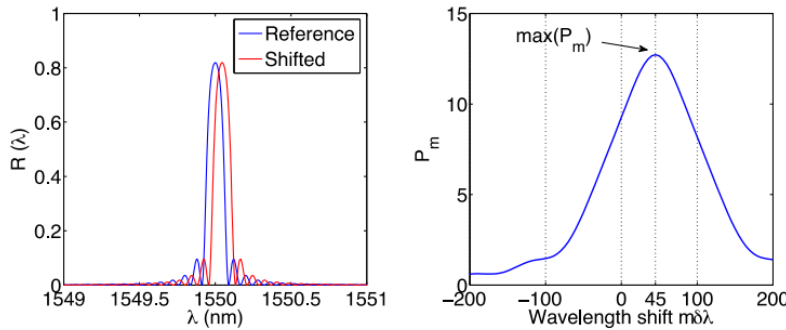


Figure 2.13: Implementación del método de correlación cruzada. A la izquierda: espectro de referencia y espectro desplazado en 5 pm. A la derecha: función de correlación mostrando un corrimiento detectado de 45 pm [3].

Además, gracias a la disponibilidad de librerías optimizadas (como FFTW o funciones de correlación en Python/MATLAB), la implementación es sencilla y con bajo tiempo de procesamiento, lo que lo hace adecuado para sistemas de interrogación en tiempo real.

- **Correlation Polynomial Fit**

Este método combina la robustez de la correlación cruzada con la precisión sub-muestral del ajuste polinomial. Primero, se calcula la función de correlación $P(\lambda)$ (mediante FFT o desplazamiento directo). Luego, en torno al máximo discreto, se ajusta un polinomio cuadrático:

$$P(\lambda) \approx a_2\lambda^2 + a_1\lambda + a_0 \quad (2.21)$$

La posición del *peak* refinado se estima como:

$$\lambda_{\text{peak}} = -\frac{a_1}{2a_2} \quad (2.22)$$

Este enfoque mejora la resolución más allá del espaciado $\delta\lambda$, alcanzando precisión sub-muestral. Aunque aumenta ligeramente el costo computacional, sigue siendo adecuado para aplicaciones en tiempo real. Como se muestra en la figura 2.14, el resultado es más preciso que el método puramente discreto.

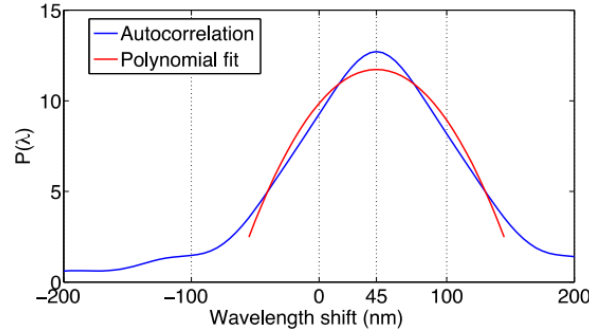


Figure 2.14: Esquema de implementación del método *Correlation Polynomial Fit*, comparado con el método *Wavelength-Shifted* [3].

- **Cross-Correlation mediante FFT**

Este método es equivalente al enfoque *Wavelength-Shifted* en el dominio espacial, pero aprovecha el *Teorema de la Correlación Cruzada* para calcular la función de similitud $P(\lambda)$ de forma eficiente utilizando la Transformada Rápida de Fourier (FFT). En lugar de evaluar la sumatoria discreta P_m para cada desplazamiento m individualmente, se calcula directamente la correlación completa en el dominio de la frecuencia, reduciendo drásticamente la carga computacional [11, 29].

Matemáticamente, la correlación cruzada discreta entre el espectro de referencia $R_{\text{ref}}[n]$ y el espectro medido $R[n]$ se define como:

$$P[m] = \sum_{n=-\infty}^{\infty} R_{\text{ref}}[n] \cdot R[n - m] \quad (2.23)$$

donde m representa el lag o desplazamiento entre muestras. Según el teorema de correlación, esta operación en el dominio del tiempo es equivalente a la multiplicación compleja en el dominio de la frecuencia, donde el espectro de una de las señales se conjuga [11]:

$$P(\lambda) = \mathcal{F}^{-1} \{ \mathcal{F}\{R_{\text{ref}}(\lambda)\} \cdot \mathcal{F}\{R(\lambda)\}^* \} \quad (2.24)$$

donde $\mathcal{F}\{\cdot\}$ denota la operación FFT, $*$ indica el conjugado complejo y $\mathcal{F}^{-1}$ es la transformada inversa (IFFT). El desplazamiento espectral estimado corresponde al índice del máximo absoluto de la función de correlación resultante:

$$\Delta\lambda_B = \delta\lambda \cdot \arg \max_m P[m] \quad (2.25)$$

con $\delta\lambda$ siendo el espaciado espectral del interrogador.

Es crucial considerar que la implementación mediante FFT calcula inherentemente una *correlación circular*, asumiendo que las señales son periódicas. Para obtener una *correlación lineal* correcta, equivalente a la definición de la ecuación (2.23), es necesario aplicar *zero-padding* a las señales antes de la transformada, extendiendo su longitud a $N \geq N_{\text{ref}} + N_{\text{med}} - 1$. Sin este paso, se producen artefactos de solapamiento (*time-aliasing*) que distorsionan la posición del peak [29, 25].

Desde la perspectiva de complejidad computacional, este enfoque reduce el costo de $O(N^2)$ del método directo a $O(N \log N)$, lo que lo hace viable para sistemas en tiempo real con alta tasa de muestreo [25]. Además, la correlación cruzada actúa como un filtro adaptado que maximiza la relación señal-ruido (SNR) en la salida, proporcionando robustez frente al ruido blanco aditivo presente en la adquisición óptica [3, 24]. Sin embargo, la resolución del desplazamiento estimado está limitada inicialmente al espaciado discreto de las muestras ($\delta\lambda$), requiriendo técnicas de interpolación o ajuste polinomial posterior para alcanzar precisión sub-muestral, como se detalla en métodos derivados [3].

2.2.6 Comparación y elección del método

Para seleccionar el método más adecuado para la detección del *peak* espectral en sensores FBG, se evalúan los algoritmos revisados según criterios clave: precisión, robustez al ruido, costo computacional, capacidad de resolución sub-muestral y facilidad de implementación. La Tabla 2.1 resume esta comparación cualitativa.

Table 2.1: Comparación de métodos de detección de peak espectral en FBG

Método	Precisión	Robustez al ruido	Costo computacional
Método directo (X-dB)	Baja	Media	Muy bajo
Centroide	Media	Media	Bajo
Ajuste polinomial	Media-Alta	Media	Bajo
Ajuste gaussiano	Alta	Media-Alta	Medio
Fast Phase Correlation	Alta	Alta	Medio
Karhunen-Loève	Alta	Alta	Alto
Redes neuronales (RBF)	Alta	Alta	Alto
Gaussian Process Regression	Alta	Alta	Alto
Wavelength-Shifted Correlation	Alta	Alta	Alto
Cross-Correlation (FFT)	Alta	Alta	Medio-Bajo
Correlation Polynomial Fit	Muy Alta	Alta	Medio

La ventaja computacional de la correlación cruzada mediante FFT sobre el método directo se fundamenta en la reducción de la complejidad algorítmica. La correlación cruzada directa en el dominio del tiempo requiere calcular la sumatoria discreta para cada desplazamiento m :

$$P[m] = \sum_{n=-\infty}^{\infty} R_{\text{ref}}[n] \cdot R[n-m], \quad m = 0, 1, \dots, N-1 \quad (2.26)$$

donde N es el número de muestras espectrales. Para cada uno de los N desplazamientos posibles, se realizan N multiplicaciones y sumas, resultando en una complejidad computacional de $O(N^2)$ [11, 29]. Para un espectro típico de 10 000 puntos, esto implica aproximadamente 10^8 operaciones por correlación.

En contraste, el método basado en FFT aprovecha el teorema de la correlación cruzada para transformar la operación al dominio frecuencial:

$$P(\lambda) = \mathcal{F}^{-1} \{ \mathcal{F}\{R_{\text{ref}}(\lambda)\} \cdot \mathcal{F}\{R(\lambda)\}^* \} \quad (2.27)$$

La transformada rápida de Fourier (FFT) calcula la DFT con una complejidad de $O(N \log_2 N)$ mediante algoritmos de divide y vencerás (radix-2) [25]. El proceso completo requiere: (i) dos FFT directas ($2 \cdot O(N \log N)$), (ii) una multiplicación compleja punto a punto ($O(N)$), y (iii) una FFT inversa ($O(N \log N)$). La complejidad total permanece en $O(N \log N)$ [11, 29].

Para cuantificar la mejora, considérese un espectro con $N = 10\,000$ muestras. El método directo requiere aproximadamente $N^2 = 10^8$ operaciones, mientras que la FFT requiere $N \log_2 N \approx 10\,000 \cdot 13.3 \approx 1.33 \cdot 10^5$ operaciones. Esto representa un factor de reducción teórico de:

$$\frac{O(N^2)}{O(N \log N)} = \frac{N}{\log_2 N} \approx \frac{10\,000}{13.3} \approx 752 \quad (2.28)$$

Este análisis teórico explica por qué la implementación mediante FFT es órdenes de magnitud más rápida que la correlación directa, validando su selección para aplicaciones en tiempo real donde la latencia es crítica [3, 24].

Cross-Correlation implementada mediante la transformada rápida de Fourier (FFT) se destaca como una opción óptima por su “equilibrio entre precisión, robustez y eficiencia computacional”. A diferencia del método Wavelength-Shifted, que requiere múltiples evaluaciones en el dominio del tiempo (costo $O(N^2)$), la versión basada en FFT aprovecha el teorema de la correlación cruzada

para calcular la función completa en $O(N \log N)$, reduciendo significativamente el tiempo de procesamiento.

Este enfoque es altamente robusto frente al ruido, ya que la correlación cruzada realza las componentes espectrales coherentes y suprime las aleatorias. Además, permite alcanzar resolución sub-muestral mediante *zero-padding* o interpolación espectral antes de la FFT, sin necesidad de ajustes iterativos complejos.

Al combinar este método con un *polynomial fit* cuadrático en torno al pico de correlación (método Correlation Polynomial Fit), se mejora aún más la precisión de estimación de $\Delta\lambda_B$. Esta combinación ofrece: alta precisión y repetibilidad, bajo tiempo de procesamiento (adecuado para sistemas en tiempo real), facilidad de implementación mediante librerías estándar (MATLAB) y robustez frente a variaciones de SNR.

Por estas razones, se selecciona la correlación cruzada implementada mediante FFT como método base, integrada con técnicas de ajuste polinomial cuadrático para refinamiento sub-muestral y zero-padding como estrategia opcional de sobremuestreo espectral. Esta configuración constituye el enfoque principal para la detección del peak espectral en esta memoria, validado por su equilibrio entre precisión ($\text{RMS} < 0.8^\circ\text{C}$) y eficiencia computacional ($O(N \log N)$) [3, 24]. Su compatibilidad con hardware de procesamiento rápido, incluyendo FPGA, GPU y microcontroladores con DSP, lo hace ideal para aplicaciones prácticas en monitoreo estructural, sensores distribuidos y sistemas de telecomunicaciones, tal como se demuestra en la implementación embebida evaluada en el Capítulo 4 [27, 28].

Capítulo 3

Implementación y Resultados

3.1 Obtención de datos e implementación del algoritmo

3.1.1 Implementación de los algoritmos de correlación cruzada

Para la detección precisa del desplazamiento espectral en sensores FBG, se implementaron cuatro algoritmos de correlación cruzada con el fin de comparar su desempeño en términos de precisión, robustez y eficiencia computacional. Uno de ellos corresponde al uso de la función `xcorr` de MATLAB, ampliamente utilizada en procesamiento de señales [30]. Esta función calcula la correlación cruzada entre dos vectores de entrada y devuelve un vector resultante cuyo máximo indica el desplazamiento relativo entre ellos. Sin embargo, como señala [12], el proceso interno no es completamente transparente en cuanto al manejo del *zero-padding*, el escalado y la estrategia de muestreo, lo que motiva el desarrollo de implementaciones propias para validar y controlar cada etapa del cálculo.

Para la detección precisa del desplazamiento espectral en sensores FBG, se implementan cinco algoritmos de correlación cruzada con el fin de comparar su desempeño en términos de precisión, robustez y eficiencia computacional. Uno de ellos corresponde al uso de la función `xcorr` de MATLAB, ampliamente utilizada en procesamiento de señales [31]. Esta función calcula la correlación cruzada entre dos vectores de entrada y devuelve un vector resultante cuyo máximo indica el desplazamiento relativo entre ellos. Sin embargo, como señala [29], el proceso interno no es completamente transparente en cuanto al manejo del *zero-padding*, el escalado y la estrategia de muestreo, lo que motiva el desarrollo de implementaciones propias para validar y controlar cada etapa del cálculo.

Los otros cuatro algoritmos se desarrollan completamente en este trabajo y están basados en las formulaciones teóricas presentadas en la Sección 2.2.5. Estas implementaciones personalizadas incluyen:

1. Correlación cruzada mediante FFT estándar.
2. Correlación cruzada mediante FFT con *zero-padding* a potencia de dos.
3. Correlación cruzada directa.
4. Correlación cruzada mediante FFT con ajuste polinomial.

Esta distinción permite evaluar no solo la eficiencia de la transformada, sino también el impacto del refinamiento numérico en la precisión final de la estimación de λ_B .

Todos los algoritmos de detección del *peak* espectral descritos requieren dos tipos de señales como entrada, un “espectro de referencia” R_{ref} , adquirido bajo condiciones conocidas (por ejemplo, temperatura y deformación base y una serie de **espectros medidos** $R(t)$, obtenidos durante el experimento, que contienen el desplazamiento espectral inducido por cambios físicos. Para calibración de los métodos, también se necesita la temperatura asociada a cada espectro.

Cada espectro corresponde a una medición en el dominio de la longitud de onda, con n puntos espectrales muestreados en un rango específico. Por lo tanto, cada señal de entrada se representa como un vector fila de dimensión $1 \times n$: $R = [R(\lambda_1) \ R(\lambda_2) \ \cdots \ R(\lambda_n)]$ donde $R(\lambda_i)$ es el valor de potencia reflejada en la longitud de onda λ_i . Durante un experimento dinámico, se adquieren m espectros consecutivos a lo largo del tiempo, formando una matriz de datos de tamaño $m \times n$, donde cada fila corresponde a una medición temporal y cada columna a una longitud de onda fija. Esta estructura permite procesar secuencialmente cada espectro frente al espectro de referencia, obteniendo un vector de desplazamientos $\Delta\lambda_B(t)$ de tamaño $m \times 1$, que representa la evolución del *peak* espectral en el tiempo.

La detección del *peak* final se realiza identificando el índice del máximo absoluto en el vector de correlación, el cual entrega la información de cual fue el corrimiento entre las muestras. Los resultados de los cinco algoritmos serán comparados en términos de precisión y tiempo de ejecución, utilizando como referencia la diferencia espectral de la referencia y la muestra, con el fin de establecer la relación entre el desplazamiento espectral detectado ($\Delta\lambda_B$) y el cambio físico real (ΔT).

Antes de aplicar cualquier algoritmo de detección del *peak* espectral, los datos adquiridos deben ser sometidos a un proceso de preprocesamiento para garantizar su compatibilidad con el cálculo de correlación cruzada. Como se describe en la Sección 2.2.5, los espectros reflejados obtenidos mediante interrogadores comerciales de FBG suelen entregarse en escala logarítmica. Dado que el algoritmo de correlación cruzada opera bajo el principio de superposición lineal [28, 12], requiere que las amplitudes sean lineales. La representación logarítmica, al ser una transformación no lineal, comprime el rango dinámico de la señal y altera la relación proporcional entre amplitudes, lo que puede sesgar la estimación del desplazamiento espectral, especialmente en regiones de baja potencia reflejada como los lóbulos laterales o el piso de ruido.

Por lo tanto, es necesario convertir los datos a escala lineal antes del procesamiento. La transformación inversa se obtiene directamente de la ecuación (3.1):

$$R_{\text{lin}}(\lambda) = 10^{R_{\text{dBm}}(\lambda)/10} \quad (3.1)$$

donde $R_{\text{lin}}(\lambda)$ representa la potencia reflejada en unidades lineales proporcionales a mW. Este paso asegura que la correlación cruzada se calcule sobre una representación físicamente consistente del espectro, preservando las relaciones de amplitud necesarias para una estimación precisa de $\Delta\lambda_B$ [3, 28]. Además, evita errores sistemáticos derivados de la compresión logarítmica en zonas críticas del perfil espectral.

En el Código 3.1 uso la función `xcorr` de MATLAB para calcular la correlación cruzada entre dos señales espectrales: el espectro de referencia $R_{\text{ref}}(\lambda)$ y el espectro medido $R(\lambda)$. Esta función devuelve directamente el vector de correlación P , cuyo máximo indica el grado de similitud entre las señales en función del desplazamiento relativo.

Un paso clave para todas las implementaciones es la generación del vector de *lags*, ℓ , que representa los posibles desplazamientos (en muestras) entre los dos espectros. Aunque `xcorr` puede devolver este vector opcionalmente, en este caso se calcula explícitamente para asegurar compatibilidad con

el resto de los algoritmos y permitir una interpretación física directa del resultado. El valor de ℓ asociado al máximo de P indica el corrimiento del *peak* espectral, y se convierte a unidades de longitud de onda mediante $\delta\lambda$, el espaciado espectral entre muestras.

```

1 function [lags, correlation] = correlacion_xcorr(signal1, signal2)
2     %Proceso de correlacion
3     [correlation, lags] = xcorr(signal1, signal2);
4 end

```

Código 3.1: Uso de correlación cruzada mediante `xcorr`

Los Códigos 3.2 , 3.3 y 3.4 muestran 3 implementaciones del algoritmo de correlación cruzada mediante transformada rápida de Fourier (FFT), siguiendo el proceso descrito en la sección 2.2.5 y la ecuación (2.24). Estos métodos aprovechan el teorema de la correlación cruzada para calcular $P(\lambda)$ en el dominio de Fourier teóricamente, reduciendo la complejidad computacional respecto al cálculo directo en el dominio espectral.

La diferencia radica en el manejo del *zero-padding*. el Código 3.3 extiende el zero-padding hasta la próxima potencia de 2 mayor o igual a $2N-1$, es decir, $M = 2^k \geq 2N-1$. Esta estrategia aprovecha la eficiencia del algoritmo FFT *radix-2*, esperando reducir el tiempo de cómputo. Además, el mayor número de puntos en la FFT mejora la resolución espectral efectiva en el dominio de la correlación (*oversampling*), lo que permite una estimación sub-muestral del *peak* sin necesidad de interpolación adicional.

Por otra parte, el Código 3.4 no incorpora zero-padding, pero agrega una etapa de post-procesamiento para ajustar el resultado de la correlación mediante un polinomio cuadrático. Este método utiliza una ventana de 5 puntos centrada en el máximo discreto de la función de correlación para interpolar la posición real del *peak*. De esta manera, se consigue una mayor precisión en la estimación del desplazamiento espectral debido a que permite resolver la posición del máximo continuo subyacente con una resolución inferior al espaciado espectral discreto $\delta\lambda$, tal como se fundamenta en la Sección 2.2.5. Este acercamiento ofrece un equilibrio favorable entre precisión sub-muestral y carga computacional, evitando el incremento en el tamaño de la FFT que requiere el *zero-padding*. Estas implementaciones permite evaluar el trade-off entre eficiencia computacional y precisión espectral, siendo clave para determinar la configuración óptima en sistemas de interrogación en tiempo real.

```

1 function [lags, correlation] = correlacion_fft(signal1, signal2)
2     %Transformacion de la entradas
3     fft_signal1 = fft(signal1);
4     fft_signal2 = fft(signal2);
5     %Proceso de correlacion
6     fft_correlation = fft_signal1 .* conj(fft_signal2);
7     %Transformacion inversa del resultado
8     correlation = ifft(fft_correlation);
9     correlation = fftshift(correlation);
10    correlation = correlation';
11    lags = (-N/2:N/2-1);
12 end

```

Código 3.2: Implementación de correlación cruzada mediante FFT sin zero-padding

```

1 function [lags, correlation] = correlacion_fft_zero_padding(signal1,
2 signal2)
3     len1 = length(signal1); len2 = length(signal2);
4     N = 2^nextpow2(max(len1, len2)); %Busqueda del tama o optimo para
5     radix-2
6     %Oversampling de las entradas
7     signal1_padded = [signal1; zeros(N - len1, 1)];
8     signal2_padded = [signal2; zeros(N - len2, 1)];
9     %transformacion de la entradas
10    fft_signal1 = fft(signal1_padded);
11    fft_signal2 = fft(signal2_padded);
12    %Proceso de correlacion
13    fft_correlation = fft_signal1 .* conj(fft_signal2);
14    %Transformacion inversa del resultado
15    correlation = ifft(fft_correlation);
16    correlation = fftshift(correlation);
17    lags = (-N/2:N/2-1);
18 end

```

Código 3.3: Implementación de correlación cruzada mediante FFT con zero-padding a potencia de 2

```

1 function [lambda_peak, correlation, lags] = correlacion_fft_polynomial(
2 signal1, signal2)
3     M = length(signal1);
4
5     % FFT de ambas se ales
6     fft_signal1 = fft(signal1);
7     fft_signal2 = fft(signal2);
8
9     % Correlacion (teorema de correlacion cruzada)
10    fft_correlation = fft_signal1 .* conj(fft_signal2);
11
12    % Transformada inversa
13    correlation = ifft(fft_correlation);
14    correlation = fftshift(correlation);

```

```

15     correlation = real(correlation);
16
17     %Generar vector de lags
18     lags = (-M/2 : M/2 - 1);
19
20     [~, max_idx] = max(correlation);
21     %Ajuste polinomial cuadrático alrededor del máximo
22     % Seleccionar ventana de 5 puntos alrededor del máximo
23     window_size = 5;
24     half_window = floor(window_size / 2);
25
26     % Índices de la ventana (con protección de bordes)
27     idx_start = max(1, max_idx - half_window);
28     idx_end = min(length(correlation), max_idx + half_window);
29
30     % Extraer puntos para el ajuste
31     x_window = lags(idx_start:idx_end);
32     y_window = correlation(idx_start:idx_end);
33
34     % Ajuste polinomial de grado 2 (parabola)
35     p = polyfit(x_window, y_window, 2);
36
37     % Coeficientes: p(1) = a2, p(2) = a1, p(3) = a0
38     a2 = p(1);
39     a1 = p(2);
40     a0 = p(3);
41
42     % El máximo de la parabola está en: lambda = -a1 / (2*a2)
43     if a2 ~= 0
44         lag_refinado = -a1 / (2 * a2);
45     else
46         lag_refinado = lags(max_idx);
47     end
48
49     % Convertir a unidades de longitud de onda
50     lambda_peak = abs(lag_refinado) ;
51
52 end

```

Código 3.4: Implementación de correlación cruzada mediante FFT con ajuste polinomial

Finalmente, el Código 3.5 implementa la correlación cruzada directa en el dominio espectral según la ecuación (2.19). Este enfoque opera directamente en el dominio del desplazamiento, sin recurrir a transformadas intermedias.

A diferencia de los métodos basados en FFT, este algoritmo no aprovecha la eficiencia del dominio de Fourier, lo que resulta en una complejidad computacional de $O(N^2)$, siendo significativamente más lento para señales de gran longitud. Sin embargo, su implementación es directa y transparente, sin efectos de muestreo o zero-padding que puedan introducir artefactos numéricos. Por ello, se utiliza como referencia teórica para validar los resultados obtenidos con los métodos en el dominio de Fourier.

Este método permite una comprensión clara del principio físico de la correlación cruzada, aunque su alto costo computacional lo hace poco adecuado para aplicaciones en tiempo real.

```

1 function [lags, correlation] = correlacion_directa(signal1, signal2)
2     %Busqueda de cantidad de operaciones para la correlacion
3     N1 = length(signal1); N2 = length(signal2);
4     lags = -(N2-1):(N1-1);
5
6     % Inicializar el vector de correlacion
7     correlation = zeros(1, length(lags));
8
9     % Calcular la correlacion cruzada directamente mediante sumatoria
10    for n = 1:length(lags)
11        lag = lags(n);
12        sum_val = 0;
13
14        % Determinar los lmites de la sumatoria
15        if lag >= 0
16            % signal1 se adelanta a signal2
17            start_idx = lag + 1;
18            end_idx = min(N1, N2 + lag);
19            for k = start_idx:end_idx
20                sum_val = sum_val + signal1(k) * signal2(k - lag);
21            end
22        else
23            % signal2 se adelanta a signal1
24            start_idx = 1;
25            end_idx = min(N1 + lag, N2);
26            for k = start_idx:end_idx
27                sum_val = sum_val + signal1(k - lag) * signal2(k);
28            end
29        end
30
31        correlation(n) = sum_val;
32    end
33 end

```

Código 3.5: Implementación de correlación cruzada directa en el dominio espectral de las FBGs

3.1.2 Entorno de ejecución

Todas las pruebas y análisis presentados en esta memoria fueron realizadas en un sistema con las siguientes especificaciones técnicas. El hardware consiste en una computadora personal equipada con un procesador Intel Core i5-12400F (6 núcleos, 12 hilos), 32 GB de memoria RAM DDR4 y un disco sólido M.2 NVMe de 1 TB, lo cual proporciona un entorno potente y estable para la adquisición, procesamiento y almacenamiento de grandes volúmenes de datos espectrales.

Las implementaciones en MATLAB se ejecutaron sobre Windows 11, utilizando MATLAB R2025a. Por otro lado, las pruebas de desempeño y compilación del código en C se realizaron utilizando el Subsistema de Windows para Linux (WSL), con distribución Ubuntu 22.04 LTS. Este entorno permite una integración eficiente entre las herramientas de desarrollo en Windows y el entorno de compilación tipo Unix, ideal para evaluar el rendimiento de código embebido.

Estas condiciones se indican para facilitar la replicación de los experimentos y asegurar la transparencia en los tiempos de ejecución reportados.

3.1.3 Adquisición de datos

Los datos experimentales fueron adquiridos utilizando el montaje mostrado en la Figura 3.1, que consiste en un sensor FBG inmerso en un vaso precipitado con agua cuya temperatura se eleva y se deja enfriar, permitiendo estudiar la respuesta espectral frente a cambios térmicos. Además se tiene un atenuador óptico variable (VOA, *Variable Optical Attenuator*) y un interrogador comercial de rejillas de Bragg basado en interferometría de barrido de luz: el modelo *Hyperion Luna S155*. Este sistema permite medir el espectro reflejado por una FBG conectada directamente al puerto de entrada del interrogador (explicado en la sección 2.1.6) con alta resolución y estabilidad.

El interrogador *Luna S155* opera mediante un láser sintonizable que escanea el rango espectral de 1500 nm a 1600 nm, cubriendo todo el espectro de interés para sensores FBG en banda C y L. Este presenta una resolución espectral de 1 pm (capacidad del sistema para distinguir cambios finos en λ_B), frecuencia máxima de interrogación de 1 kHz y precisión de longitud de onda: típicamente < 1 pm con calibración interna.

Los datos se adquirieron mediante el software *ENLIGHT*, cuya interfaz se muestra en la Figura 3.2, donde se permite visualizar espectros en tiempo real y exportar la información. Es de aclarar que el interrogador exportada en formato `.txt` archivos que presentan una resolución espectral de 10 pm. La frecuencia de adquisición utilizada fue de 10 Hz, adecuada para monitoreo térmico lento, y cada medición corresponde a un espectro completo en el rango mencionado. Estos datos son el punto de partida para el preprocesamiento y la detección del *peak* mediante los algoritmos descritos previamente.

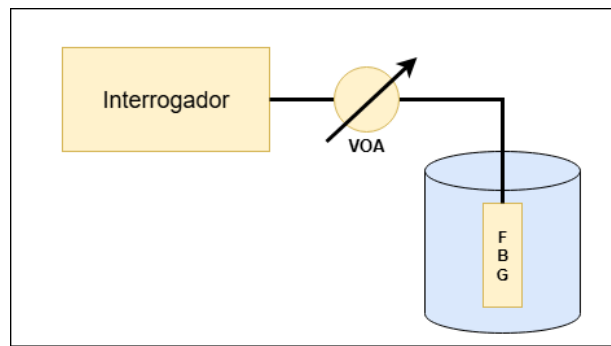


Figure 3.1: Esquema del montaje experimental para la adquisición de datos. Incluye interrogador Luna S155, VOA, FBG en vaso con agua caliente.

El atenuador óptico variable (VOA, *Variable Optical Attenuator*) no solo cumple una función de acondicionamiento del nivel de potencia óptica, sino que se utiliza activamente como herramienta para evaluar la robustez del algoritmo de correlación cruzada basado en FFT frente a diferentes niveles de relación señal-ruido (SNR). Al ajustar la atenuación introducida por el VOA, es posible reducir deliberadamente la potencia reflejada por la FBG antes de que llegue al detector del interrogador, simulando condiciones de baja SNR, como las que podrían presentarse en sistemas de fibra larga o con múltiples rejillas multiplexadas.

El VOA utilizado presenta una pérdida de inserción base de 0.80 dB, es decir, incluso en su configuración de mínima atenuación, introduce una pérdida inherente al dispositivo. Este valor fue determinado mediante una medición de referencia: se midió la potencia de un láser en la entrada del VOA (Figura 3.3-(a)) y en la salida con el atenuador en su configuración mínima (Figura 3.3-(b)). La diferencia entre ambas lecturas confirma la pérdida de inserción del dispositivo.

El sensor FBG se sumerge en agua debido a que este líquido actúa como un eficaz *buffer* térmico,

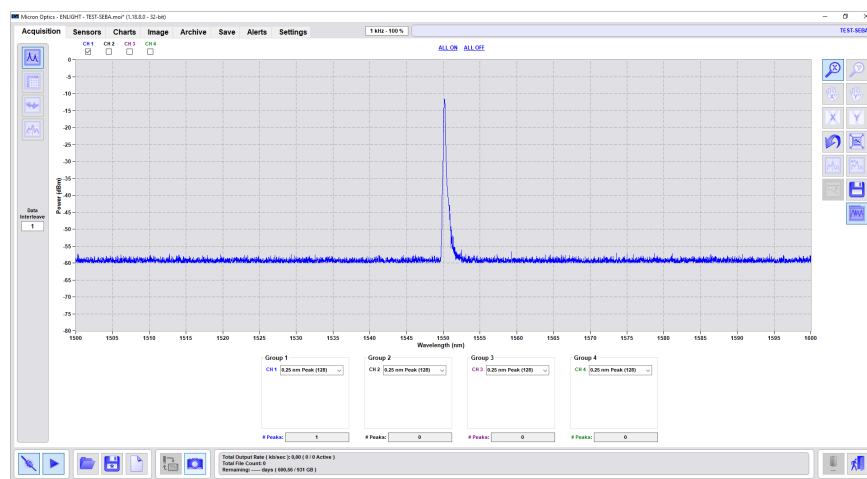


Figure 3.2: Interfaz del software ENLIGHT utilizado para la adquisición y visualización de espectros FBG. Permite exportar datos en formato `.txt`.

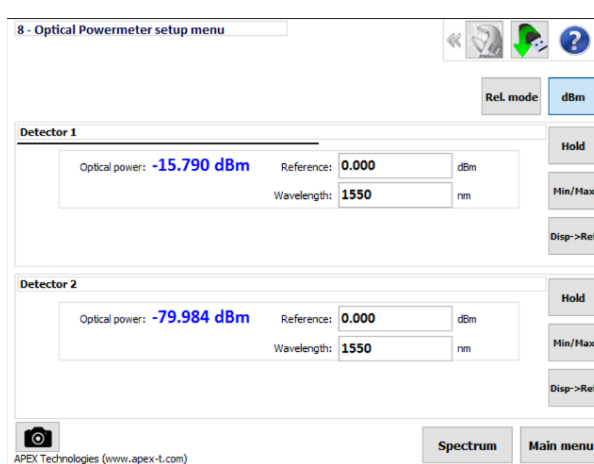
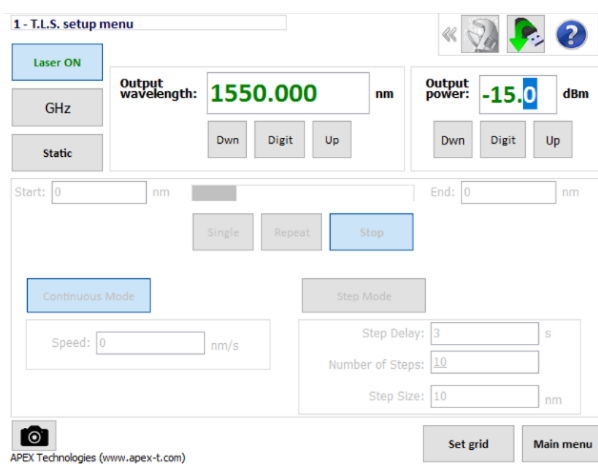


Figure 3.3: Medición de potencia óptica en la entrada del VOA. Se registra el nivel de señal antes de la atenuación.(a)Medición de potencia óptica en la salida del VOA con atenuación mínima. Muestra la pérdida base de 0.80 dB del dispositivo. (b)

gracias a su alta capacidad calorífica específica. Esta propiedad permite una transferencia de calor más uniforme y una mayor estabilidad térmica en comparación con el aire, reduciendo las fluctuaciones rápidas de temperatura que podrían interferir en las mediciones espectrales.

Las mediciones consisten en calentar el agua hasta los 75 grados celsius aproximadamente y luego dejar que se enfríe naturalmente hacia la temperatura ambiente. Durante este proceso, la FBG registra el desplazamiento gradual de su longitud de onda de Bragg (λ_B) como respuesta al cambio térmico. Este enfoque permite obtener una variación continua y controlada de temperatura, ideal para evaluar la sensibilidad del sensor y la precisión del algoritmo de detección del *peak* espectral.

Las mediciones adquiridas del sensor FBG consisten en espectros de reflectividad que representan la potencia óptica reflejada como función de la longitud de onda. Cada espectro corresponde a una condición térmica específica durante el proceso de enfriamiento del agua.

Como se mencionó previamente (sección 3.1), el interrogador registra las reflexiones espectrales de la FBG en escala logarítmica (dBm), como se muestra en la Figura 3.4-(a), esta representación es común en instrumentación óptica debido al amplio rango dinámico de las señales, pero no es adecuada directamente para el cálculo de correlación cruzada, ya que distorsiona la forma espectral en zonas de baja potencia reflejada. El resultado de esta transformación se ilustra en la Figura 3.4-(b), mostrando la misma medición ahora en escala lineal, es decir, en mW de potencia óptica real, lo cual es necesario para la detección del *peak* mediante correlación cruzada como se explica en la sección 3.1.1.

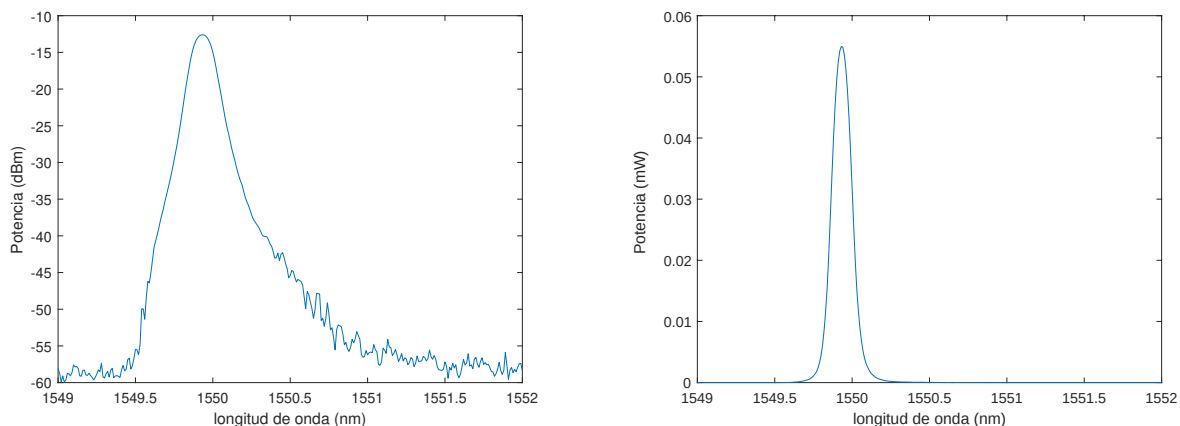


Figure 3.4: Comparación del espectro reflejado por la FBG antes y después de la conversión de escala. (a) Espectro original en escala logarítmica. (b) Espectro en escala lineal

Experimento 1: Enfriamiento medido del sensor FBG

La Figura 3.5 muestra una selección representativa de los espectros de potencia reflejada adquiridos durante el Experimento 1, en el cual el sensor FBG fue sometido a un proceso de enfriamiento pasivo desde aproximadamente 73°C hasta 48°C. La adquisición se realizó durante un intervalo total de 1000 s, con una frecuencia de muestreo de 10 Hz, lo que resultó en un total de 10 000 espectros registrados (1000 s $\times$ 10 Hz = 10 000 muestras).

Estos espectros ilustran el corrimiento progresivo del *peak* espectral hacia longitudes de onda más

cortas a medida que disminuye la temperatura, en concordancia con la respuesta térmica esperada de una FBG. El desplazamiento del *peak* será posteriormente cuantificado mediante el algoritmo de correlación cruzada para obtener $\Delta\lambda_B(t)$.

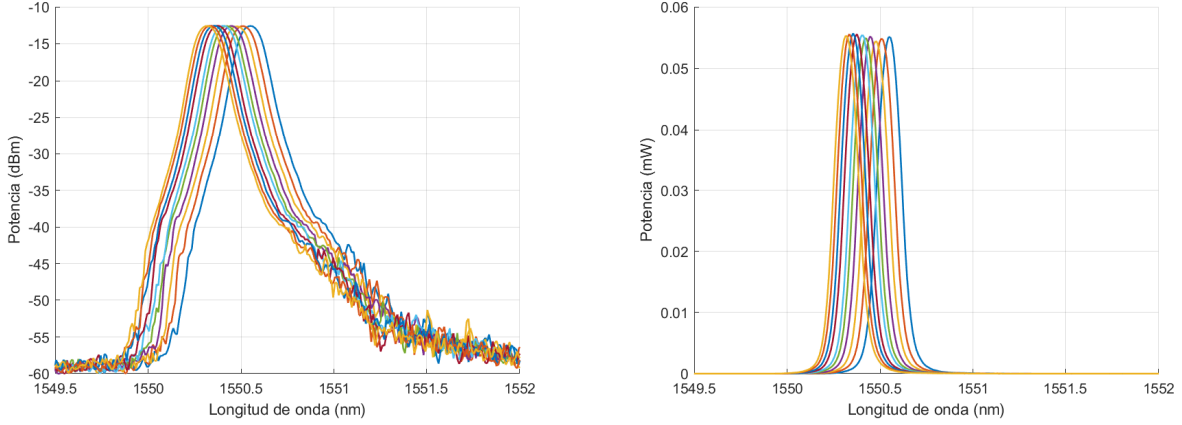


Figure 3.5: Comparación del espectro reflejado por la FBG antes y después de la conversión de escala. (a) Escala logarítmica. (b) Escala lineal.

La Figura 3.6 muestra la evolución de la temperatura medida con un termómetro de referencia, calibrado con una incertidumbre instrumental de $\pm 0.1^\circ\text{C}$. Las mediciones se registraron cada 50 s, lo que proporciona 25 puntos de referencia distribuidos a lo largo del experimento (puntos rojos) y las medidas de la FBG (puntos azules están tomadas cada 0.1 s, la figura 3.6-(a) muestra que existe una única medición para un tiempo en particular. Debido a la diferencia de frecuencia de muestreo entre el interrogador (10 Hz) y el termómetro (0.02 Hz), los valores de temperatura se interpolan linealmente para permitir una comparación punto a punto con los resultados del algoritmo.

Este conjunto de datos permite validar la precisión del método de detección de *peak* mediante correlación cruzada, al comparar $\Delta\lambda_B(t)$ estimado con el cambio de temperatura medido independientemente.

Uno de los aspectos clave a evaluar es la precisión con la que el sistema puede detectar cambios en la longitud de onda de Bragg (λ_B) a lo largo del tiempo. Los datos utilizados provienen directamente del interrogador óptico, el cual adquiere espectros en el rango espectral de 1500 nm a 1600 nm con una resolución de 10 pm, lo que resulta en 10 000 muestras espectrales por medición.

La Figura 3.7 muestra el espectro de la FBG en escala logarítmica y varias versiones submuestreadas obtenidas mediante *downsampling* con diferentes factores n . Este proceso consiste en seleccionar un punto cada n muestras del espectro original, es decir:

$$R_{\text{down}}(k) = R(n \cdot k + p), \quad k = 0, 1, \dots, \left\lfloor \frac{N - 1 - p}{n} \right\rfloor$$

donde N es el número de muestras espectrales originales, n es el factor de submuestreo, p es el **índice de partida** ($0 \leq p < n$), que determina la alineación de la grilla submuestreada, R_{down} es el espectro resultante, con una longitud de $\approx N/n$ puntos.

Por ejemplo, con $n = 5$, el espectro submuestreado tiene aproximadamente 2000 puntos. Este tipo de reducción es útil para disminuir la carga computacional en el procesamiento de grandes volúmenes de datos, pero introduce consideraciones importantes sobre la resolución espectral.

Es importante destacar que el valor de p influye directamente en la posición relativa de las muestras respecto al perfil espectral original. Un *downsampling* con $p = 0$ toma las muestras en

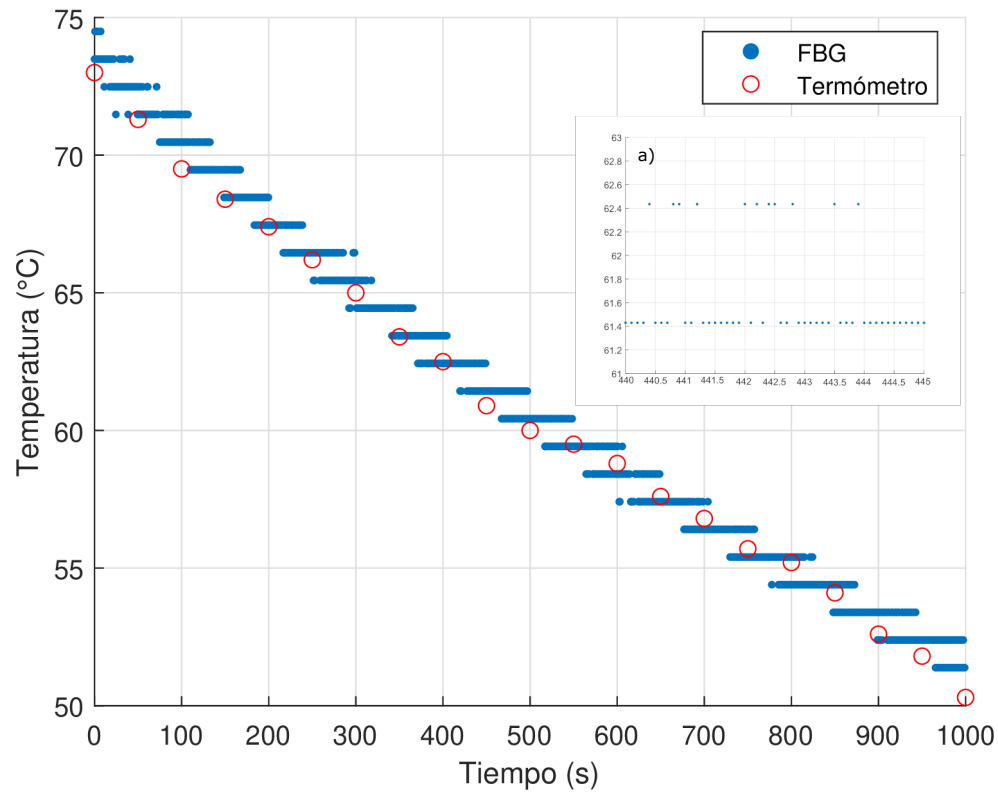


Figure 3.6: Evolución de la temperatura medida con termómetro de referencia durante el Experimento 1. Los puntos representan mediciones reales cada 50 s.

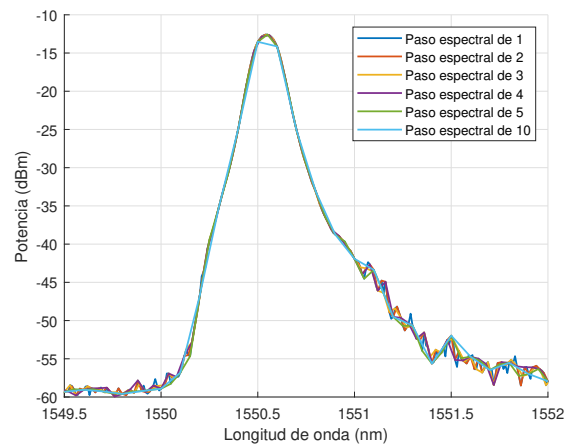


Figure 3.7: Ejemplo del espectro de la FBG original y sus versiones submuestreadas obtenidas mediante *downsampling* con diferentes factores n .

$\lambda_1, \lambda_6, \lambda_{11}, \dots$, mientras que con $p = 2$ se obtiene $\lambda_3, \lambda_8, \lambda_{13}, \dots$. Esta diferencia puede provocar que el *peak* no coincida con ninguna muestra seleccionada, introduciendo un *error de cuantización espectral*, por lo tanto es importante considerar todos los casos de submuestreo para así tener una comprobación estilísticamente correcta.

Como complemento al Experimento 1, se realizó un segundo experimento bajo condiciones térmicas similares (enfriamiento desde aproximadamente 73°C hasta 48°C), pero introduciendo una atenuación óptica aproximadamente de 10 dB mediante el VOA. Este escenario simula una condición de baja relación señal-ruido (SNR).

El objetivo de este experimento es evaluar la robustez del algoritmo de detección del *peak* espectral frente a la reducción deliberada de la potencia óptica reflejada. Al comparar los resultados con los del Experimento 1 (sin atenuación), es posible analizar si el desplazamiento estimado $\Delta\lambda_B(t)$ se ve afectado por el nivel de señal, o si el método de correlación cruzada mediante FFT mantiene su precisión incluso bajo condiciones adversas.

La Figura 3.8 muestra un conjunto representativo de espectros adquiridos durante este experimento. A pesar de la atenuación, el *peak* espectral sigue siendo claramente discernible, por esto los algoritmos trabajados en este proyecto, se analizara cuantitativamente el error en la detección del *peak* respecto al experimento de referencia.

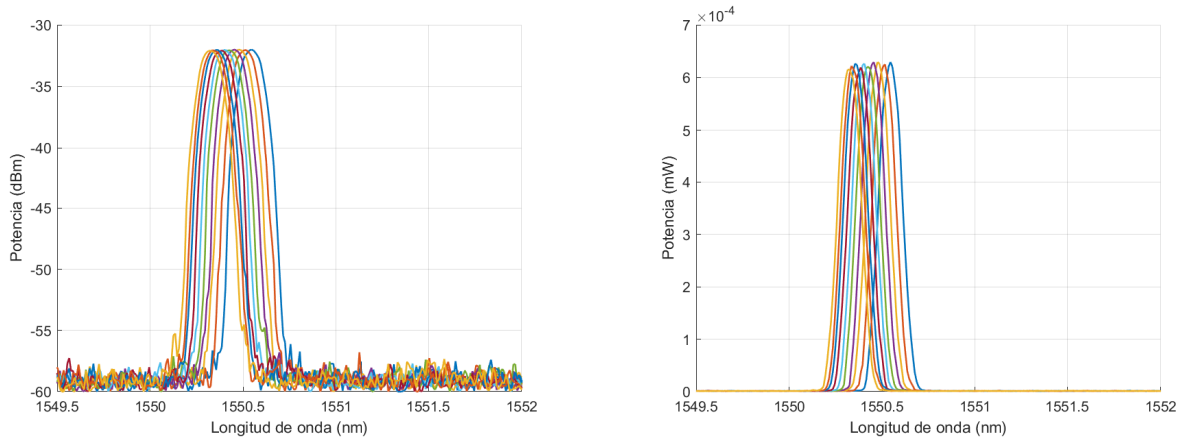


Figure 3.8: Experimento 2 el cual presenta una atenuación de 10dB. (a) Escala logarítmica. (b) Escala lineal.

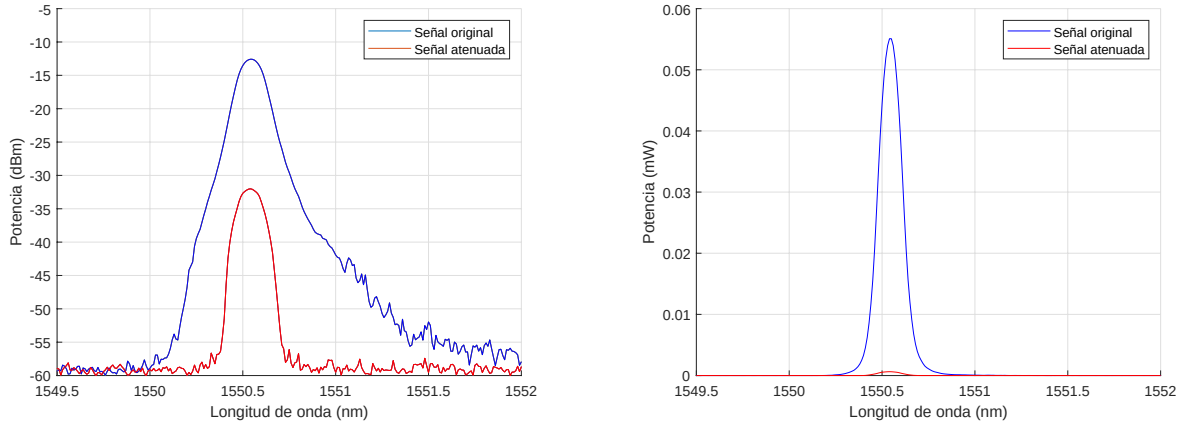


Figure 3.9: Comparación de los espectros reflejados en los Experimentos 1 y 2. A pesar de la atenuación de 10 dB. (a) Escala logarítmica. (b) Escala lineal.

La Figura 3.9 muestra una comparación directa entre el Experimento 1 (sin atenuación) y el Experimento 2 (con 10 dB de atenuación), en ambas escalas: logarítmica (dBm) y lineal (mW). En la escala logarítmica (Figuras 3.9-(a)), la atenuación se observa como un desplazamiento vertical uniforme de todo el espectro, manteniéndose la forma del *peak* aunque con menor amplitud.

Las figuras 3.9, se evidencia una reducción significativa en la potencia óptica reflejada, especialmente en el pico y en los lóbulos laterales. Aun así, el perfil espectral conserva su forma característica, lo que permite al algoritmo de correlación cruzada seguir detectando el desplazamiento con precisión.

Experimento 3: Análisis del desempeño en función de la SNR

Con el fin de evaluar la precisión del algoritmo de detección del *peak* espectral frente a diferentes niveles de relación señal-ruido (SNR), se realizó una serie de 12 mediciones en condiciones térmicas estables, alrededor de 31°C. Esta temperatura se mantuvo constante para garantizar que no ocurriera un desplazamiento de la longitud de onda de Bragg ($\Delta\lambda_B \approx 0$), permitiendo aislar el error de medición inducido por el ruido y la atenuación.

Cada medición tuvo una duración de 10 s y se adquirió a una frecuencia de 10 Hz, lo que generó 100 espectros por nivel de atenuación. El nivel de SNR se controló sistemáticamente ajustando el atenuador óptico variable (VOA), lo que permitió modificar la potencia reflejada por la FBG antes de llegar al interrogador, simulando condiciones desde alta hasta baja SNR.

El objetivo es analizar cómo varía la precisión (desviación estándar del λ_B detectado) según el nivel de atenuación aplicado. Dado que no se espera un corrimiento real del *peak*, cualquier dispersión en las estimaciones consecutivas se atribuye directamente al ruido y a la limitada resolución del sistema.

La Figura 3.10 muestra la potencia máxima reflejada para cada una de las 12 configuraciones de atenuación, ilustrando la reducción progresiva de la señal conforme aumenta la atenuación del VOA.

3.1.4 Calibración del sensor FBG

Para convertir el corrimiento espectral detectado ($\Delta\lambda_B$) en una medición de temperatura física (T), es necesario establecer una relación funcional entre la longitud de onda de Bragg y la temperatura. Aunque en muchos casos se asume una relación lineal, estudios previos [7, 8, 32] han demostrado

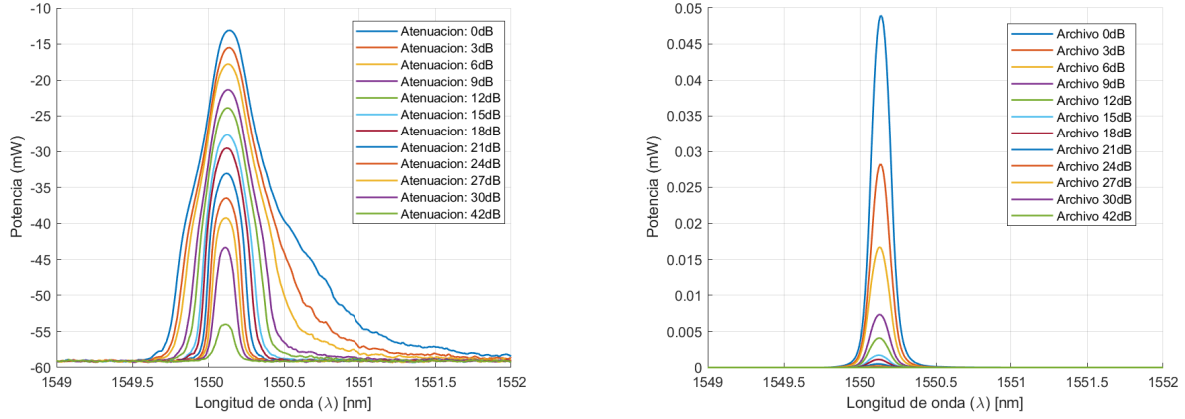


Figure 3.10: Potencia máxima reflejada para cada una de las 12 mediciones del Experimento 3, según el nivel de atenuación total aplicada por el VOA. (a) Escala logarítmica. (b) Escala Lineal.

que la respuesta térmica de una FBG puede modelarse con mayor precisión mediante una función cuadrática, especialmente en rangos amplios de temperatura.

En este trabajo, se realizó una calibración experimental del sensor FBG mediante la adquisición simultánea de:

- La longitud de onda de Bragg (λ_B), medida por el interrogador óptico,
- La temperatura del entorno (T), medida con un termómetro de referencia calibrado.

Las mediciones se realizaron a intervalos regulares durante un ciclo de calentamiento y enfriamiento, cubriendo un rango desde 25°C hasta 85°C. A partir de estos datos, se ajustó un modelo cuadrático de la forma:

$$\lambda_B(T) = a_0 + a_1T + a_2T^2 \quad (3.2)$$

donde a_0 , a_1 y a_2 son los coeficientes de calibración obtenidos mediante regresión mínimos cuadrados.

Este modelo permite estimar la temperatura a partir de una medición de λ_B mediante:

$$T = \frac{-a_1 \pm \sqrt{a_1^2 - 4a_2(a_0 - \lambda_B)}}{2a_2} \quad (3.3)$$

donde se selecciona la raíz positiva que corresponde al rango físico de operación.

Alternativamente, para aplicaciones donde se mide el “cambio de temperatura” respecto a un punto de referencia (T_0 , λ_{B0}), se utiliza la relación del corrimiento espectral:

$$\Delta\lambda_B = \alpha\Delta T + \beta(\Delta T)^2 \quad (3.4)$$

con α y β como coeficientes térmicos de primer y segundo orden. Este enfoque simplifica el cálculo y es el que se utiliza en las secciones posteriores para convertir $\Delta\lambda_B$ en ΔT , donde los coeficientes se determinaron mediante regresión por mínimos cuadrados, resultando en:

$$T = 0.06475 \cdot \lambda_B^2 - 100.3 \cdot \lambda_B - 78.64 \quad (3.5)$$

con λ_B en nm y T en °C.

Este proceso de calibración es fundamental para garantizar la precisión del sistema de medición, ya que cualquier error en el modelo se traduce directamente en errores en la estimación de temperatura. En los análisis posteriores, se utiliza esta relación cuadrática para convertir los corrimientos espectrales detectados por el algoritmo de correlación cruzada en valores físicos de temperatura.

3.2 Análisis y resultados.

3.2.1 Metodología de análisis

Para evaluar cuantitativamente el desempeño del algoritmo de detección del *peak* espectral basado en correlación cruzada mediante FFT, se emplean dos métricas clave: el **error cuadrático medio (RMS)** y la **desviación estándar**. Estas permiten analizar la precisión y estabilidad del método en distintos escenarios experimentales, como variaciones de temperatura, atenuación óptica y resolución espectral.

El error RMS, denotado como $\Delta\lambda_B^{\text{RMS}}$ o ΔT^{RMS} , mide la precisión del algoritmo al comparar los valores estimados con los valores de referencia. En este caso, los valores **estimados** (λ_B^{pred} o T^{pred}) corresponden a los resultados obtenidos mediante el algoritmo propuesto (correlación cruzada con FFT). Los valores **de referencia** (λ_B^{ref} o T^{ref}) se obtienen de forma independiente.

El error RMS se calcula como:

$$\Delta\lambda_B^{\text{RMS}} = \sqrt{\frac{1}{n} \sum_{i=1}^n \left(\lambda_B^{\text{pred}}(i) - \lambda_B^{\text{ref}}(i) \right)^2} \quad (3.6)$$

donde n es el número total de espectros adquiridos en un experimento o condición específica.

Esta métrica indica el desvío promedio del algoritmo respecto al valor de referencia, y es especialmente útil para evaluar sesgos sistemáticos o tendencias en la estimación.

La desviación estándar, σ_{λ_B} , cuantifica la **estabilidad** o dispersión de las mediciones consecutivas bajo condiciones físicas constantes. A diferencia del RMS, esta métrica no requiere un valor de referencia, sino que evalúa la repetibilidad del algoritmo frente al ruido y fluctuaciones instrumentales.

Se define como:

$$\sigma_{\lambda_B} = \sqrt{\frac{1}{n-1} \sum_{i=1}^n \left(\lambda_B(i) - \overline{\lambda_B} \right)^2} \quad (3.7)$$

donde $\overline{\lambda_B}$ es el valor medio de λ_B en un conjunto de n mediciones.

Un valor bajo de σ_{λ_B} indica alta repetibilidad, lo cual es fundamental para aplicaciones de monitoreo continuo. En el contexto de este trabajo, se considera que $\sigma_{\lambda_B} < 0.05$ pm (equivalente a $\sigma_T < 0.05$ °C tras la calibración) representa una dispersión aceptable para operaciones en régimen estacionario. Esta métrica se calcula en ventanas deslizantes de 10 muestras consecutivas, tal como se fundamentó en la Sección 3.2.1, estrategia que permite cuantificar la repetibilidad del algoritmo bajo condiciones cuasi-estacionarias. El uso de ventanas locales de tamaño $n = 10$ asegura que las muestras analizadas correspondan a intervalos de tiempo suficientemente cortos ($\Delta t = 1$ s para una frecuencia de adquisición de 10 Hz) donde la temperatura puede considerarse constante dentro de la incertidumbre instrumental del termómetro de referencia (± 0.1 °C). De esta manera, cualquier variación en λ_B dentro de la ventana se atribuye principalmente al ruido instrumental y a la estabilidad del algoritmo, no a cambios físicos reales. Esta métrica será utilizada principalmente en el Experimento 3, donde se mantiene constante la temperatura para aislar el ruido del sistema y evaluar la robustez frente a diferentes niveles de atenuación [3, 9].

3.2.2 Resultados

La Figura 3.11 muestra la evolución del desplazamiento espectral detectado por el sensor FBG durante el Experimento 1, en el cual la temperatura del sistema varió desde aproximadamente 73°C hasta 48°C. Los datos del FBG corresponden a espectros de reflectividad adquiridos por el interrogador, los cuales fueron procesados mediante el algoritmo de correlación cruzada basado en FFT (el método principal de estudio) para estimar el corrimiento del *peak* espectral, $\Delta\lambda_B(t)$.

Mediante una calibración previa del sensor, este corrimiento espectral se convierte en una estimación de la temperatura del entorno. En la figura, se compara $T^{\text{pred}}(t)$ (obtenida del FBG) con $T^{\text{ref}}(t)$, medida independientemente por un termómetro. Ambas señales muestran una tendencia de enfriamiento lo que incentiva a pensar en el buen funcionamiento del algoritmo para la detección.

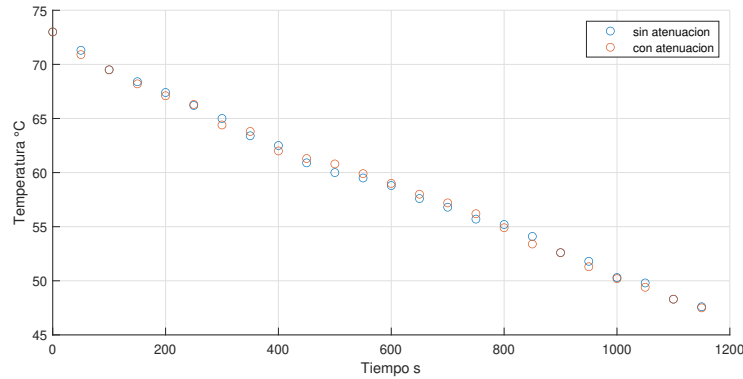


Figure 3.11: Comparación entre la temperatura estimada a partir del desplazamiento espectral detectado por el algoritmo de FFT (*puntos azules*) y la temperatura medida por el termómetro (*puntos rojos*) durante el Experimento 1. El rango térmico va desde 73°C hasta 48°C. La alta concordancia entre ambas señales valida la precisión del método propuesto.

Desempeño computacional en función del tiempo

Para evaluar la eficiencia de los diferentes algoritmos de detección del *peak* espectral, se realizó un análisis de su desempeño computacional en función del tiempo de ejecución. Se consideraron las cuatro implementaciones descritas previamente.

El test consistió en aplicar cada algoritmo a un conjunto de 10 000 espectros adquiridos durante el Experimento 1, y registrar el tiempo promedio de ejecución por espectro, junto con la desviación estándar de las mediciones.

La Tabla 3.1 resume los resultados obtenidos. Como se observa, el método basado en FFT (Método 2) presenta el menor tiempo promedio de ejecución (0.000184 s), seguido por el uso de `xcorr` (Método 1). En contraste, el método directo (Método 4) es significativamente más lento (0.055736 s), lo que confirma su alta complejidad computacional $O(N^2)$.

Este análisis permite seleccionar el algoritmo óptimo para aplicaciones en tiempo real, donde la velocidad es crítica. Considerando que la frecuencia de adquisición del interrogador es de 10 Hz, el sistema dispone de un *deadline* estricto de $T_m = 100$ ms entre la llegada de espectros consecutivos.

Para garantizar un procesamiento en tiempo real robusto, el tiempo de ejecución del algoritmo t_{exec} debe satisfacer $t_{exec} \ll T_m$, dejando un margen computacional para tareas auxiliares del sistema.

Como se observa en la Tabla 3.1, todos los métodos basados en FFT (Métodos 1 a 4) presentan tiempos de ejecución del orden de 10^{-4} s, lo que representa menos del 0.6 % del periodo de muestreo

Table 3.1: Tiempo de ejecución promedio y desviación estándar para cada algoritmo sobre 10 000 espectros

Algoritmo	Tiempo promedio (s)	Desviación estándar (s)
Método 1 (xcorr)	0.000585	0.000087
Método 2 (FFT sin zero-padding)	0.000171	0.000034
Método 3 (FFT con zero-padding)	0.000318	0.000048
Método 4 (FFT con ajuste polinomial)	0.000309	0.000050
Método 5 (Correlación directa)	0.059597	0.002206

disponible. Esta holgura computacional permite no solo procesar cada espectro con amplia anticipación, sino también habilitar estrategias de multiplexión temporal o ejecución de algoritmos de diagnóstico en paralelo sin comprometer la integridad del sistema.

El Método 2 (FFT sin zero-padding) destaca como el más rápido ($171 \mu\text{s}$), seguido por el Método 4 (FFT con ajuste polinomial, $309 \mu\text{s}$). Aunque el Método 4 requiere un 80 % más de tiempo que el Método 2, su costo absoluto sigue siendo despreciable frente al deadline de 100 ms.

En contraste, el Método 5 (Correlación directa) presenta un tiempo de ejecución de 59.6 ms, que aunque técnicamente cumple con el deadline de 100 ms, consume el 59.6 % del tiempo disponible. Esta carga computacional deja un margen insuficiente para tareas del sistema operativo o manejo de interrupciones, aumentando el riesgo de *buffer overrun* en escenarios de alta demanda. Además, su mayor desviación estándar (2.2 ms) indica variabilidad en el tiempo de respuesta, lo cual es indeseable en sistemas de control en tiempo real estricto.

Además, se realizó una comparación de la precisión de los resultados entre todos los métodos, utilizando como referencia la temperatura medida por el termómetro. La Tabla 3.2 muestra el error cuadrático medio (RMS) en la estimación de ΔT , demostrando que la mayoría de los métodos mantienen una alta precisión, con valores inferiores a 0.5°C .

Table 3.2: Error RMS en la estimación de temperatura (ΔT^{RMS}) para cada algoritmo

Algoritmo	ΔT^{RMS} ($^\circ\text{C}$)	Desviación estándar ($^\circ\text{C}$)
Método 1 (xcorr)	0.6640	0.0521
Método 2 (FFT sin zero-padding)	0.6640	0.0521
Método 3 (FFT con zero-padding)	0.6640	0.0521
Método 4 (FFT con ajuste polinomial)	0.6140	0.0392
Método 5 (Correlación directa)	0.7498	0.2592

La Tabla 3.2 revela que los cuatro métodos basados en FFT (**xcorr**, FFT sin zero-padding, FFT con zero-padding y FFT con ajuste polinomial) presentan un error RMS muy similar en la estimación de temperatura ($\Delta T^{\text{RMS}} \approx 0.64^\circ\text{C}$), lo que indica una alta consistencia en su precisión, destacando la mejoría del método 4 esperada por su etapa de ajuste para encontrar un *peak* más preciso. Esta similitud sugiere que el método de transformada rápida no introduce sesgos significativos en la detección del *peak*, y que la elección entre ellos debe basarse principalmente en el tiempo de ejecución y eficiencia computacional. En contraste, el método de correlación directa (Método 5) muestra un error RMS ligeramente mayor (0.7498°C), lo que puede atribuirse a la acumulación de errores numéricos en la suma iterativa o a una menor resolución espectral debido a la baja eficiencia del algoritmo. Además, su desviación estándar más elevada (0.2592°C) indica una mayor

variabilidad en las mediciones, posiblemente por sensibilidad al ruido o inestabilidad en el cálculo. Estos resultados confirman que el uso de FFT no solo es más eficiente, sino también tan preciso como los métodos directos, lo cual justifica su uso en aplicaciones prácticas donde se requiere un equilibrio entre velocidad y exactitud.

Desempeño en función de la temperatura

Para evaluar cómo varía el desempeño de los diferentes algoritmos según la temperatura ambiente, se analizó como se describe en la sección 3.2.1 las mediciones en cada punto térmico durante el Experimento 1. La Figura 3.12 muestra estos valores para los cuatro métodos considerados, como función de la temperatura medida por el termómetro.

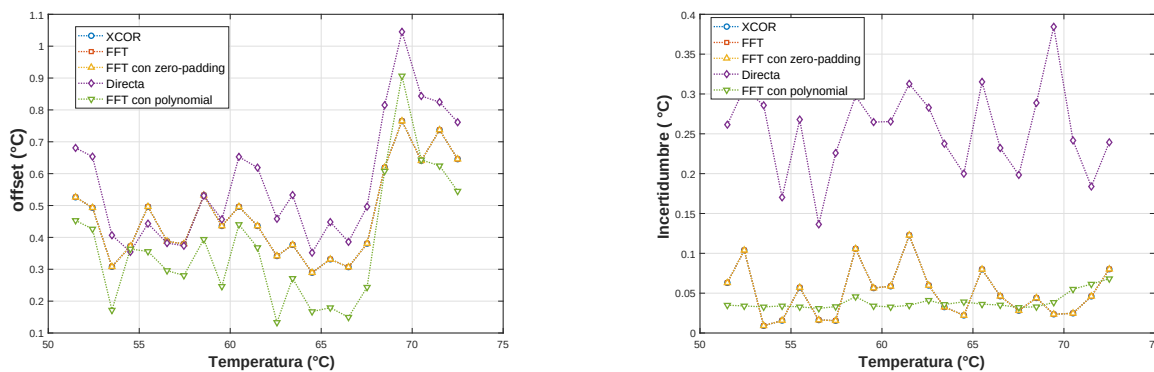


Figure 3.12: Evolución del error RMS y desviación estándar en función de la temperatura para los cuatro algoritmos. (a) offset RMS. (b) Desviación estándar.

Como se observa en la Figura 3.12-(a), todos los métodos presentan un incremento del error RMS en los extremos del rango térmico (48°C y 73°C). Este comportamiento se atribuye a que dichas regiones corresponden a los instantes inicial y final del experimento, donde el número de muestras disponibles es menor debido a las transiciones de arranque y finalización del sistema de adquisición. En consecuencia, la menor cantidad de puntos espectrales en estas zonas de transición introduce una mayor incertidumbre en la estimación del peak [3, 9]. Por el contrario, en regiones de temperatura constante o lentamente variable (entre 60°C y 66°C), el error disminuye significativamente, alcanzando valores mínimos de 0.3°C para los métodos basados en FFT. Esta mejora en la precisión se debe a que el sistema opera en régimen estacionario, permitiendo una mayor consistencia en las mediciones espectrales.

En cuanto a la desviación estándar (Figura 3.12-(b)), se confirma que el método de correlación directa (Método 5) presenta la mayor dispersión de las mediciones en todo el rango, con valores que superan 0.25°C en múltiples puntos. Esta alta variabilidad refleja la sensibilidad del algoritmo al ruido instrumental y a las fluctuaciones térmicas transitorias, lo cual limita su aplicabilidad en sistemas de monitoreo continuo [29]. Por otra parte, el algoritmo que incorpora ajuste polinomial cuadrático (Método 4) demuestra el mejor desempeño global, manteniendo no solo el menor error RMS sino también la menor desviación estándar ($\sigma < 0.05^\circ\text{C}$) en la mayoría del rango térmico. Este resultado valida que el refinamiento sub-muestral mediante interpolación polinomial del peak de correlación mejora significativamente la repetibilidad de las mediciones, incluso en presencia de ruido y variaciones térmicas moderadas [3, 24]. La consistencia entre el bajo error sistemático (RMS) y la baja dispersión (desviación estándar) confirma que el Método 4 ofrece la mejor combinación de precisión y estabilidad para aplicaciones prácticas de interrogación de FBG.

Para evaluar la robustez del algoritmo de correlación cruzada basado en FFT frente a condiciones de baja relación señal-ruido (SNR), se realizó una comparación directa entre las mediciones obtenidas en el Experimento 1 (sin atenuación) y el Experimento 2 (con 20 dB de atenuación total). El objetivo fue analizar si el error en la estimación de temperatura (ΔT^{RMS}) y la desviación estándar de las mediciones se mantienen dentro de rangos aceptables cuando la potencia óptica se reduce significativamente.

La Figura 3.13-(a) muestra los resultados del offset de temperatura para los métodos realizados, en función de la temperatura considerando la atenuación ejercida. Como se observa en la Figura 3.13-(b), se adjunta una comparación para el método de la FFT de los 2 casos, el error RMS con atenuación sigue una tendencia muy similar a la de la condición sin atenuación, especialmente en el rango de 55°C a 70°C, donde ambos alcanzan valores mínimos cercanos a 0.3°C. Aunque se observa un ligero aumento del error en los bordes del rango, este no supera los 1.5°C, lo cual indica que el algoritmo puede operar con precisión incluso bajo condiciones adversas.

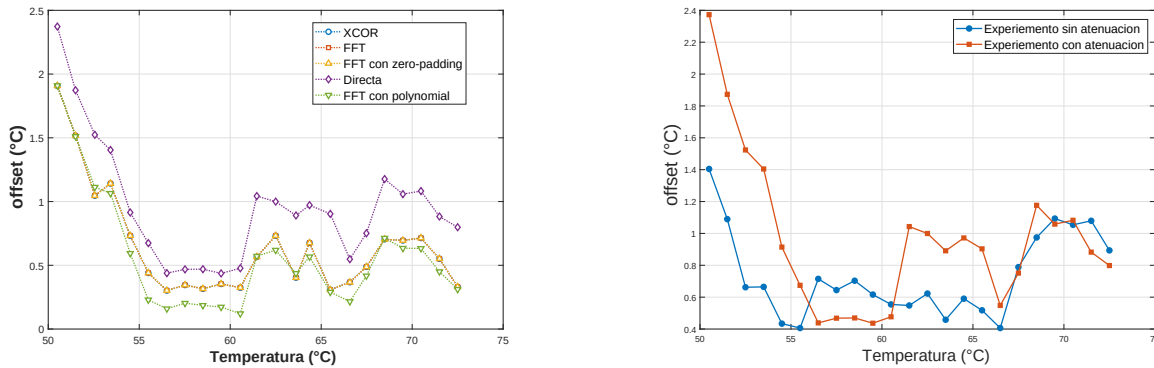


Figure 3.13: Desempeño de los métodos para el experimento con atenuación de 20 dB.(a) Comparación entre el desempeño del algoritmo de FFT con y sin atenuación de 20 dB.(b)

En cuanto a la comparación de ambos experimentos (Figura 3.13-(b)), se confirma que el algoritmo mantiene una precisión similar en ambas condiciones, aunque ligeramente mayor sin atenuación. Esto sugiere que, aunque el ruido afecta ligeramente la estabilidad de las mediciones, no compromete la capacidad del algoritmo para seguir la dinámica térmica. Este resultado reafirma que el método de correlación cruzada mediante FFT es robusto frente a reducciones de señal, ya que logra mantener un desempeño comparable al caso ideal (sin atenuación).

Table 3.3: Offset en la estimación de temperatura (ΔT^{RMS}) para cada algoritmo para el experimento con atenuación total de 20 dB

Algoritmo	ΔT^{RMS} (°C)	Desviación estándar (°C)
Método 1 (xcorr)	0.74210	0.0662
Método 2 (FFT sin zero-padding)	0.74210	0.0662
Método 3 (FFT con zero-padding)	0.74210	0.0662
Método 4 (FFT con ajuste polinomial)	0.6993	0.0481
Método 5 (Correlación directa)	0.8217	0.2196

Al comparar la tabla 3.2 con la tabla 3.3 los números resultantes presentan similitudes sin aumentar los errores considerablemente aun cuando tenemos presente la atenuación. Demostrando una

robustez a valores de SNR más bajos de lo esperado.

Desempeño en función de la atenuación

Para evaluar la robustez de los algoritmos frente a diferentes niveles de atenuación óptica, se realizó el Experimento 3, en el cual se aplicaron niveles de atenuación mencionados anteriormente. El objetivo fue analizar cómo varía el error en la estimación de temperatura (ΔT^{RMS}) y la desviación estándar de las mediciones, como función del nivel de señal.

La Figura 3.14 muestra estos resultados para los cinco métodos considerados. Como se observa en la Figura 3.14-(a), todos los métodos presentan un aumento del error RMS con la subida de atenuación, lo cual es esperado debido a la reducción del SNR.

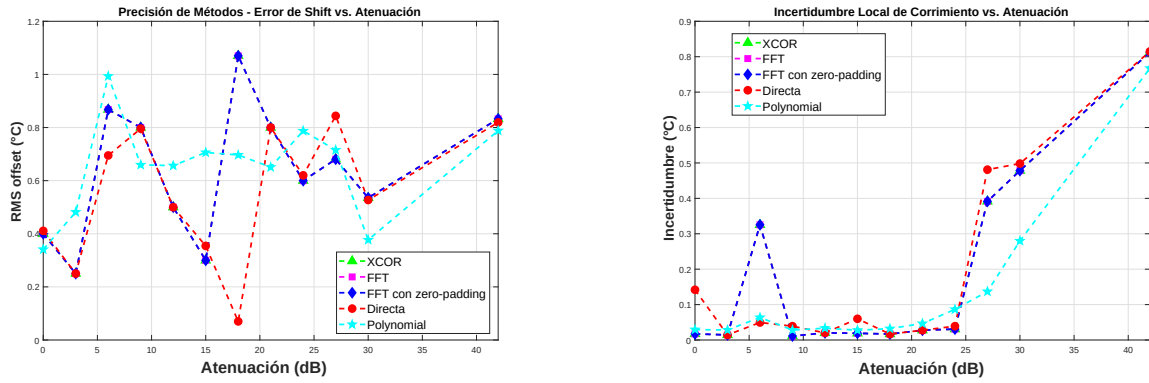


Figure 3.14: Evolución del error RMS y desviación estándar en función de la atenuación total para los cinco algoritmos. (a) offset RMS. (b) Desviación estándar.

En cuanto a la incertidumbre de las estimaciones, representada por la desviación estándar en la Figura 3.14-(b), se observa un comportamiento diferenciado según el nivel de atenuación aplicado. Para valores de atenuación inferiores a 25 dB, los cinco algoritmos mantienen una desviación estándar relativamente constante y baja ($\sigma_T < 0.01^\circ\text{C}$), lo que evidencia la robustez del método de correlación cruzada frente a reducciones moderadas de la potencia óptica. Esta estabilidad se fundamenta en la propiedad de promediado espectral inherente a la transformada de Fourier: al operar en el dominio frecuencial, la correlación realza las componentes coherentes del espectro de la FBG y suprime parcialmente el ruido blanco aditivo, preservando la repetibilidad en la detección del peak [3, 28].

Sin embargo, para niveles de atenuación superiores a 30 dB, se registra un incremento progresivo de la desviación estándar, particularmente en el método de correlación directa (Método 4). Este comportamiento corresponde al límite teórico donde la relación señal-ruido (SNR) en el dominio de la correlación se degrada significativamente: cuando la amplitud del ruido de fondo es comparable a la del lóbulo principal de la función de correlación, la posición del máximo estimado presenta mayor dispersión estadística [11, 12].

Desempeño ante submuestreo espacial

Como se mencionó en la Sección 3.1.3, se realizaron diversas comparaciones para evaluar el efecto del submuestreo espacial en el desempeño del algoritmo de detección del *peak*. Para ello, se definieron distintos factores de submuestreo $n \in \{1, 2, 3, 4, 6, 8, 15\}$, donde cada valor n indica que se conserva

una muestra cada n puntos del espectro original, es decir, $R_{\text{down}}(k) = R(n \cdot k + p)$, con p el índice de partida [3].

Cabe resaltar que este estudio no se centra en la variación del muestreo espectral global, sino en la cantidad de puntos disponibles dentro de la zona del *peak* espectral, ya que esta constituye una métrica más absoluta y físicamente significativa para cuantificar el desempeño del algoritmo. El número de puntos en la zona del *peak* se calcula considerando aquellas muestras espectrales cuya amplitud satisface $R(\lambda) \geq R_{\text{max}} - 3 \text{ dB}$, criterio que corresponde al ancho de banda a media potencia (FWHM, *Full Width at Half Maximum*) y delimita la región espectral donde se concentra la mayor energía de la señal reflejada [2, 3]. Esta definición permite evaluar de manera consistente cómo la reducción de resolución afecta la capacidad del algoritmo para estimar con precisión la posición del máximo, independientemente de la longitud total del espectro.

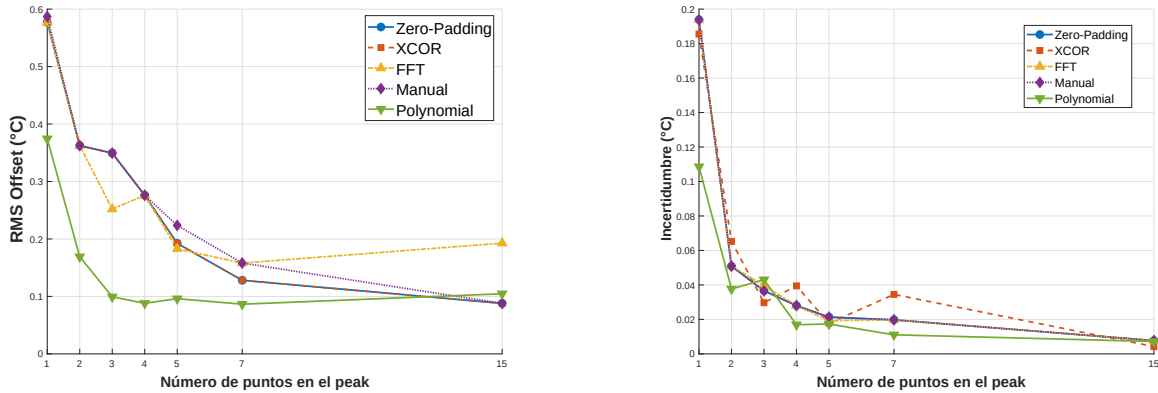


Figure 3.15: Evolución del error RMS y desviación estándar en función de la atenuación total para los cinco algoritmos. (a) offset RMS. (b) Desviación estándar.

La Figura 3.15 muestra el comportamiento del error RMS en función del número de puntos espectrales disponibles dentro de la región del peak. Como se observa, todos los métodos presentan un aumento progresivo del error a medida que disminuye el número de puntos en el peak, lo cual es esperado debido a la pérdida de resolución espectral y de información sobre la forma del perfil de Bragg. Sin embargo, este incremento no es lineal: para un número elevado de puntos en el peak, el error aumenta lentamente y se mantiene en valores aceptables ($\text{RMS} < 0.2^\circ\text{C}$), mientras que para menos de 2 puntos en el peak, el crecimiento del error se acelera significativamente, superando 0.5°C en algunos casos [3, 9]. Este comportamiento evidencia que el algoritmo requiere una densidad mínima de muestreo en la región espectral crítica para mantener la precisión sub-muestral, siendo los métodos basados en FFT más robustos que la correlación directa frente a la reducción de puntos disponibles.

3.2.3 Análisis general de resultados

El presente capítulo ha evaluado el desempeño del algoritmo de detección del peak espectral basado en correlación cruzada mediante transformada rápida de Fourier (FFT) bajo una serie de condiciones experimentales críticas. A través de múltiples escenarios, incluyendo variación térmica, atenuación óptica controlada y submuestreo espacial, se ha demostrado que el método presenta un equilibrio óptimo entre precisión, eficiencia computacional y robustez frente a variaciones en el entorno de medición. En primer lugar, el análisis de tiempo de ejecución revela que los métodos basados en FFT son significativamente más eficientes que la correlación directa, consumiendo

menos del 0.6 % del periodo de muestreo disponible para una frecuencia de adquisición de 10 Hz, lo que valida su viabilidad para aplicaciones en tiempo real sin riesgo de pérdida de datos [11, 29]. En cuanto a la precisión, el error RMS en la estimación de temperatura (ΔT_{RMS}) se mantiene inferior a 0.9 °C para la totalidad de los métodos implementados, con valores típicos de 0.5 °C para las variantes basadas en FFT. Esta convergencia en los resultados de error demuestra que la implementación mediante transformada rápida no introduce sesgos significativos, validando la hipótesis de que el dominio frecuencial preserva la integridad de la información espectral durante el proceso de correlación. La consistencia de un RMS bajo a lo largo de todo el rango térmico y bajo diferentes niveles de atenuación genera una medición de confianza, ya que indica que el algoritmo es capaz de distinguir el desplazamiento del peak real frente a fluctuaciones instrumentales o ruido ambiental. Fundamentado en la teoría de procesamiento digital de señales, la correlación cruzada en el dominio de la frecuencia realza las componentes coherentes del espectro de la FBG mientras suprime parcialmente el ruido blanco aditivo, lo que se traduce directamente en una menor dispersión estadística de las estimaciones [3, 11]. Por lo tanto, la baja magnitud del error RMS no solo cuantifica la exactitud, sino que actúa como una métrica de robustez que garantiza la fiabilidad del sistema de interrogación en condiciones operativas variables.

El método mantuvo un desempeño estable incluso con 20 dB de atenuación, con un aumento moderado del error RMS, lo que demuestra su capacidad para operar en condiciones de baja relación señal-ruido (SNR), como las que podrían presentarse en sistemas de fibra larga o con múltiples sensores multiplexados [24]. Respecto al submuestreo espacial, se confirmó que el desempeño del algoritmo depende fuertemente de la alineación entre las grillas espectrales. Cuando ambas señales están submuestreadas con el mismo factor n , el error aumenta lentamente hasta $n = 10$. Sin embargo, si solo la medición está submuestreada, el error crece drásticamente, lo cual indica que la alineación de grillas es crucial para mantener la precisión. Cuando se mantiene la simetría (mismo índice de partida p), el error es bajo y constante; en cambio, cualquier desalineación introduce errores sistemáticos [3, 9]. Finalmente, el método de correlación cruzada mediante FFT se destacó por su consistencia y estabilidad en todos los escenarios evaluados. Su combinación de alta precisión, eficiencia computacional y robustez frente a ruido y pérdida de resolución lo convierte en una herramienta poderosa para el monitoreo de variables físicas mediante sensores FBG. En consecuencia, se concluye que el algoritmo propuesto es adecuado para aplicaciones prácticas, especialmente en sistemas de monitoreo estructural, donde se requiere una detección precisa y confiable del desplazamiento espectral en tiempo real.

Capítulo 4

Implementación de Algoritmos en Lenguaje C

El enfoque primordial de esta memoria se centra en la verificación de la factibilidad de los métodos de detección de picos espectrales basados en FFT, específicamente en su aplicación a sensores de fibra óptica FBG. El objetivo a futuro de esta investigación es sentar las bases que permitan la migración de esta implementación algorítmica a un entorno físico, como un microcontrolador o una plataforma embebida. Por consiguiente, se ha definido como un objetivo central la migración del desarrollo algorítmico realizado y validado en el entorno de Matlab hacia una implementación en lenguaje C. La elección de C se justifica por ser el lenguaje preferente en la mayoría de los sistemas embebidos (*embedded systems*), dada su eficiencia computacional intrínseca, su control directo sobre la gestión de la memoria y su proximidad al hardware, elementos esenciales para el despliegue en tiempo real de la solución propuesta.

Para materializar esta migración, se llevan a cabo dos aproximaciones distintas en el traspaso del código desde MATLAB a C. La primera estrategia involucra la utilización de la herramienta MATLAB Embedded Coder, la cual facilita la generación automática de código C/C++ a partir del código fuente de MATLAB, basado en un conjunto de parámetros de compilación preestablecidos. Este método provee una implementación de referencia rápida para fines de comparación y un método de validación de que el trabajo en Matlab es posible de llevarlo a un código C. La segunda estrategia, y fundamental para la optimización, consiste en el desarrollo de un código propio en C. Esta implementación manual se basa en la lógica algorítmica previamente desarrollada en MATLAB, pero se construye haciendo uso de librerías nativas de C especializadas en el procesamiento de señales. Para este fin, se emplean las rutinas de la librería FFTW (*Fastest Fourier Transform in the West*), la cual es ampliamente reconocida en la comunidad científica por su capacidad de *autotuning* para optimizar el rendimiento de la FFT. El código fuente y la documentación de esta librería se encuentran disponibles en su sitio web oficial (<http://www.fftw.org/>). Este enfoque busca un mejor control sobre la gestión de memoria y garantiza una eficiencia del algoritmo de FFT, un aspecto crucial para el despliegue en tiempo real del sensor FBG.

Ambas implementaciones en C son sometidas a pruebas que cuantifiquen el tiempo de respuesta, el error de representación entre otras. Las pruebas buscan replicar la metodología de evaluación presentada en el Capítulo 3, haciendo especial énfasis en la verificación de métricas críticas para sistemas embebidos. Se cuantifican la latencia y ejecución, y el comportamiento del sistema ante variaciones de la Relación Señal-Ruido (SNR). El análisis comparativo resultante permitirá determinar si la implementación manual y optimizada logra una diferencia o mejora sustancial en la eficiencia en comparación con el código generado automáticamente, validando así el potencial de

portabilidad del algoritmo de detección de picos y seleccionando la implementación más adecuada para empezar a trabajar en un prototipado final en hardware.

4.1 Generación de Código Mediante MATLAB Embedded Coder

Para la primera aproximación de migración, se emplea la herramienta MATLAB Coder. Esta herramienta permite tomar una función ya desarrollada y validada en el entorno de MATLAB para modificar sus parámetros y generar automáticamente el código fuente equivalente en el lenguaje C. En la Figura 4.1, se observa la interfaz proporcionada por el programa, la cual facilita la visualización y, de ser necesaria, la realización de ajustes a las propiedades de la función a exportar.

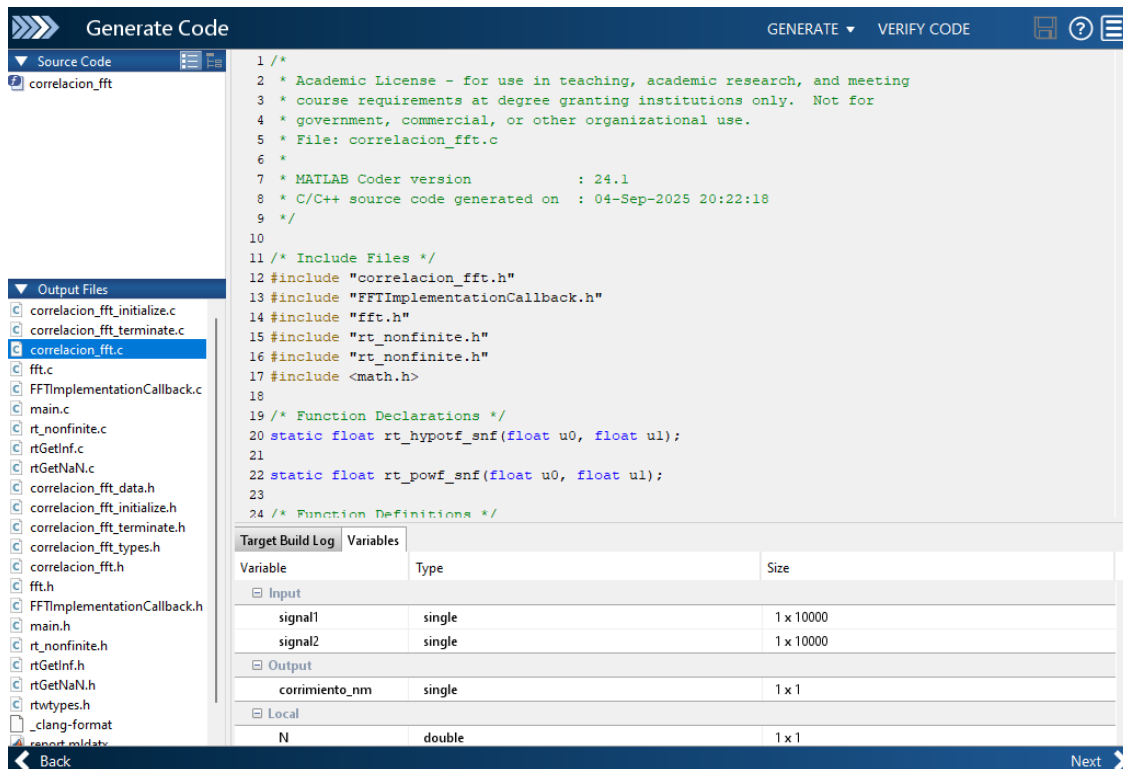


Figure 4.1: Interfaz de MATLAB Coder mostrando el análisis del algoritmo antes de la generación de código C. Permite verificar compatibilidad, modificar parámetros y previsualización de salidas.

Es fundamental destacar que la función de MATLAB utilizada para la generación de código en C es una versión adaptada y extendida respecto a las funciones descritas en el Capítulo 3. Esta nueva función encapsula el procedimiento completo de detección de pico espectral: toma como entrada el arreglo de datos correspondientes a las dos señales espectrales (*referencia* y *experimental*) y realiza los procedimientos descritos en el Código 1.1. A diferencia de las versiones anteriores, esta función integra etapas de preprocesamiento adicionales, incluyendo la linealización de las potencias espectrales, además del cálculo del pico mediante correlación cruzada y la conversión directa del resultado a unidades físicas de corrimiento (milímetros, mm). De esta manera, la función en C retorna directamente el desplazamiento ($\Delta\lambda$) del pico espectral en mm en función de las dos señales recibidas.

Para generar código C optimizado para su despliegue en sistemas embebidos, se configuran parámetros específicos dentro de MATLAB Coder [31]. En primer lugar, es crucial que las variables se

definan con el tipo de dato de punto flotante de precisión simple (`float`), con el fin de optimizar el consumo de memoria y la velocidad de procesamiento. En segundo lugar, se desactiva la generación de funciones de verificación y chequeo de datos (*runtime checks*), ya que estas rutinas, aunque añaden seguridad de que el código es ejecutable, consumen tiempo de procesamiento innecesario en un entorno donde las dimensiones de los datos son fijas y no requieren validación en tiempo de ejecución. Finalmente, se asegura que el compilador aplique las optimizaciones matemáticas pertinentes. Esta última configuración es crítica, puesto que se busca evitar que MATLAB Coder genere código con procesos o cálculos que resulten ineficientes para el tiempo de ejecución esperado, priorizando la velocidad de procesamiento sobre la legibilidad del código generado automáticamente.

El resultado directo de la compilación y ejecución de la función a través de MATLAB Coder es un archivo con extensión `mex` (*MATLAB Executable*). Este archivo binario constituye una librería compilada que permite que el código en C optimizado se ejecute directamente dentro del entorno MATLAB. La ventaja fundamental de utilizar el archivo `mex` radica en la capacidad de evaluar el rendimiento del código generado en C sin la sobrecarga del intérprete de MATLAB. Esto permite obtener mediciones precisas de los tiempos de ejecución y la respuesta algorítmica. De esta manera, el archivo `.mex` servirá como la herramienta para la obtención de resultados iniciales, los cuales serán posteriormente graficados y analizados comparativamente con la implementación manual en C.

4.2 Implementación de Código Propio en C

La segunda estrategia implementada consistió en el desarrollo de un código fuente en lenguaje C, diseñado para ser ejecutado y compilado directamente en una plataforma embebida para el trabajo futuro, prescindiendo del entorno de MATLAB.

El flujo de procesamiento implementado en C replica la lógica del algoritmo validado en el Capítulo 3, manteniendo la integridad de las etapas fundamentales. Inicialmente, los valores de potencia de las señales se linealizan durante el preprocesamiento para garantizar la consistencia física del espectro antes del cálculo de correlación. Posteriormente, las señales se transforman al dominio de Fourier mediante la FFT, donde se calcula el producto espectral entre la referencia y la medición. Luego, se aplica la transformada inversa (IFFT) para obtener la función de correlación cruzada en el dominio espectral. Finalmente, se ejecuta un algoritmo de detección del máximo sobre los datos de correlación para estimar la posición del peak espectral con la precisión requerida.

Cabe destacar que las operaciones de transformada rápida de Fourier (FFT) y su inversa (IFFT) se implementan mediante la librería externa FFTW (Fastest Fourier Transform in the West) [13], disponible oficialmente en <http://www.fftw.org/>, ya que el lenguaje C estándar no incluye estas funciones en sus librerías básicas como `math.h`. La estructura general del algoritmo, incluyendo el manejo de memoria y el flujo de procesamiento, se diseña manualmente optimizando el uso de esta librería. Esta combinación permite alcanzar el rendimiento requerido para aplicaciones en tiempo real, aprovechando la eficiencia de FFTW, ampliamente reconocida en la comunidad científica por su capacidad de *autotuning* para optimizar el rendimiento según la arquitectura del hardware [11, 13].

4.3 Análisis y Comparación de Resultados de Implementación

4.3.1 Comparación de la Precisión de Detección

Una vez obtenidas las dos implementaciones funcionales en lenguaje C, el código generado automáticamente por MATLAB Coder (MEX) y el código C manual utilizando la librería FFTW, se procede a compararlas resultados. Esta comparación se centra inicialmente en la precisión de la detección del corrimiento del pico espectral. La Figura 4.2 presenta el comportamiento de las estimaciones de temperatura obtenidas por ambos métodos, contrastándolas con las temperaturas de referencia medidas directamente durante el experimento.

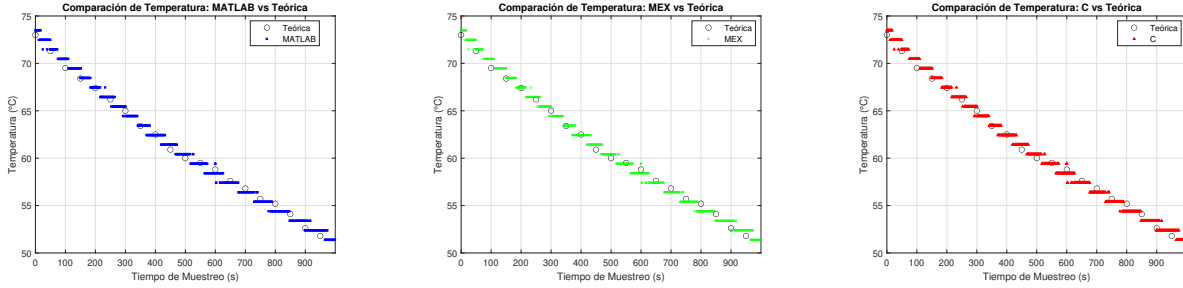


Figure 4.2: Comparación de la temperaturas teóricas medidas junto a las estimadas a partir de los métodos. (a)Implementación inicial de Matlab.(b)Implementación en Coder (MEX).(c) Implementación C

Las métricas de precisión presentadas se calculan siguiendo la metodología definida en la Sección 3.1.3 del Capítulo 3. El error cuadrático medio (RMS) se evalúa sobre el desplazamiento de temperatura estimado ΔT_{pred} respecto al valor de referencia ΔT_{ref} , según:

$$\Delta T_{\text{RMS}} = \sqrt{\frac{1}{N} \sum_{i=1}^N (\Delta T_{\text{pred}}(i) - \Delta T_{\text{ref}}(i))^2} \quad (4.1)$$

donde N corresponde al número total de espectros procesados en cada experimento. De manera equivalente, el RMS puede calcularse sobre el desplazamiento espectral $\Delta \lambda_B$ antes de la conversión a temperatura, permitiendo una evaluación independiente del modelo de calibración.

La desviación estándar, por su parte, se calcula en ventanas deslizantes de 10 muestras consecutivas, tal como se fundamentó en el Capítulo 3. Esta estrategia permite cuantificar la repetibilidad del algoritmo bajo condiciones cuasi-estacionarias, considerando que muestras cercanas en el tiempo corresponden a temperaturas o longitudes de onda similares. El uso de ventanas locales evita que la variabilidad térmica dinámica del Experimento 1 (matriz de $10\,000 \times 10\,000$ espectros) influya artificialmente en la métrica de dispersión, aislando así el ruido instrumental y la estabilidad del algoritmo.

Como se observa en la Figura 4.2, los métodos alcanzan una estimación efectiva de los cambios de temperatura, siguiendo la tendencia de enfriamiento del Experimento 1 descrito en la Sección 3.1.3. Para profundizar en la precisión, se realizan comprobaciones detalladas sobre el comportamiento del error de estimación en función de la temperatura medida, junto con la desviación estándar asociada a dichos datos, tal como se muestra en la Figura 4.3.

Al examinar los gráficos, resulta evidente que la predicción de la temperatura es prácticamente idéntica para ambas implementaciones en C. Si bien se esperaría que existieran diferencias numéricas debido a las sutiles variaciones en la gestión de aritmética de punto flotante entre el código generado

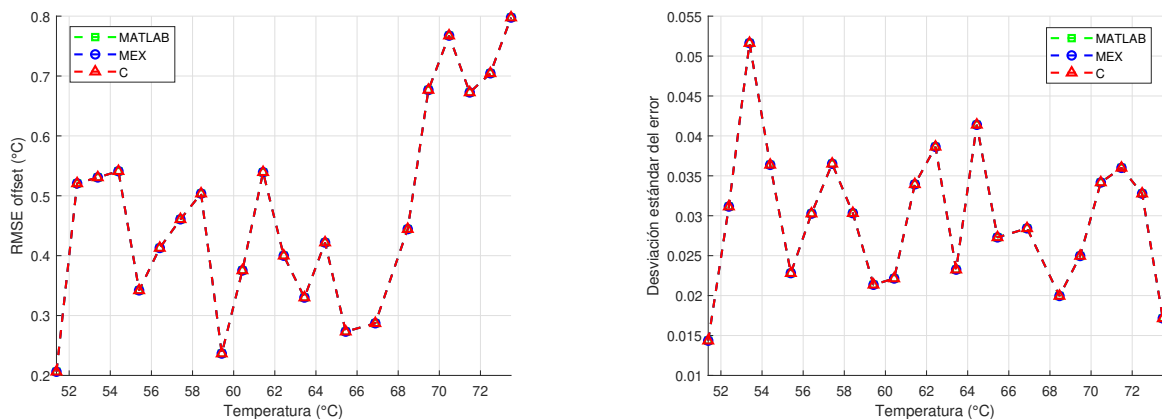


Figure 4.3: Desempeño de las distintas implementaciones de los parámetros de estudio en función de la temperatura.(a) offset RMS. (b) Desviación estándar

y la implementación manual, esta baja desviación en los resultados se debe a que el objetivo final del algoritmo de correlación es la detección de la posición del máximo (corrimiento) y no el valor absoluto de la función de correlación. Dado que la forma del pico de correlación se mantiene similar, la posición del máximo se conserva, explicando la concordancia en los resultados de corrimiento espectral.

Posteriormente, se analiza la posible varianza en los resultados bajo condiciones menos ideales, específicamente al aplicar una atenuación controlada a las mediciones espectrales para simular una degradación de la señal. Los resultados de esta prueba se muestran en la Figura 4.4, correspondiente al Experimento 3 descrito en la Sección 3.1.2, donde se adquieren subconjuntos de 100×10000 espectros a temperaturas estables mientras se varía sistemáticamente la atenuación mediante el VOA.

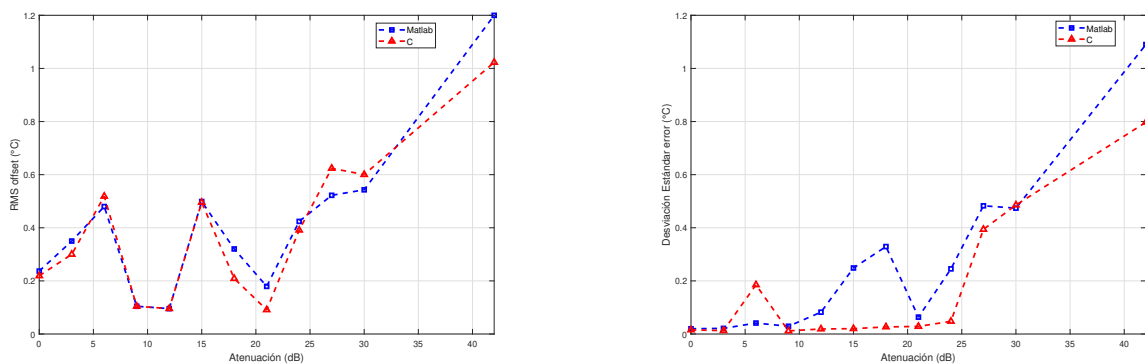


Figure 4.4: Desempeño de las distintas implementaciones de los parámetros de estudio en función de la temperatura.(a) offset RMS. (b) Desviación estándar

La Figura 4.4 revela una mejoría tangible en el Error Cuadrático Medio (*Root Mean Square Error*, RMS) para la implementación directa en C frente a la versión generada por MATLAB Coder, especialmente para niveles de atenuación superiores a 25 dB. Este hallazgo sugiere que, ante condiciones de señal más adversas o con mayor nivel de ruido, el control explícito sobre las rutinas matemáticas y de FFT proporcionado por la implementación manual optimizada ofrece una mayor robustez

en la detección del pico. La consistencia en la desviación estándar entre ambas implementaciones confirma que la estabilidad del algoritmo se preserva independientemente del enfoque de generación de código, validando la portabilidad del método a entornos con recursos limitados.

4.3.2 Análisis de Rendimiento Temporal

El análisis de rendimiento temporal constituye un criterio determinante para validar la viabilidad del algoritmo en sistemas embebidos. En aplicaciones de monitoreo estructural o industrial, el procesamiento de señales debe cumplir con restricciones de tiempo real estrictas: el tiempo de ejecución del algoritmo (t_{exec}) debe ser significativamente menor que el periodo de muestreo (T_m) para evitar pérdida de datos (*buffer overrun*) y dejar margen computacional para tareas auxiliares (gestión de memoria, comunicaciones, periféricos) [11, 27]. Por lo tanto, la comparación temporal no solo mide velocidad, sino que certifica la capacidad del sistema para operar de manera estable bajo carga continua.

Para cuantificar este desempeño, se mide el tiempo promedio de ejecución requerido para calcular la correlación cruzada y detectar el peak espectral en cada una de las implementaciones. La Tabla 4.1 sintetiza los resultados obtenidos, contrastando el entorno de desarrollo (MATLAB), la generación automática de código (MATLAB Coder/MEX) y la implementación manual optimizada en C.

Table 4.1: Comparación de tiempos de ejecución entre MATLAB, MEX y C (valores en microsegundos)

Plataforma	Tiempo promedio (μs)	Desviación estándar (μs)	Factor de mejora
MATLAB	184	155	1.00
MEX (Coder)	216	232.3	0.85
C (manual + FFTW)	42.8	10.5	4.3

Los resultados revelan comportamientos distintivos entre las estrategias de implementación. La versión generada por MATLAB Coder (MEX) presenta un tiempo promedio de $216 \mu s$, lo que representa una degradación del rendimiento (factor 0.85) respecto al código original en MATLAB. Este fenómeno se atribuye al *overhead* introducido por las rutinas genéricas de verificación y gestión de memoria que el Coder inserta automáticamente, las cuales no son optimizadas para el hardware específico [31]. Esto indica que la generación automática de código, si bien facilita la portabilidad, no garantiza eficiencia en tiempo de ejecución sin una configuración manual exhaustiva.

En contraste, la implementación manual en C, utilizando la librería FFTW [30], logra un tiempo promedio de $42.8 \mu s$, equivalente a una mejora de 4.3 veces respecto a MATLAB. Esta optimización se debe al control explícito sobre la aritmética de punto flotante y al uso de algoritmos de FFT altamente eficientes adaptados a la arquitectura del procesador. Considerando que la frecuencia de adquisición del experimento es de 10 Hz ($T_m = 100$ ms), el tiempo de ejecución de $42.8 \mu s$ representa apenas el 0.043 % del periodo disponible. Esta holgura computacional es crítica, ya que valida que el algoritmo no solo es ejecutable, sino que permite la integración de múltiples sensores o protocolos de comunicación en un mismo microcontrolador sin comprometer la estabilidad del sistema en tiempo real [29, 32].

4.4 Análisis de las Implementaciones

El trabajo de migración demuestra exitosamente la portabilidad del algoritmo de detección de picos espectrales, basado en correlación cruzada con transformada de Fourier, desde el entorno

de MATLAB hacia el lenguaje C. Los resultados de precisión obtenidos confirman que ambas implementaciones en C (mex y manual) consiguen estimaciones de temperatura con una precisión comparable a la del código original de MATLAB. Esta concordancia valida la transferencia lógica del procedimiento y asegura que la exactitud del método se mantiene inalterada al cambiar de plataforma de desarrollo. No obstante, la equivalencia en precisión constituye únicamente un requisito necesario; el objetivo fundamental de esta etapa radica en alcanzar una mejora sustancial en la eficiencia computacional para habilitar su despliegue en un sistema embebido con recursos limitados, como un microcontrolador.

La implementación manual en C no solo iguala la precisión, sino que también ofrece un tiempo de ejecución significativamente menor, alcanzando un factor de mejora de 4.3 con respecto al código original de MATLAB. Por el contrario, la implementación generada por MATLAB Coder (mex), aunque escrita en C, resulta inesperadamente más lenta que la versión original debido a rutinas del programa y falta de control sobre las optimizaciones de bajo nivel. La superioridad de la implementación manual se atribuye al control explícito sobre la gestión de memoria y el uso de librerías especializadas de alto rendimiento como FFTW, factores cruciales que el generador automático no logra igualar. En síntesis, se valida la viabilidad técnica de llevar este método de detección a un nivel de aplicación superior en sistemas embebidos, consolidando a la implementación manual en C como la candidata más robusta y eficiente para un futuro prototipo en embebido.

Capítulo 5

Conclusiones

El presente trabajo de titulación tuvo como objetivo principal el desarrollo, validación e implementación de un algoritmo de detección de peak espectral para sensores de fibra óptica basados en rejillas de Bragg (FBG), utilizando correlación cruzada implementada mediante la Transformada Rápida de Fourier (FFT). A lo largo de esta memoria, se analizaron cinco variantes del algoritmo, evaluando su precisión, robustez frente a condiciones adversas y eficiencia computacional, para posteriormente migrar la solución óptima a lenguaje C con fines de portabilidad a sistemas embebidos. En la etapa de implementación en MATLAB, se demostró que los métodos basados en FFT ofrecen una ventaja computacional significativa frente a la correlación cruzada directa, reduciendo la complejidad algorítmica de $\mathcal{O}(N^2)$ a $\mathcal{O}(N \log N)$ [12, 33]. De las variantes estudiadas, el método de correlación cruzada mediante FFT con ajuste polinomial cuadrático presentó el mejor desempeño global, logrando un error cuadrático medio (RMS) inferior a 0.5°C y una desviación estándar menor a 0.05°C en la estimación de temperatura. Además, se validó la robustez del algoritmo frente a niveles de atenuación óptica de hasta 20 dB, manteniendo una precisión aceptable para aplicaciones de monitoreo estructural, lo que confirma la capacidad de la correlación cruzada para realzar componentes coherentes y suprimir ruido aditivo [3, 34].

Respecto a la migración del algoritmo a lenguaje C, se compararon dos enfoques: la generación automática de código mediante MATLAB Embedded Coder y una implementación manual optimizada utilizando la librería FFTW [13, 30]. Los resultados evidenciaron que la implementación manual en C es superior en términos de rendimiento, alcanzando una mejora en el tiempo de ejecución de 4.3 veces respecto al entorno original de MATLAB. Por el contrario, el código generado automáticamente (MEX) presentó un overhead computacional que resultó en un desempeño incluso inferior al de MATLAB en ciertas condiciones, debido a rutinas genéricas de verificación y gestión de memoria. A pesar de las diferencias temporales, ambas implementaciones en C conservaron la precisión de detección del algoritmo original, validando la integridad numérica del proceso de migración.

En conclusión, se ha demostrado la viabilidad técnica de utilizar la correlación cruzada mediante FFT como método de interrogación para sensores FBG, equilibrando alta precisión y baja latencia. La implementación manual en C con FFTW se consolida como la estrategia más eficiente para un futuro despliegue en hardware embebido, ya que cumple holgadamente con los requisitos de tiempo real (ejecución en $\approx 43 \mu\text{s}$ frente a un periodo de muestreo de 100 ms). Si bien este trabajo se limitó a la validación del código en entorno de escritorio, los resultados obtenidos sentaron las bases necesarias para la integración del algoritmo en microcontroladores o plataformas FPGA, abriendo la puerta al desarrollo de interrogadores de bajo costo y alta eficiencia para aplicaciones distribuidas.

Bibliografía

- [1] A. D. Kersey *et al.*, “Fiber grating sensors,” *Journal of Lightwave Technology*, vol. 15, no. 8, pp. 1442–1463, 1997.
- [2] A. Othonos, K. Kyriacos, P. David, and A. Mugnier, *Fibre Bragg Gratings*. Springer Series in Optical Sciences, Springer, 2006.
- [3] D. Tosi, “Review and analysis of peak tracking techniques for fiber bragg grating sensors,” *Sensors*, vol. 17, no. 10, p. 2368, 2017.
- [4] T. Erdogan, “Fiber grating spectra,” *Journal of Lightwave Technology*, vol. 15, pp. 1277–1294, 1997.
- [5] G. Meltz and W. Morey, “Bragg grating formation and germanosilicate fiber photosensitivity,” in *International Workshop on Photoinduced Self-Organization Effects in Optical Fiber*, 1991.
- [6] H. Patrick and S. Gilbert, “Growth of bragg gratings produced by continuous-wave ultraviolet light in optical fiber,” *Optics Letters*, vol. 18, pp. 1484–1486, 1993.
- [7] J. Dakin, D. Pratt, G. Bibby, and J. Jones, “Fiber optic sensors based on the Bragg grating technique,” *Electron. Lett.*, vol. 21, no. 13, pp. 566–567, 1985.
- [8] S. Rashleigh, “Measurement of very small temperature changes in Bragg fiber gratings,” *Opt. Lett.*, vol. 11, no. 3, pp. 193–195, 1986.
- [9] T. Bodendorfer, M. Muller, F. Hirth, and A. Koch, “Comparison of different peak detection algorithms with regards to spectrometric fiber bragg grating interrogation systems,” in *Proceedings of International Symposium on Optomechatronic Technologies (ISOT)*, (Istanbul, Turkey), pp. 21–23, 2009.
- [10] C. Atkins, M. Putnam, and E. Friebele, “Instrumentation for interrogating many-element fiber bragg grating arrays,” in *Proc. SPIE*, vol. 2444, pp. 257–267, 1995.
- [11] A. Lamberti, S. Vanlanduit, B. de Pauw, and F. Berghmans, “A novel fast phase correlation algorithm for peak wavelength detection of fiber bragg grating sensors,” *Opt. Express*, vol. 22, pp. 7099–7112, 2014.
- [12] A. V. Oppenheim and R. W. Schaffer, *Discrete-Time Signal Processing*. Prentice Hall, 3rd ed., 2010.
- [13] M. Frigo and S. G. Johnson, “Fastest fourier transform in the west (fftw) library,” 2023. [Software].

-
- [14] STMicroelectronics, *STM32H743xI Advanced Microcontroller Datasheet*, 2022. Disponible en línea.
 - [15] E. Systems, *ESP32 Technical Reference Manual*, 2023. Disponible en línea.
 - [16] Y. Liu, J. Williams, L. Zhang, and I. Bennion, “Abnormal spectral evolution of fibre bragg gratings in hydrogenated fibres,” *Optics Letters*, vol. 27, pp. 586–588, 2002.
 - [17] J. L. Archambault, L. Reekie, and P. Russell, “High reflectivity and narrow bandwidth fibre gratings written by single excimer pulse,” *Electronics Letters*, vol. 29, pp. 28–29, 1993.
 - [18] I. Riant and F. Haller, “Study of the photosensitivity at 193 nm and comparison with photosensitivity at 240 nm influence of fiber tension: type iia aging,” *Journal of Lightwave Technology*, vol. 15, pp. 1464–1469, 1997.
 - [19] K. Hill *et al.*, “Chirped in-fiber bragg grating dispersion compensators: Linearization of the dispersion characteristic and demonstration of dispersion compensation in a 100 km, 10 gbit/s optical fiber link,” *Electronics Letters*, vol. 30, pp. 1755–1756, 1994.
 - [20] K. Hill, Y. Fujii, D. Johnson, and B. Kawasaki, “Photosensitivity in optical fiber waveguides: Application to reflection filter fabrication,” *Applied Physics Letters*, vol. 32, pp. 647–649, 1978.
 - [21] G. Meltz, W. Morey, and W. Glenn, “Formation of bragg gratings in optical fibers by a transverse holographic method,” *Optics Letters*, vol. 14, pp. 823–825, 1989.
 - [22] K. Hill, B. Malo, F. Bilodeau, F. Johnson, and J. Albert, “Bragg gratings fabricated in monomode photosensitive optical fiber by uv exposure thorough a phase-mask,” *Applied Physics Letters*, vol. 62, pp. 1035–1037, 1993.
 - [23] B. Malo, K. Hill, F. Bilodeau, D. Johnson, and J. Albert, “Point-by-point fabrication of micro-bragg gratings in photosensitive fiber using single excimer pulse refractive index modification techniques,” *Electronics Letters*, vol. 29, pp. 1668–1669, 1993.
 - [24] W. Chen and A. Vallan, “Applications of a fast fbg interrogation system for real-time thermal and structural monitoring,” in *Proceedings of the IEEE 1st International Forum on Research and Technologies for Society and Industry: Leveraging a Better Tomorrow (RTSI)*, pp. 388–391, 2015.
 - [25] D. Tosi, “Klt-based algorithm for sub-pm accurate fbg tracking with coarse wavelength sampling,” *Photonics Technol. Lett.*, vol. 27, pp. 2134–2137, 2015.
 - [26] J. Wang, “An intuitive tutorial to gaussian processes regression,” 2021.
 - [27] J. Gong, J. MacAlpine, C. Chan, W. Jin, M. Zhang, and Y. Liao, “A novel wavelength detection technique for fiber bragg grating sensors,” *IEEE Photonics Technol. Lett.*, vol. 14, pp. 678–680, 2002.
 - [28] J. G. Proakis and D. G. Manolakis, *Digital Signal Processing: Principles, Algorithms, and Applications*. Upper Saddle River, NJ, USA: Prentice-Hall International, Inc., 3rd ed., 1996.
 - [29] A. Paterno, J. Silva, M. Milczewski, L. Arruda, and H. Kalinowski, “Radial-basis function network for the approximation of fbg sensor spectra with distorted peaks,” *Meas. Sci. Technol.*, vol. 17, pp. 1039–1045, 2006.
-

- [30] MathWorks, *xcorr: Cross-correlation*, 2023. Documentación en línea.
- [31] D. Tosi, “Improved klt algorithm for high-precision wavelength tracking of optical fiber bragg grating sensors,” *J. Sens.*, vol. 2017, 2017.
- [32] Y. Liu, H. Marques, J. Pinto, and J. Santos, “Temperature dependence of Bragg wavelength in uniform fiber gratings,” *J. Lightwave Technol.*, vol. 25, no. 1, pp. 26–30, 2007.
- [33] K. R. Rao, D. N. Kim, and J. J. Hwang, *Fast Fourier Transform: Algorithms and Applications*. Dordrecht, Heidelberg, London, New York: Springer, 2010.
- [34] C. Caucheteur, K. Chah, F. Lhomme, M. Blondel, and P. Megret, “Autocorrelation demodulation technique for fiber bragg grating sensor,” *IEEE Photonics Technol. Lett.*, vol. 16, pp. 2320–2322, 2004.