

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA  
DEPARTAMENTO DE ELECTRÓNICA  
VALPARAÍSO - CHILE



**“ANÁLISIS COMPARATIVO DE MÉTODOS DE  
DESENVOLVIMIENTO DE FASE PARA SENSORES  
ACÚSTICOS DISTRIBUIDOS EN FIBRA ÓPTICA”**

**FELIPE GONZALO MEJÍAS OSORIO**

MEMORIA DE TITULACIÓN PARA OPTAR AL TÍTULO DE INGENIERO CIVIL  
ELECTRÓNICO

PROFESOR GUÍA : MARCELO SOTO  
PROFESOR CORREFERENTE : MELISSA DIAGO

JULIO - 2025



## 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:** Análisis Comparativo de Métodos de Desarrollo de Fase para Sensores Acústicos Distribuidos en Fibra Óptica

**Nombre del candidato(a):** Felipe Gonzalo Mejías Osorio

**Carrera / Grado:** Ing. Civil Electrónica

**Campus:** Casa Central Valparaíso; **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

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:

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:** 20.10.2025

**Firma:** 

**Estudiante o Candidato(a):**

**Fecha:** 20.10.2025

**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 expresar mis agradecimientos a mi familia, amigos y todas las personas que me apoyaron y ayudaron durante toda esta etapa y desarrollo de esta memoria.

A mis padres, Claudia y Jaime, por apoyarme siempre y darme la oportunidad de llegar hasta aquí. A mi pareja, Mariel, por estar ahí en todo este proceso, por creer siempre en mí, escucharme, darme perspectiva, apoyarme y alentarme en los momentos que más lo necesitaba.

A mis amigos y compañeros, porque nunca faltaron las risas, y por compartir conmigo durante toda la carrera.

También quiero dar las gracias a Sebastián por su ayuda y apoyo en el desarrollo experimental de este trabajo. Finalmente, agradecer a mi profesor guía, Marcelo Soto, por darme la oportunidad de desempeñarme en este campo que se encontraba fuera de mi área.

También quiero agradecer a los siguientes proyectos por su apoyo financiero y por hacer posible la parte experimental de esta memoria:

- FONDECYT Regular 1241085, ANID
- FONDEQUIP EQM180226, ANID
- FONDEQUIP EQM220113, ANID
- Basal AFB240002, ANID
- Global X-Challenge N629092412124, Office of Naval Research (ONR), USA



UNIVERSIDAD TÉCNICA  
FEDERICO SANTA MARÍA



# “Análisis Comparativo de Métodos de Desarrollo de Fase para Sensores Acústicos Distribuidos en Fibra Óptica”

**Felipe Gonzalo Mejías Osorio**

Memoria para Optar al Título de Ingeniero Civil Electrónico

**Universidad Técnica Federico Santa María**

Profesor Guía: Marcelo Soto

Profesor Correfente: Melissa Diago

Julio - 2025

## **Resumen**

Este trabajo tiene como principal objetivo implementar y comparar algoritmos de desarrollo de fase provenientes de distintas áreas y aplicarlos al sensado acústico distribuido (DAS) en fibra óptica, trabajando con datos reales obtenidos de un esquema de reflectometría óptica en el dominio temporal sensible a fase ( $\Phi$ -OTDR) heterodino implementado en laboratorio.

El uso de la tecnología DAS permite medir perturbaciones en una fibra óptica a través de largas distancias con una alta precisión a partir de la fase de la luz retrodispersada dentro de la misma fibra.

Los métodos implementados en este trabajo son modificados en caso de ser necesario para traducir su rendimiento al caso de trazas obtenidas del sensado DAS. El objetivo es comparar diversos métodos, con el fin de encontrar métodos desde distintas áreas que sean generalizables a tecnología DAS que sean robustos, flexibles y computacionalmente eficientes. En este trabajo se revisan métodos provenientes de tecnología de radar de apertura sintética, procesamiento de imágenes, filtros de Kalman y redes neuronales.

**Palabras claves:** *Desarrollo de fase, sensores acústicos distribuidos, reflectometría óptica en el dominio temporal sensible a fase.*



UNIVERSIDAD TÉCNICA  
FEDERICO SANTA MARÍA



# “Comparative Analysis of Phase Unwrapping Methods for Distributed Acoustic Sensors in Fiber Optics”

**Felipe Gonzalo Mejías Osorio**

Final Project Report towards the fulfillment of the Ingeniero Civil Electrónico, Minor in Computers degree (6 year program).

**Universidad Técnica Federico Santa María**

Advisor: Marcelo Soto

Co-Advisor: Melissa Diago

July - 2025

## **Abstract**

The main objective of this work is to implement and compare phase unwrapping algorithms from different areas and apply them to optical fiber Distributed Acoustic Sensor (DAS) data, using real data obtained from a heterodyne phase sensitive optical time domain reflectometry ( $\Phi$ -OTDR) scheme implemented in a laboratory.

The use of DAS allows for the measurement of disturbances in an optical fiber over long distances with high precision by measuring the phase of the backscattered light within the optical fiber.

The unwrapping methods that are implemented in this work are modified if necessary to translate their performance from their respective field to DAS data. The goal is to compare a diverse set of methods, and find methods from different areas that are applicable to DAS data that are robust, flexible and computationally efficient. In this work the methods reviewed come from uses like synthetic aperture radar, image processing, Kalman filtering and neural networks.

**Keywords:** *Phase unwrapping, distributed acoustic sensing, phase sensitive optical time domain reflectometry.*

# Índice general

<b>1. Introducción .....</b>	<b>1</b>
1.1 Motivación.....	1
1.2 Objetivos.....	2
1.3 Estructura del Documento .....	2
<b>2. Marco Teórico .....</b>	<b>3</b>
2.1 Esparcimiento en fibras ópticas.....	3
2.1.1 Esparcimiento y retrodispersión Rayleigh .....	4
2.1.2 Reflectometría óptica en el dominio temporal (OTDR) .....	5
2.1.3 OTDR sensible a fase ( $\Phi$ -OTDR).....	6
2.1.4 Sensado distribuido de vibración (DVS) .....	10
2.1.5 Sensado distribuidos acústicos (DAS).....	10
2.1.6 Métricas relevantes en el funcionamiento de un sensor DAS .....	15
2.2 Desenvolvimiento de fase.....	21
2.3 Métodos de desenvolvimiento de fase.....	23
2.3.1 Método <i>Unwrap</i> de Matlab .....	24
2.3.2 Método basado en Ecuación de Transporte de Intensidad (TIE) .....	24
2.3.3 Filtro de Kalman.....	25
2.3.4 Métodos basados en Machine Learning .....	27
<b>3. Implementación experimental .....</b>	<b>32</b>
3.1 Métodos de desenvolvimiento de fase.....	32
3.1.1 <i>Unwrap</i> de MATLAB .....	32
3.1.2 Basado en ecuación de transporte de intensidad (TIE) .....	32
3.1.3 Filtro de Kalman <i>Unscented</i> .....	32
3.1.4 Red Neuronal U-Net .....	34
3.1.5 Red Neuronal LSTM.....	34
3.1.6 Datos de entrenamiento para métodos de machine learning.....	34

3.1.7	Estrategias de desenvolvimiento de fase .....	35
3.2	Esquema $\Phi$ -OTDR heterodino.....	36
3.2.1	Oscilador local.....	36
3.2.2	Generación de pulsos.....	38
3.2.3	Esquema implementado .....	39
3.3	Mediciones experimentales .....	40
3.4	Procesamiento de datos .....	40
<b>4.</b>	<b>Resultados.....</b>	<b>41</b>
4.1	Datos Obtenidos.....	41
4.1.1	Traza sin perturbación.....	42
4.1.2	Traza con perturbación .....	43
4.2	Desenvolvimiento de fase con datos experimentales sin perturbación .....	43
4.3	Desenvolvimiento de fase con datos experimentales con perturbación.....	48
4.4	Downsampling/Dropout .....	60
4.5	Rendimiento computacional .....	68
<b>5.</b>	<b>Conclusiones .....</b>	<b>69</b>

# Índice de figuras

2.1	Diagrama efecto de esparcimiento [6].	3
2.2	Espectro de luz retrodispersada en fibra óptica. [7]	4
2.3	Esquema OTDR clásico [10].	5
2.4	Traza OTDR típica [11].	6
2.5	Modelo de fibra con y sin perturbación, a la derecha e izquierda respectivamente [6].	7
2.6	Traza típica obtenida con esquema $\Phi$ -OTDR.	9
2.7	Diagrama de un esquema de detección heterodino típico [5].	12
2.8	a) Fase acumulada a lo largo de una fibra a través del tiempo, y b) Fase diferencial a lo largo de una fibra en el tiempo [5]	14
2.9	Desfase de 90 logrado con híbrida óptica [16].	15
2.10	Distribución de probabilidad de las intensidades en una traza $\Phi$ -OTDR [11].	19
2.11	Funcionamiento de SERVS [38].	20
2.12	Precisión de fase experimental y calculada a lo largo de una fibra interrogada con un pulso de 800 ns [41].	21
2.13	Fase diferencial envuelta y desenvuelta obtenida en una medición con esquema heterodino [5].	22
2.14	Desenvolvimiento de fase para matrices con función incluida en MATLAB [3].	24
2.15	Arquitectura de red U-Net propuesta por Ronneberger, et al [66]. Los números en la parte inferior de cada caja representan el tamaño de la imagen en esa etapa y los de la parte superior indican la cantidad de canales, mientras que las flechas denotan operaciones descritas en la parte inferior derecha.	28
2.16	Arquitectura de la red neuronal propuesta por Wang [65]. Los números en la parte inferior de cada caja representan el tamaño de la imagen en esa etapa y los de la parte superior indican la cantidad de canales, mientras que las flechas denotan operaciones descritas en la parte inferior derecha.	29

2.17	Bloque fundamental LSTM clásico. $\times$ indica multiplicación, $+$ suma elemento a elemento, $\sigma$ la función de activación sigmoide, $g$ la función de activación de entrada y $h$ la de salida. ....	30
3.1	Oscilador local del esquema $\Phi$ -OTDR heterodino. ....	36
3.2	Evolución del espectro de la señal óptica a medida que pasa por el oscilador local. Las flechas corresponden a los espectros ideales, y las líneas punteadas indican filtrado. ....	37
3.3	Sección del esquema para generar pulsos interrogadores. ....	38
3.4	Sincronización de pulsos del SOA y EOM. El EOM modula el pulso ya modulado por el SOA para mejorar la razón de extinción. ....	38
3.5	Esquema completo de detección $\Phi$ -OTDR heterodino. Las líneas sólidas indican señales ópticas, y las punteadas señales eléctricas. ....	39
3.6	Flujo de procesamiento de datos adquiridos. ....	40
4.1	Organización temporal y espacial de las mediciones. Las trazas verdes representan una traza a lo largo de la fibra, y los puntos que pasan por la línea roja punteada representan los datos asociados a una misma posición a través del tiempo. ....	41
4.2	SNR's eléctricas para potencias ópticas de entrada a) 20 dB, b) 15 dB, c) 10 dB, d) 5 dB ....	42
4.3	Diagrama simplificado del esquema utilizado para realizar mediciones con perturbación en la fibra. ....	43
4.4	Desenvolvimiento de fase en sentido espacial para señal de pulso 20 dBm. ....	44
4.5	Desenvolvimiento de fase en sentido temporal para señal con pulso de 20 dBm. ..	44
4.6	Desenvolvimiento de fase en sentido espacial para señal de pulso 10 dBm. ....	45
4.7	Desenvolvimiento de fase en sentido temporal para señal de pulso 10 dBm. ....	46
4.8	Resultados de desenvolvimiento cuasi-bidimensional usando unwrap de MATLAB sobre la señal con pulso de 20 dBm. a) corresponde a una traza desenvuelta, mientras que b) a un punto en la fibra a través del tiempo. ....	46
4.9	Resultados de desenvolvimiento cuasi-bidimensional usando UFK sobre la señal con pulso de 20 dBm. a) corresponde a una traza desenvuelta, mientras que b) a un punto en la fibra a través del tiempo. ....	47
4.10	Resultados de desenvolvimiento cuasi-bidimensional usando unwrap de MATLAB sobre la señal con pulso de 10 dBm. a) corresponde a una traza desenvuelta, mientras que b) a un punto en la fibra a través del tiempo. ....	47

4.11	Resultados de desenvolvimiento cuasi-bidimensional usando UFK sobre la señal con pulso de 10 dBm. a) corresponde a una traza desenvuelta, mientras que b) a un punto en la fibra a través del tiempo. ....	48
4.12	Señal temporal rescatada utilizando <i>unwrap()</i> en sentido espacial para la señal de 20Hz a 20 dBm de potencia de entrada a la fibra. ....	49
4.13	Señal temporal rescatada utilizando <i>unwrap()</i> en sentido temporal para la señal de 20 Hz a 20 dBm de potencia de entrada a la fibra. ....	49
4.14	Señal rescatada utilizando TIE en sentido espacial para la señal de 20 Hz a 20 dBm de potencia de entrada a la fibra. ....	50
4.15	Señal rescatada utilizando UKF en sentido espacial para la señal de 20 Hz a 20 dBm de potencia de entrada a la fibra. ....	50
4.16	Señal rescatada utilizando UKF en sentido temporal para la señal de 20 Hz a 20 dBm de potencia de entrada a la fibra. ....	51
4.17	Señal rescatada utilizando LSTM en sentido espacial para la señal de 20 Hz a 20 dBm de potencia de entrada a la fibra. ....	51
4.18	Señal rescatada utilizando LSTM en sentido temporal para la señal de 20 Hz a 20 dBm de potencia de entrada a la fibra. ....	52
4.19	Señal rescatada utilizando <i>unwrap()</i> en sentido espacial para la señal de 200 Hz a 20 dBm de potencia de entrada a la fibra. ....	52
4.20	Señal rescatada utilizando <i>unwrap()</i> en sentido temporal para la señal de 200 Hz a 20 dBm de potencia de entrada a la fibra. ....	53
4.21	Señal rescatada utilizando TIE en sentido espacial para la señal de 200 Hz a 20 dBm de potencia de entrada a la fibra. ....	53
4.22	Señal rescatada utilizando UKF en sentido espacial para la señal de 200 Hz a 20 dBm de potencia de entrada a la fibra. ....	54
4.23	Señal rescatada utilizando UKF en sentido temporal para la señal de 200 Hz a 20 dBm de potencia de entrada a la fibra. ....	54
4.24	Señal rescatada utilizando LSTM en sentido espacial para la señal de 200 Hz a 20 dBm de potencia de entrada a la fibra. ....	55
4.25	Señal rescatada utilizando LSTM en sentido temporal para la señal de 200 Hz a 20 dBm de potencia de entrada a la fibra. ....	55
4.26	Señal rescatada utilizando <i>unwrap()</i> , primero en sentido espacial y luego temporal para la señal de 20 Hz a 20 dBm de potencia de entrada a la fibra. ....	56
4.27	Señal rescatada utilizando <i>unwrap()</i> , primero en sentido temporal y luego espacial para la señal de 20 Hz a 20 dBm de potencia de entrada a la fibra. ....	56

4.28 Señal rescatada utilizando UKF, primero en sentido espacial y luego temporal para la señal de 20 Hz a 20 dBm de potencia de entrada a la fibra.....	57
4.29 Señal rescatada utilizando UKF, primero en sentido temporal y luego espacial para la señal de 20 Hz a 20 dBm de potencia de entrada a la fibra.....	57
4.30 Señal rescatada utilizando LSTM, primero en sentido espacial y luego temporal para la señal de 20 Hz a 20 dBm de potencia de entrada a la fibra.....	58
4.31 Señal rescatada utilizando LSTM, primero en sentido temporal y luego espacial para la señal de 20 Hz a 20 dBm de potencia de entrada a la fibra.....	58
4.32 Fase diferencial desenvuelta con MATLAB y la señal resultante al suavizar con media móvil de tamaño 20. ....	59
4.33 Fase diferencial desenvuelta con UKF y la señal resultante al suavizar con media móvil de tamaño 20. ....	59
4.34 Fase diferencial desenvuelta con UKF y la señal resultante al suavizar con media móvil de tamaño 5. ....	60
4.35 Fase diferencial desenvuelta con LSTM y la señal resultante al suavizar con media móvil de tamaño 5. ....	60
4.36 Señales de referencia (sin dropout) para comparación de dropout. Perturbaciones obtenidas para potencia de entrada 20 dBm, con frecuencia a) 20 Hz y b) 200 Hz.	62
4.37 Resultados de desenvolvimiento con perturbación de 20Hz y 10 % dropout. a) MATLAB, b) TIE, c) UKF, d) LSTM. ....	62
4.38 Resultados de desenvolvimiento con perturbación de 20Hz y 30 % dropout. a) MATLAB, b) TIE, c) UKF, d) LSTM. ....	63
4.39 Resultados de desenvolvimiento con perturbación de 20Hz y 50 % dropout. a) MATLAB, b) TIE, c) UKF, d) LSTM. ....	64
4.40 Resultados de desenvolvimiento con perturbación de 20Hz y 10 % dropout. a) MATLAB, b) TIE, c) UKF, d) LSTM. ....	65
4.41 Resultados de desenvolvimiento con perturbación de 20Hz y 30 % dropout. a) MATLAB, b) TIE, c) UKF, d) LSTM. ....	66
4.42 Resultados de desenvolvimiento con perturbación de 20Hz y 50 % dropout. a) MATLAB, b) TIE, c) UKF, d) LSTM. ....	67

# Índice de tablas

4.1	Error cuadrático medio (en radianes) para cada método y tasa de dropout. ....	67
4.2	Tiempo medio de cómputo de fase desenvuelta para los distintos métodos. (*: Para TIE solo se consideran el cómputo en sentido espacial).....	68

# 1 Introducción

## 1.1 Motivación

El desenvolvimiento de fase es un paso clave en procesos de medición donde los datos se obtienen con módulo  $2\pi$ , como imágenes de ultrasonido [1] hasta modelamiento geofísico [2]. Este proceso corresponde a llevar una señal de fase de estar acotada entre  $-\pi$  y  $\pi$  a una señal de fase continua sin acotación. En condiciones ideales, esto corresponde simplemente a detectar saltos de  $2\pi$  en la señal envuelta y compensar sumando o restando  $2\pi$  a la señal final. En la práctica rara vez resulta así de simple, ya que existe un gran número de factores en cada paso del sensado, muestreado y procesado de los datos que introducen ruido a la señal captada. Es por esto que resulta necesario contar con métodos para obtener la fase desenvuelta que sean robustos ante los entornos sub-óptimos de detección. Existen una variedad de técnicas que se utilizan actualmente para el desenvolvimiento de fase, desde técnicas básicas como el caso ideal ya mencionado [3], hasta técnicas avanzadas que utilizan redes neuronales para llevar a cabo la tarea [4].

Una aplicación que utiliza desenvolvimiento de fase y que se ha vuelto muy relevante es el sensado distribuido acústico, o DAS por sus siglas en inglés. Este tipo de sensado utiliza las propiedades físicas de las fibras ópticas para medir temperatura y vibraciones de manera muy precisa, y sobre largas distancias. Destaca ya que es inmune a ruido electromagnético, lo que le da la posibilidad de medir donde sensores electrónicos clásicos fallan. El desenvolvimiento de fase es clave para esta técnica, ya que la información que permite caracterizar de manera precisa las perturbaciones que afectan a la fibra viene dada por la fase sensada. Si el desenvolvimiento falla, cualquier procesado que venga después será en vano, ya que los datos vendrán inutilizables desde el principio.

Existen muchos y variados métodos de desenvolvimiento de fase aplicados a una amplia gama de campos. Sin embargo, los métodos desarrollados suelen ser creados para usos específicos, y no se prueban en otros casos donde podrían ser útiles. El objetivo de este trabajo es implementar y evaluar el desempeño de una serie de métodos de desenvolvimiento de distintas disciplinas, aplicados con un enfoque a datos reales capturados por un esquema DAS, para así determinar métodos que se puedan generalizar más allá del uso específico para el que fueron diseñados.

### 1.2 Objetivos

El objetivo principal de este proyecto es comparar la efectividad y robustez de algoritmos de desenvolvimiento de fase aplicados a DAS en fibra óptica, buscando presentar de manera concisa las fortalezas y debilidades de cada método en distintos escenarios. Para lograr esto se consideran los siguientes objetivos secundarios:

- Diseñar e implementar pruebas sobre el esquema DAS que permitan caracterizar de manera robusta los métodos de desenvolvimiento de fase.
- Obtener datos experimentales con un esquema de detección  $\Phi$ -OTDR implementado el laboratorio.
- Implementar y adaptar distintos métodos de desenvolvimiento de fase para ser usados en DAS.
- Obtener una comparativa sólida de los métodos probados que permita elegir de forma informada un método de desenvolvimiento según las necesidades de alguna investigación futura.

### 1.3 Estructura del Documento

La información presentada en esta memoria se organiza en los siguientes capítulos:

- **Capítulo 2:** Se presentan los antecedentes y bases teóricas del sensado DAS, esquemas  $\Phi$ -OTDR y desenvolvimiento de fase.
- **Capítulo 3:** Se describen las implementaciones de los distintos métodos de desenvolvimiento de fase, y los detalles asociados.
- **Capítulo 4:** Se describe la metodología para obtener datos experimentales, y el flujo del procesamiento de datos.
- **Capítulo 5:** Se presentan los resultados obtenidos.
- **Capítulo 6:** Se discuten conclusiones e ideas para trabajos futuros.

## 2 Marco Teórico

### 2.1 Esparcimiento en fibras ópticas

Las fibras ópticas generalmente se componen de un núcleo de vidrio por el cual se propaga la luz, un revestimiento que permite la reflexión total interna de forma de mantener la luz dentro del núcleo, y un recubrimiento que protege la fibra y entrega mayor integridad estructural. A pesar de ser un material continuo, el núcleo de una fibra óptica es un material heterogéneo debido a diversos aspectos propios de la fabricación del material, como lo son variaciones de densidad a lo largo de la fibra e inclusiones de materiales dopantes en la misma [5]. Cuando el tamaño de estas imperfecciones es comparable a la longitud de onda de la señal lumínica enviada a través de la fibra se producen interacciones entre la onda electromagnética y la fibra a nivel molecular, por los cambios en el índice de refracción de la fibra debido a las imperfecciones, produciendo múltiples reflexiones en distintas direcciones.

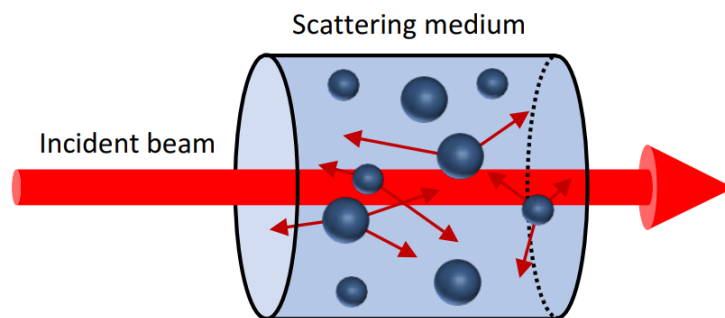


Fig. 2.1: Diagrama efecto de esparcimiento [6].

Las reflexiones que se producen pueden ser clasificadas en dos tipos, dependiendo de si inducen cambios de frecuencia óptica (longitud de onda) en la luz reflejada: elástica e inelástica. El esparcimiento elástico se caracteriza por producir luz reflejada con la misma frecuencia óptica que la del rayo incidente, es decir, no hay pérdida o ganancia de energía. En esta categoría encontramos el esparcimiento de Rayleigh (RBS por sus siglas en inglés). Por el otro lado, el

esparcimiento elástico produce luz con una frecuencia distinta a la incidente, es decir, hay una pérdida o ganancia de energía en la interacción con el medio. En esta categoría encontramos el esparcimiento de Raman y de Brillouin. En la figura 2.2 se muestra el espectro óptico de los efectos de esparcimiento. El esparcimiento Rayleigh tiene la mayor potencia, y mantiene la frecuencia de la señal incidente, mientras que los esparcimientos inelásticos de Raman y Brillouin producen reflexiones de menor potencia y desplazadas en frecuencia en 11 GHz y a 13.2 THz, respectivamente.

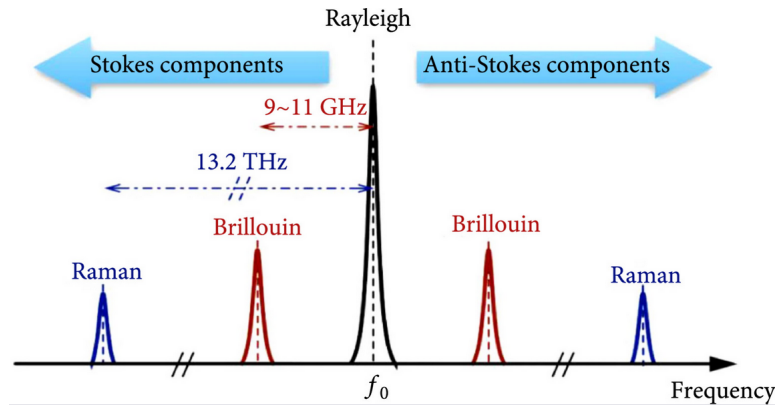


Fig. 2.2: Espectro de luz retrodispersada en fibra óptica. [7]

### 2.1.1 Esparcimiento y retrodispersión Rayleigh

El esparcimiento Rayleigh corresponde a un esparcimiento elástico, generando rayos reflejados con la misma longitud de onda del rayo incidente. Como se comentó anteriormente, este fenómeno se debe a las interacciones entre fluctuaciones de densidad del medio y la onda electromagnética que se propaga a través de él. El esparcimiento Rayleigh ocurre cuando el tamaño de las heterogeneidades es del orden de  $1/10$  de la longitud de onda del rayo incidente [5]. Este esparcimiento es causa de la mayor pérdida de potencia en la fibra, generando atenuación proporcionalmente inversa a la cuarta potencia de la longitud de onda,  $\lambda^{-4}$  [7].

Históricamente, se ha trabajado en reducir el esparcimiento de Rayleigh en fibras ópticas, dado que inducen la mayor pérdida de potencia en la señal enviada, sin embargo, para algunas aplicaciones esta luz puede entregar información relevante acerca de la fibra. El esparcimiento Rayleigh crea reflexiones en todos los sentidos. Parte de esta luz se refleja en dirección opuesta a la de la propagación de la onda se denomina retrodispersión de Rayleigh. Como esta luz se propaga en dirección opuesta puede ser capturada y medida en la misma punta de la fibra de donde se inyecta la luz incidente. Esto sienta la base para la reflectometría óptica en el dominio del tiempo (OTDR por sus siglas en inglés) [8].

Las técnicas de OTDR varían en sus implementaciones y en la información que permiten extraer

de la retrodispersión, y serán explicadas a continuación.

### 2.1.2 Reflectometría óptica en el dominio temporal (OTDR)

De la forma más simple, la OTDR consiste en la inyección de un pulso interrogador de luz incoherente en la fibra y la medición de la retrodispersión o backscattering producida por éste. Fue presentada en 1976 por Barnoski et al con el fin de caracterizar la atenuación de la fibra y detectar discontinuidades en fibras ópticas sin necesidad de acceder a ambas puntas de ésta. [9]. El pulso incidente viaja a través de la fibra, perdiendo potencia por esparcimiento y generando una señal retrodispersada. Esta señal a su vez pierde potencia al propagarse en sentido inverso por la fibra, y es captada por un fotodetector al inicio de la fibra y transformada a un voltaje para su digitalización y procesamiento. En la figura 2.3 se muestra un esquema OTDR simple.

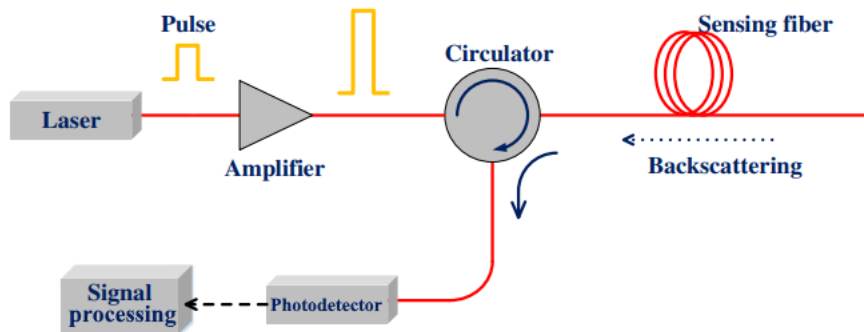


Fig. 2.3: Esquema OTDR clásico [10].

La señal captada por el OTDR corresponde a la potencia óptica recibida producto del backscatter. A medida que se avanza en la fibra, la señal recibida por la retrodispersión de Rayleigh tiene menor potencia, ya que al recorrer mayor largo de fibra se pierde más energía tanto del pulso incidente como de la señal retrodispersada [11]. En la figura 2.4 se observa una curva OTDR típica. En la curva medida se observan como discontinuidades y conexiones en la fibra cambios en la potencia recibida en distintos puntos.

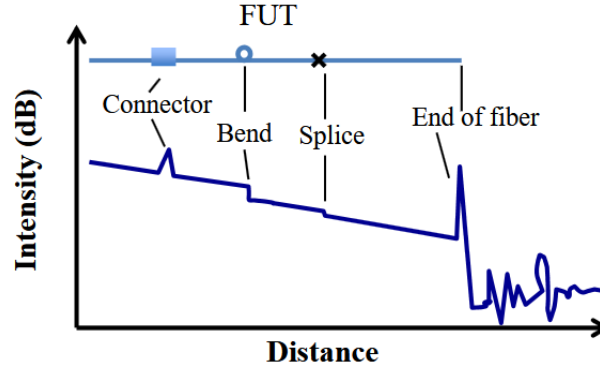


Fig. 2.4: Traza OTDR típica [11].

Además, la pendiente de la curva OTDR representa el doble del factor de pérdida, ya que incluye la potencia perdida tanto por la señal incidente como la reflejada. Para fibras monomodo esta pendiente es de aproximadamente 0.4 dB/m a 1550 nm [6].

La detección OTDR tiene algunos parámetros que caracterizan la medición: resolución espacial ( $W$ ) y frecuencia de repetición de pulsos ( $f_{rep}$ ). La resolución espacial indica la cercanía máxima con la que se pueden diferenciar dos perturbaciones, y depende del ancho de pulso interrogador y de características intrínsecas de la fibra:

$$W = \frac{\tau_p c}{2n_{eff}} \quad (2.1)$$

Donde  $\tau_p$  es el ancho temporal del pulso interrogador,  $c$  es la velocidad de la luz en el vacío y  $n_{eff}$  es el índice de refracción efectivo de la fibra. La frecuencia de repetición de pulsos está limitada por el largo de la fibra, de acuerdo a:

$$f_{rep} \leq \frac{c}{2Ln_{eff}} \quad (2.2)$$

Donde  $L$  corresponde al largo de la fibra. Respetar esta limitación asegura que no se solapa la retrodispersión de un pulso con aquella del siguiente pulso. De esta forma, el ancho de banda de vibraciones medidas será la mitad de  $f_{rep}$ , siguiendo el teorema de Nyquist [12]. A esto se le denomina ancho de banda acústico de un sensor DAS.

### 2.1.3 OTDR sensible a fase ( $\Phi$ -OTDR)

Como se vio en la sección anterior, el uso de luz incoherente para interrogar una fibra nos permite caracterizar su pérdida, además de ubicar fallas y discontinuidades dentro de ella.

La reflectometría óptica en el dominio del tiempo sensible a fase o  $\Phi$ -OTDR cambia la fuente

del pulso incidente a una coherente, cambiando así el comportamiento de la luz retrodispersada por el efecto Rayleigh. Este sensado se basa en que la luz captada en este caso es sensible a la fase relativa de la retrodispersión de Rayleigh en la fibra.

Para ilustrar esto, se considera un ejemplo simplificado con dos centros de dispersión (es decir, dos puntos donde hay cambio de índice de refracción que generan dispersión Rayleigh), posicionados en  $z_A$  y  $z_B$  [6]. Los centros de dispersión  $A$  y  $B$  generan retrodispersión con campo eléctrico  $E_A$  y  $E_B$ , respectivamente. En la figura 2.5 se presenta un diagrama del ejemplo.

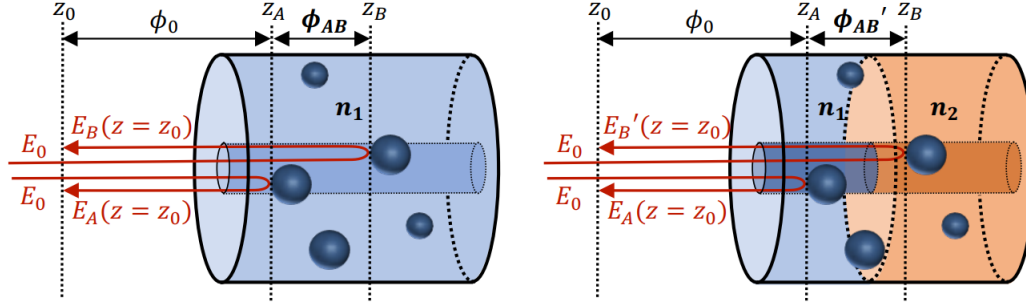


Fig. 2.5: Modelo de fibra con y sin perturbación, a la derecha e izquierda respectivamente [6].

Los campos eléctricos mencionados se pueden expresar como [6]:

$$E_A(z = z_0) = E_0 r \cdot \exp(i\omega t + i\phi_0) \quad (2.3)$$

$$E_B(z = z_0) = E_0 r \cdot \exp(i\omega(t - t_{AB}) + i\phi_0) \quad (2.4)$$

Donde  $r$  es la reflectividad de los centros de dispersión,  $t_{AB} = \frac{2n_1(z_B - z_A)}{c}$  es el tiempo de ida y vuelta de la luz recorrida entre  $z_A$  y  $z_B$ , creando un cambio de fase por propagación. Entonces, la diferencia de tiempo  $t - t_{AB}$  en la ecuación 2.4 se puede expresar como este cambio de fase [6]:

$$E_B(z = z_0) = E_0 r \cdot \exp(i\omega t + i(\phi_0 + \phi_{AB})) \quad (2.5)$$

Donde  $\phi_{AB} = \omega t_{AB}$ . Omitiendo los efectos de pérdidas en la fibra por simplicidad, la potencia óptica resultante de los campos eléctricos retrodispersados está dado por:

$$P(z = z_0) \propto |E_A + E_B|^2 = 2|E_0 r|^2 (1 + \cos(\phi_{AB})) \quad (2.6)$$

En la ecuación 2.6 se puede ver la dependencia de la señal generada en la fase por la retrodispersión de Rayleigh. Esta es la diferencia clave entre  $\Phi$ -OTDR y OTDR clásico, es decir, al cambiar

de una fuente incoherente a una coherente. Las potencias ópticas en cada caso son [6]:

$$P_{incoherente}(z = z_0) \propto 2|E_0 r|^2 \quad (2.7)$$

$$P_{coherente}(z = z_0) \propto 2|E_0 r|^2(1 + \cos(\phi_{AB})) \quad (2.8)$$

Como se indica en la ecuación 2.7, los cambios de fase en la detección incoherente o OTDR clásico no se reflejan en la potencia óptica retrodispersada. Esto se debe a que la frecuencia de una fuente incoherente presenta leves variaciones aleatorias, las cuales, al pasar suficiente tiempo tienden a promediar cero. En el caso de la luz coherente, la frecuencia óptica es altamente estable, haciendo que el valor de la fase relativa entre los dos centros de dispersión  $\phi_{AB}$  sea distinta de cero, afectando directamente la potencia óptica recibida de la retrodispersión.

Cuando hay una perturbación en la fibra, ya sea por cambio de temperatura o deformación, el índice de refracción de la misma cambia en la sección sujeta a la perturbación. Siguiendo el ejemplo, se presenta una perturbación en la sección de fibra con el segundo centro de dispersión, como se muestra en la figura 2.5:

Ahora la segunda sección presenta un índice de refracción  $\eta$  distinto a la primera parte. Esto genera un cambio en el tramo entre  $z_A$  y  $z_B$ , y de fase  $\Delta\phi$ . El nuevo campo eléctrico  $E'_B$  producido por la retrodispersión en  $B$  es:

$$E'_B(z = z_0) = E_0 r \cdot \exp(i\omega t + i(\phi_0 + \phi'_{AB})) \quad (2.9)$$

Donde  $\phi'_{AB}$  representa el cambio de fase  $\phi_{AB} + \Delta\phi$ . La potencia óptica (con fuente coherente) recibida en este nuevo caso es [6]:

$$P(z = z_0) \propto 2|E_0 r|^2(1 + \cos(\phi_{AB} + \Delta\phi)) \quad (2.10)$$

En la ecuación 2.10 se refleja el cambio en la potencia recibida al tener una perturbación en uno de los centros de dispersión, y la ventaja que posee el uso de esquemas  $\Phi$ -OTDR al poder medir a través de la potencia retrodispersada cambios pequeños de temperatura o deformación a lo largo de la fibra.

Al extender este ejemplo a una fibra completa con  $M$  centros de dispersión se obtiene un campo eléctrico generado por las retrodispersiones dado por la ecuación 2.11 [6]:

$$E(t = 2T, z = 0) = E_0 e^{-2\alpha z} e^{i\omega t} \sum_{m=1}^M r_m e^{i\phi_m} \quad (2.11)$$

En este caso el campo se mide en el punto de detección de la luz retrodispersada, es decir, el inicio de la fibra ( $z = 0$ ). El tiempo que demora la luz incidente en llegar a un punto  $z$  en la

## 2. MARCO TEÓRICO

fibra se denota con  $T$ , por lo que el tiempo en que se demora en llegar al punto de dispersión  $z$  y volver al inicio  $z = 0$  para ser detectado corresponde a  $2T$ .  $r_m$  es la reflectividad del centro de dispersión  $m$ , y  $\phi_m$  la fase relativa del mismo. Siguiendo los pasos del ejemplo simplificado, la potencia óptica recibida en la entrada de la fibra está dada por [6]:

$$P_N \propto |E|^2 = \left| \sum_{m=1}^M r_m e^{i\phi_m} \right|^2 = \sum_{m=1}^M r_m^2 + 2 \sum_{i=1}^{M-1} \sum_{j=i+1}^M r_i r_j \cos(\phi_i - \phi_j) \quad (2.12)$$

Esta ecuación muestra como interfiere la luz reflejada para crear la traza  $\Phi$ -OTDR. Una traza típica de  $\Phi$ -OTDR se muestra en la figura 2.6. Es importante notar que la traza tiene la misma forma mientras no se perturbe la fibra, aunque dicha forma sea aleatoria. Esta aleatoriedad se debe a que la distribución de centros de dispersión es aleatoria, aunque las ubicaciones de cada una es fija.

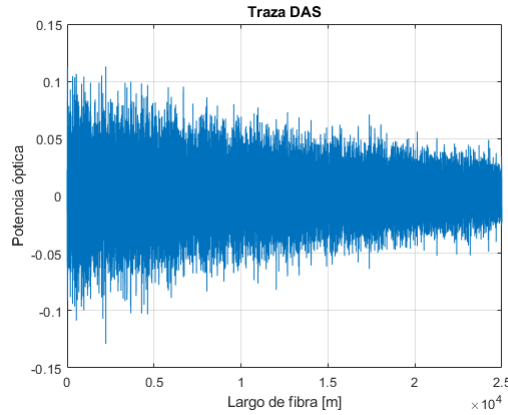


Fig. 2.6: Traza típica obtenida con esquema  $\Phi$ -OTDR.

Al experimentar una perturbación en la fibra, la dispersión en ese punto cambia, y la traza recibida por la detección cambia. Estos cambios son los que permiten detectar las perturbaciones en los distintos puntos de la fibra. El cambio en la fase creado por una perturbación en la fibra se describe en la ecuación 2.13, donde  $\Delta n$  es el cambio del índice de refracción y  $\Delta z$  el el largo de la sección de fibra sujeta a la perturbación.

$$\Delta\phi = 2\frac{2\pi}{\lambda}\Delta n\Delta z \quad (2.13)$$

Este cambio de fase se puede expresar de forma distinta para cambios de fase relativas producidas por cambios de temperatura ( $\Delta T$ ) o deformaciones ( $\Delta\varepsilon$ ). El cambio de fase en el segundo

caso se expresa como [6]:

$$\Delta\phi = \left[ 1 - \frac{1}{2}n_{eff}^2 [(1 - \mu)p_{12} - \mu p_{11}] \right] \cdot \beta\Delta\varepsilon\Delta z \quad (2.14)$$

Donde  $\beta$  es la constante de fase,  $p_{11}$  y  $p_{12}$  constantes opto-elásticas y  $\mu$  la razón de Poisson. En el caso de los cambios de temperatura, el cambio de fase inducido es:

$$\Delta\phi = \left( \frac{1}{\Delta z} \frac{dz}{dT} + \frac{1}{n_{eff}} \frac{dn}{dT} \right) \cdot \beta\Delta\varepsilon\Delta z \quad (2.15)$$

Donde  $\frac{1}{\Delta z} \frac{dz}{dT}$  es el cambio de largo de la fibra debido al cambio de temperatura, y  $\frac{dn}{dT}$  el cambio del índice de refracción generado por el cambio de temperatura. Estas ecuaciones permiten describir los cambios de fase inducidos en la fibra por las perturbaciones de temperatura y deformación.

### 2.1.4 Sensado distribuido de vibración (DVS)

El sensado distribuido de vibración (DVS por sus siglas en inglés) usa las mediciones de potencia óptica captadas por un sistema  $\Phi$ -OTDR para obtener la ubicación de vibraciones que se producen en la fibra. Es un tipo de sensado dinámico, ya que pueden detectar variaciones rápidas de deformación. Si bien este tipo de sensado entrega mediciones no calibradas, se puede determinar con buena seguridad la existencia de la perturbación, y presenta suficiente información para poder clasificarla en distintos tipos de perturbaciones [5]. A diferencia del sensado OTDR clásico, que entrega resultados a partir del promediado de las trazas detectadas, DVS obtiene sus resultados a partir de una serie de trazas. La frecuencia y ubicación de la perturbación se obtiene a partir de las diferencias en la serie de trazas DVS obtenidas [13]. La respuesta a las vibraciones es de carácter no lineal, lo que hace imposible recuperar de forma cuantitativa la perturbación que ocurre en la fibra. Esto presenta una deficiencia clave del sensado DVS: no se recupera la forma de la perturbación, solo su ubicación y frecuencia. La forma clásica de recuperar la información en DVS es obtener la variación de potencia en la fibra restando las trazas obtenidas.

### 2.1.5 Sensado distribuidos acústicos (DAS)

A diferencia del sensado con DVS, el sensado DAS utiliza la fase de la señal de retrodispersión medida para caracterizar perturbaciones en la fibra. Esta fase posee una relación lineal con los cambios de índice de refracción que se producen por perturbaciones en la fibra, como se muestra

en la ecuación 2.13. Esta relación lineal permite recuperar desde la fase la forma de la perturbación aplicada a la fibra. Esto presenta la ventaja que tiene el uso de DAS por sobre DVS: se puede recuperar la forma de la perturbación en el tiempo, además de su ubicación en la fibra [5]. Una gran limitación de usar la señal de retrodispersión de Rayleigh para obtener información de la fibra es que ésta, a pesar de ser la retrodispersión con mayor potencia, sigue siendo baja relativa al pulso incidente. Por lo tanto, para recuperar la fase se deben implementar pasos adicionales en el esquema de detección de forma de mejorar la señal que llega al fotodetector. Como se presentó anteriormente, para recuperar la fase de la señal de RBS se debe usar una fuente interrogadora de luz altamente coherente, y por lo tanto, se deben usar técnicas de detección coherente. En general, esto consiste en separar el haz de luz incidente en dos caminos: uno pasa a ser modulado para generar los pulsos que interrogarán a la fibra, mientras que el otro se dirige a un acoplador óptico donde se combinará con la señal de retrodispersión proveniente de la fibra. A esto se le denomina el oscilador local óptico. Esta señal combinada es la que llega al receptor, de preferencia balanceado [5]. Esta detección se separa en dos tipos, dependiendo de la frecuencia en las que operan el oscilador local y el pulso interrogador: detección homodina y heterodina. La distinción entre ambos métodos viene de la diferencia de frecuencia entre el oscilador local y el pulso interrogador: si ambas señales tienen la misma frecuencia, se denomina detección homodina, y si a la frecuencia óptica del pulso incidente se le aplica un corrimiento de frecuencia  $\Delta\omega$  se le denomina detección heterodina. A continuación se presenta ambos tipos de detección en mayor detalle, incluyendo sus ventajas y desventajas.

### Detección heterodina

Como se mencionó, la detección heterodina se basa en aplicar un corrimiento de frecuencia al pulso interrogador o al oscilador local óptico (OLO). Para lograr esto, un esquema de detección  $\Phi$ -OTDR heterodino utiliza típicamente un modulador acusto-óptico (AOM por sus siglas en inglés) para generar el corrimiento de frecuencia en los pulsos interrogadores. Un esquema típico heterodino se muestra en la figura 2.7.

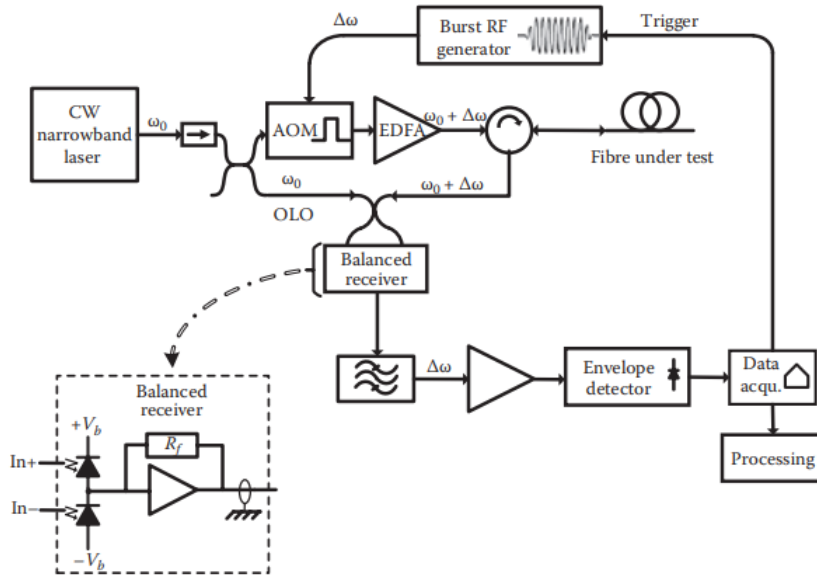


Fig. 2.7: Diagrama de un esquema de detección heterodino típico [5].

Parte de la señal óptica emitida por el láser se utiliza como OLO, para luego ser acoplada con la señal de retrodispersión mediante un acoplador 2x2. El uso de este acoplador permite acoplar la señal débil de la retrodispersión con la señal fuerte del OLO. Esta señal es luego mezclada en un fotodetector balanceado, el cual se utiliza para eliminar el ruido introducido por el OLO. Esta señal se conoce como la señal batida, y es digitalizada y demodulada con demodulación I/Q con el fin de obtener la fase de la señal RBS.

Nótese que se suele usar un amplificador de fibra dopada con erbio (EDFA por sus siglas en inglés) para que el pulso incidente tenga mayor potencia al entrar a la fibra.

Si consideramos  $\omega$  la frecuencia óptica de la fuente coherente,  $\Delta\omega$  el corrimiento de frecuencia generado en el esquema por el AOM y las fases iniciales de las señales de retrodispersión y del oscilador local  $\phi_{RBS}$  y  $\phi_{OL}$ , respectivamente, además de  $\Delta\phi$  como el cambio de fase generado en la fibra por perturbaciones, podemos representar los campos eléctricos de las señales de

retrodispersión y del oscilador local [14]:

$$E_{RBS}(t) = |E_{RBS}(t)| e^{j(\omega t + \Delta\omega t + \Delta\phi(t) + \phi_{RBS})} \quad (2.16)$$

$$E_{OL}(t) = |E_{OL}(t)| e^{j(\omega t + \phi_{OL})} \quad (2.17)$$

Aquí los operadores  $|\cdot|$  indican el cálculo de la amplitud de la señal. Antes de la detección en el fotodetector, estas dos señales se mezclan en el acoplador 2x2 disminuyéndolas a la mitad de la potencia óptica de entrada, debido al factor de acoplamiento del acoplador óptico. Además, una señal se desfasa  $\pi/2$ , quedando las señales de campo eléctrico a la salida del acoplador:

$$E_{OC_1}(t) = \frac{1}{2} [E_{RBS}(t) + E_{OL}(t)e^{j\pi/2}] \quad (2.18)$$

$$E_{OC_2}(t) = \frac{1}{2} [E_{RBS}(t)e^{j\pi/2} + E_{OL}(t)] \quad (2.19)$$

Las señales de salida del acoplador pasan al fotodetector balanceado, donde son independientemente transformadas a señales eléctricas. Además se produce el batimiento de las señales de retrodispersión y del OLO, obteniendo las señales de corriente que se muestran en las ecuaciones 2.20 y 2.21, con  $r$  siendo la responsividad de los fotodiodos y  $*$  indicando complejo conjugado.

$$i_1(t) = rE_{OC_1}(t)E_{OC_1}(t)^* = \frac{r}{4} [(E_{RBS}(t) + E_{OL}(t)e^{j\pi/2})(E_{RBS}(t) + E_{OL}(t)e^{j\pi/2})^*] \quad (2.20)$$

$$i_2(t) = rE_{OC_2}(t)E_{OC_2}(t)^* = \frac{r}{4} [(E_{RBS}(t)e^{j\pi/2} + E_{OL}(t))(E_{RBS}(t)e^{j\pi/2} + E_{OL}(t))^*] \quad (2.21)$$

Reemplazando en estas ecuaciones las expresiones de  $E_{RBS}$  y  $E_{OL}$  (2.18 y 2.19, respectivamente) y aplicando la fórmula de Euler ( $\cos(x) = \frac{e^{jx} + e^{-jx}}{2}$ ), se obtienen las siguientes expresiones para las corrientes del fotodetector:

$$i_1(t) = \frac{r}{4} [|E_{RBS}(t)|^2 + |E_{OL}(t)|^2 + 2|E_{RBS}(t)||E_{OL}(t)|\cos(\Delta\omega t + \Delta\phi(t) + \phi_{RBS} - \phi_{OL} - \pi/2)] \quad (2.22)$$

$$i_2(t) = \frac{r}{4} [|E_{RBS}(t)|^2 + |E_{OL}(t)|^2 - 2|E_{RBS}(t)||E_{OL}(t)|\cos(\Delta\omega t + \Delta\phi(t) + \phi_{RBS} - \phi_{OL} + \pi/2)] \quad (2.23)$$

Como se mencionó anteriormente, las señales del fotodetector se restan para eliminar el componente DC de la señal, generando la señal batida:

$$i_{balanceada} = r|E_{RBS}(t)||E_{OL}(t)|\cos(\Delta\omega t + \Delta\phi(t) + \phi_{RBS} - \phi_{OL} - \pi/2) \quad (2.24)$$

Nótese que la señal balanceada elimina el componente DC siempre y cuando la potencia óptica

## 2. MARCO TEÓRICO

entre ambos canales de entrada al fotodetector sean iguales. El espectro de esta señal se centra en la frecuencia intermedia  $\Delta\omega$ , la cual generalmente es del orden de algunos cientos de megahertz [15] y con ancho de banda inversamente proporcional a  $1/\tau_p$ , con  $\tau_p$  siendo el ancho del pulso incidente. Para extraer la fase de la señal en frecuencia intermedia, primero se obtienen las componentes *in-phase* (I) y *quadrature-phase* (Q) aplicando multiplicaciones de seno y coseno a la señal eléctrica digitalizada. De aquí, la fase de la señal se calcula con el arcotangente de Q/I. La fase encontrada contiene la información de las perturbaciones a las que fue sometida la fibra, y permite reconstruirlas en el tiempo.

Respecto a la obtención de información relevante a partir de esta señal de fase se hacen dos acotaciones. Primero, que para obtener la ubicación y caracterización de la perturbación en la fibra se debe obtener una serie de muestras temporales de la fibra y restar las señales de fase resultantes en segmentos espaciales regulares, conocidos como *gauge length*, el cual determina la resolución espacial del sensor. Esto se debe a que las perturbaciones generan un cambio de fase que se propaga a través de toda la fibra, acumulándose con todos los otros cambios de fase que se van generando en la fibra [5]. Obteniendo la fase diferencial entre puntos se localiza y caracteriza la perturbación medida. Este proceso se muestra de forma gráfica en la figura 2.8, donde al restar las fases se obtiene la perturbación en un punto de la fibra presente en b).

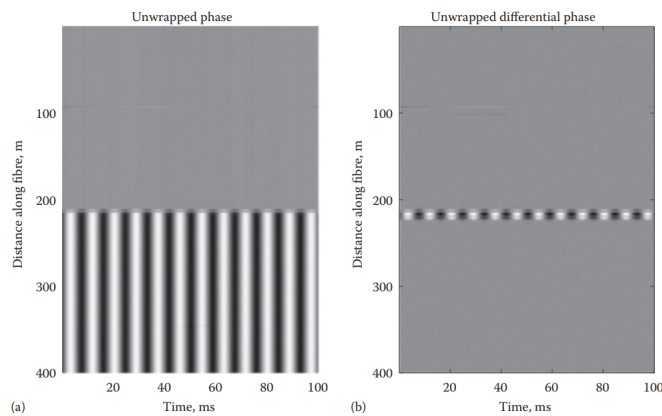


Fig. 2.8: a) Fase acumulada a lo largo de una fibra a través del tiempo, y b) Fase diferencial a lo largo de una fibra en el tiempo [5]

El segundo punto, y la causa y enfoque de este trabajo es que como la señal de fase se recupera mediante el uso de la función arcotangente, está restringida o *envuelta* entre  $-\pi$  y  $\pi$ . Esto significa que se debe aplicar algún método para poder desenvolver esta fase y así reconstruir la fase original para poder ubicar y caracterizar correctamente las perturbaciones en la fibra. Como se verá más adelante, hay una gran cantidad y variedad de métodos de desenvolvimiento de fase, con diferentes niveles de complejidad, y distintas fortalezas y debilidades.

### DetECCIÓN HOMODINA

En un esquema de detección homodina, la señal de la retrodispersión se mezcla con una señal de la misma frecuencia en el oscilador local. Esto significa que la señal resultante está en bandabase, reduciendo el ancho de banda necesario en el fotodetector. Para extraer la fase de la señal batida en esta configuración, es necesario usar una óptica híbrida, la cual toma las señales de entrada y las desfasa en 90 grados a la salida. Esto se muestra en la figura 2.9. De esta forma se obtienen las componentes I y Q necesarias para obtener la fase de la señal, al igual que en la detección heterodina [16]. Al igual que la detección heterodina también es necesario aplicar algún método de desenvolvimiento de fase a la señal obtenida, ya que también está limitada entre  $-\pi$  y  $\pi$ .

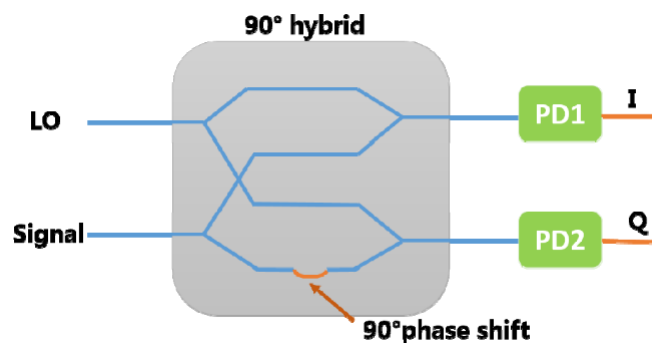


Fig. 2.9: Desfase de 90 logrado con híbrida óptica [16].

### 2.1.6 Métricas relevantes en el funcionamiento de un sensor DAS

Para caracterizar el rendimiento de un sensor DAS se consideran cuatro parámetros claves: resolución espacial, rango o distancia máxima, resolución de medición y ancho de banda acústico [17]. La resolución espacial describe distancia mínima entre puntos que se pueden medir de manera distintiva. Una mayor resolución espacial permite localizar de manera más precisa la ubicación de las perturbaciones que afectan a la fibra. Este parámetro es afectado principalmente por el ancho del pulso interrogador y el ancho de banda del esquema de detección. El rango indica el punto de máxima distancia que se puede medir en la fibra, y depende de un número de factores como la potencia óptica del pulso de entrada, ruido y pérdidas en la fibra. El ancho de banda acústico es determinado por la frecuencia de repetición de los pulsos interrogadores en la fibra, cuyo límite superior está definido en la ecuación 2.2.

También se define la razón señal a ruido (SNR por sus siglas en inglés) como la proporción entre señal útil y ruido percibido. Esta métrica depende tanto de la resolución espacial como el rango, y un mayor SNR significa una señal más clara y distinguible del ruido, lo que permite mejores mediciones.

## 2. MARCO TEÓRICO

---

Para lograr mediciones correctas, se deben considerar todos los factores mencionados y la interacción entre ellos, pues frecuentemente mejorar una de estas métricas implica un *tradeoff* en otra. Además se deben considerar aspectos propios de las fibras ópticas, como las no linealidades que se producen a partir de ciertos niveles de potencia. Si se quisiera obtener una traza de mayor potencia y/o tener una mayor distancia de sensado se debe incrementar la potencia óptica de entrada o aumentar el ancho del pulso interrogante. La potencia óptica tiene un límite superior, que una vez superado induce efectos no lineales en la fibra [18]. Al aumentar el ancho del pulso de entrada, se pierde resolución espacial. De esta misma forma al mejorar parámetros por una parte genera empeoramientos en otros.

### **Rango o distancia de sensado**

Al utilizar la luz del RBS, la señal que capta el esquema  $\Phi$ -OTDR es de muy baja potencia. Además, a medida que aumenta la distancia total de sensado, la potencia de la señal disminuye de manera exponencial producto de pérdidas de la fibra. Existen distintas formas de aumentar la distancia de sensado, como aumentar la potencia de entrada via un EDFA. Esto trae consigo las dificultades mencionadas anteriormente, arriesgando producir efectos no lineales en la fibra, induciendo distorsiones en las mediciones.

Se han desarrollado diferentes técnicas para combatir esto y aumentar el rango del sensado DAS, como el uso de amplificación Raman bidireccional [19]. Este esquema ha logrado un rango de 62 km de sensado, con una resolución espacial de 100 m. Posteriormente esquemas utilizando versiones de mayor orden lograron sensar alrededor de 130 km, usando la misma amplificación pero de segundo orden [20], y combinando amplificación de primer orden junto con detección heterodina [19]. Otros esquemas propuestos utilizando amplificación Brillouin en conjunto con detección heterodina han logrado una distancia de sensado de 124 km [21]. En [22] se combinaron las amplificaciones de Raman de primer y segundo orden junto con la de Brillouin logrando un rango de 175 km.

Otro esquema ha logrado un rango de 150 km utilizando repetidores ópticos consistentes de dos EDFA y un amplificador Raman [23]. A pesar de los buenos resultados, lo complejo y costoso de este esquema lo hace poco práctico para usos reales. En [24], mediante el uso de moduladores opto-acústicos (AOMs) se logró una distancia de sensado de 94,8 km en un esquema de detección directa.

### **Razón señal a ruido (SNR)**

La razón señal a ruido está definida por casi todas las otras métricas importantes en el sensado  $\Phi$ -OTDR: rango, resolución espacial, ancho de banda acústico y resolución de medición. Es por

## 2. MARCO TEÓRICO

---

esto que es un factor altamente importante a la hora de determinar el rendimiento del sensado. Por definición, la SNR se mejora aumentando la potencia de la señal medida, y disminuyendo la intensidad del ruido. El primero se logra aumentando la potencia del pulso incidente, mientras que el segundo puede lograrse con una variedad de métodos que incluyen los procesos de fotodetección, amplificación óptica, generación de pulsos, entre otros [6]. El proceso de fotodetección, el cual transforma la señal óptica en una corriente eléctrica de manera proporcional:

$$I = \eta P \quad (2.25)$$

Donde  $I$  es la corriente generada por el fotodetector,  $\eta$  la responsividad del mismo, y  $P$  la potencia óptica recibida. Los ruidos que más afectan a este proceso son los de disparo y térmico, los cuales afectan la señal eléctrica de salida de manera aditiva. En el sistema heterodino, el ruido de disparo está descrito por:

$$\sigma_{sh}^2 = 2qI_p B = 2q\eta(P_{OL} + P_{RBS})B \approx 2q\eta P_{OL}B \quad (2.26)$$

Con  $q$  la carga de un electrón,  $P_{OL}$  y  $P_{RBS}$  las potencias del oscilador local y de la retrodispersión, respectivamente.  $B$  es el ancho de banda del fotodetector, y  $I_p$  la corriente generada [25], considerando:

$$I_p \approx \eta(P_{OL} + P_{RBS}) \quad (2.27)$$

Esta aproximación es válida para  $P_{LO} \gg P_{RBS}$ , que es normal en sistemas coherentes.

Para el ruido térmico se considera generalmente una constante de valor despreciable dada la diferencia de potencia entre el oscilador local y RBS recién descrito. Está descrita por la ecuación 2.28:

$$\sigma_{th} = \frac{4k_b T}{R_L} B \quad (2.28)$$

Con  $k_b$  la constante de Boltzmann,  $T$  la temperatura absoluta y  $R_L$  la resistencia de carga asociada al fotodetector [26].

Considerando los ruidos de disparo y térmico ( $\sigma_{sh}$  y  $\sigma_{th}$ , respectivamente) se puede obtener una expresión para la SNR. Primero se considera la potencia eléctrica promedio:

$$\langle I_p^2 \rangle = \eta^2 [P_{OL}^2 + P_{RBS}^2] \approx \eta^2 P_{OL}^2 \quad (2.29)$$

Y se divide por la potencia del ruido de fotodetección para obtener la expresión de SNR [26]:

$$SNR = \frac{\eta^2 P_{OL}^2}{\sigma_n^2} = \frac{2\eta^2 P_{OL}^2}{2q\eta P_{OL}B + \sigma_{th}^2} \approx \frac{\eta P_{OL}}{qB} \quad (2.30)$$

Asumiendo aún que la potencia del ruido de disparo es mucho mayor que la del ruido térmico. Otra fuente considerable de ruido proviene de la amplificación de la señal óptica mediante el uso de amplificadores como los EDFA. Los EDFA generan emisión espontánea además de la estimulada (la cual amplifica la señal) [26]. La emisión espontánea corresponde a una señal óptica que no es parte de la señal original que se busca amplificar, creando ruido denominado emisión espontánea amplificada (ASE por sus siglas en inglés). La introducción de este ruido empeora la SNR, y es acumulable si se usan varios amplificadores en serie. Además, este ruido tiene un amplio espectro, haciéndose más amplio a mayores rangos de ganancia [26]. Para mitigar este ruido, se suelen usar filtros ópticos, y seleccionar ganancias que minimicen el ASE.

El proceso de generar los pulsos interrogadores también es una fuente de ruido. Al modular la señal óptica con un AOM una pequeña parte de la luz fuera del pulso también pasa a la fibra, contaminando la señal [27]. Esto se cuantifica con el *extinction ratio*, de aproximadamente 50 dB [28]. En este trabajo se usa un amplificador óptico semiconductor (SOA por sus siglas en inglés) para generar los pulsos, el cual posee un *extinction ratio* similar a los AOM, alrededor de 50 dB [29, 30].

El láser que genera los pulsos también puede ser una fuente de ruido en el sistema debido a inestabilidad del mismo [26]. Variaciones en la temperatura o cambios mecánicos en el láser producen variaciones aleatorias en la fase y/o intensidad del láser, produciendo a su vez inestabilidades en la intensidad y fase de la señal de RBS. También se pueden producir pequeñas inestabilidades en la frecuencia del láser, disminuyendo su coherencia al aumentar su ancho espectral, afectando la precisión de las mediciones [31]. Esto resalta la relevancia de usar fuentes con un *line width* angosto.

Por su definición, la SNR se puede mejorar de dos formas: aumentando la potencia de la señal de interés o reduciendo el nivel de ruido presente. Existen diferentes formas de lograr esto desde el esquema de detección, generalmente aumentando la complejidad y/o costo del sistema. Por este aumento en complejidad y costo es que resulta más oportuno utilizar métodos de mejoramiento de señal en el posprocesamiento de las señales adquiridas. Existen muchas formas de mejorar la señal con procesamiento de señales, logrando muy buenos resultados. En [32] se utiliza un algoritmo de filtrado 2D bilateral, logrando mejorar la SNR en más de 14 dB a partir de 6,43 dB en una fibra de 27,6 km. Esto se logró filtrando una imagen bidimensional, creada a partir de los datos captados en un esquema  $\Phi$ -OTDR. Usando esta misma transformación a imagen 2D, en [33] se propone *curvelet denoising*, logrando un aumento de hasta 8 dB en el SNR, comparado

con métodos tradicionales. Con *empirical mode decomposition* (EMD) se ha logrado llegar a un SNR cercano a 40 dB en una fibra de 2 km [34].

### Intensity fadings

Como se mencionó anteriormente, la señal de RBS producida en la fibra tiene una forma aleatoria debido a la distribución aleatoria de los centros de dispersión en la fibra. Además de esta aleatoriedad en la forma, cada punto en la traza de RBS tiene una intensidad aleatoria que sigue una distribución exponencial [11], como se muestra en la figura 2.10. Esta distribución significa que en cualquier punto de la fibra hay una probabilidad considerable de tener una intensidad baja. A las posiciones en las cuales la intensidad o amplitud óptica está por debajo de un valor umbral, se les llama *fading points*, o puntos de desvanecimiento. La existencia de estos puntos induce un empeoramiento en la SNR en aquellos puntos de la fibra [35], generando complicaciones en los distintos procesos de posprocesamiento, incluyendo el cálculo de la fase y su desenvolvimiento.

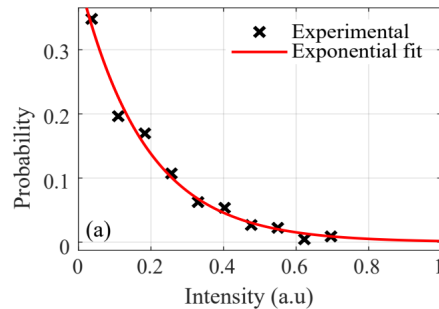


Fig. 2.10: Distribución de probabilidad de las intensidades en una traza  $\Phi$ -OTDR [11].

Existen varias técnicas para disminuir el efecto de los puntos de desvanecimiento en esquemas  $\Phi$ -OTDR. Por ejemplo, el uso de un pulso doble desplazado en fase ha logrado una mejora de SNR de 20 dB en una fibra de 4 km, a cambio de una reducción del ancho de banda acústico a la mitad [36]. En [37] se utiliza un pulso modulado en múltiples frecuencias, consiguiendo una mejora de 45 dB en la SNR. Es importante notar que estos métodos logran estas mejoras mediante cambios en el esquema físico de  $\Phi$ -OTDR, lo que, al igual como se mencionó anteriormente, aumenta complejidad y costos del sistema.

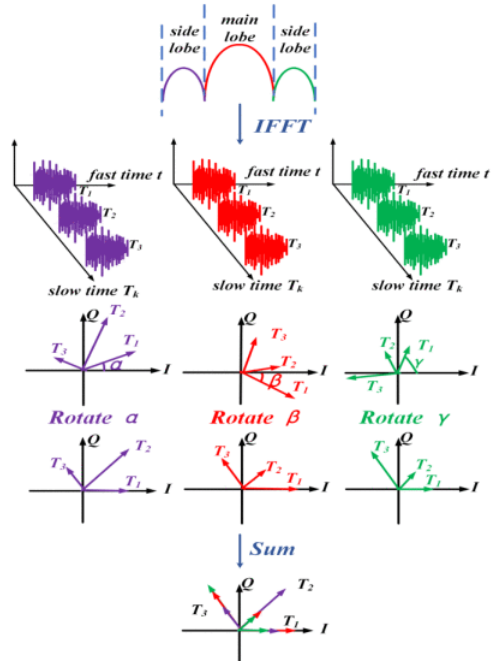


Fig. 2.11: Funcionamiento de SERVS [38].

Otra método es *spectrum-extraction and rotated-vector-sum* (SERVS) [38], ilustrado en la figura 2.11, el cual mejora la SNR en puntos de desvanecimiento mediante posprocesamiento. Primero, se extraen distintas partes del espectro de la traza  $\Phi$ -OTDR, normalmente el lóbulo principal y los lóbulos laterales, ya que presentan baja correlación y fluctuaciones independientes cuando no se solapan en frecuencia [39]. Luego, las señales extraídas se transforman al dominio temporal y se combinan mediante *rotated-vector sum* [40], corrigiendo las diferencias de fase entre mediciones contiguas que impedirían una suma directa. La rotación de la primera traza a fase cero, aplicada de forma consistente al resto, preserva las fases relativas entre mediciones y lóbulos, permitiendo una suma constructiva que reduce el efecto de los desvanecimientos y mejora la SNR local en la fibra.

### Resolución de medición

La resolución de medición corresponde al menor valor que es posible medir con un sensor. Para el esquema  $\Phi$ -OTDR corresponde a la menor deformación o cambio de temperatura medible mediante la fase de la señal de RBS en la fibra. Siguiendo la línea de [25], se define la desviación estándar de la fase diferencial para cuantificar la resolución, la cual está afectada por los ruidos de intensidad y de fase en el sistema. El ruido de intensidad fue descrito anteriormente, compuesto de ruido de disparo y térmico. Dado que estos dos ruidos se consideran constantes, el ruido de intensidad también se considera constante. El ruido de fase se acumula a lo largo de la fibra, debido a ruido de fase en la luz incidente, inestabilidad térmica, ruido introducido por los

distintos componentes del esquema, conectores, entre otros. Al acumularse, a mayor distancia en la fibra se presenta un mayor ruido de fase como puede visualizarse en la figura 2.12.

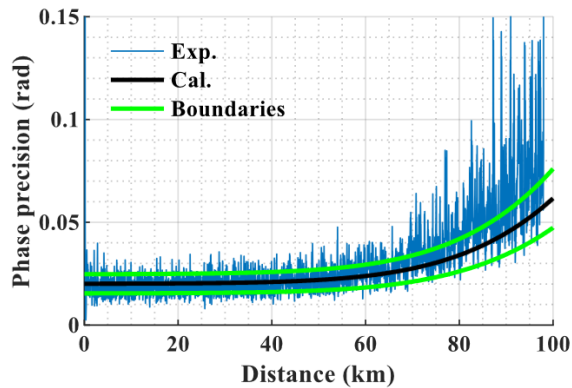


Fig. 2.12: Precisión de fase experimental y calculada a lo largo de una fibra interrogada con un pulso de 800 ns [41].

## 2.2 Desenvolvimiento de fase

Como se ha descrito en la sección anterior, la información de la perturbación aplicada a la fibra en un sistema  $\phi$ -OTDR se encuentra en la fase de la RBS. Sin embargo, al recuperar esta fase desde la demodulación I/Q usando la función de arcotangente se obtiene una señal envuelta en el rango  $[-\pi, \pi]$ , siendo similar a una señal de diente de sierra, lo cual no representa la fase real de la señal. A esta fase con saltos le llamamos fase envuelta, fase enrollada, o *wrapped phase*. Una señal con fase envuelta y desenvuelta se muestra en la figura 2.13.

Se debe recuperar la fase original o desenvuelta para poder trabajar sobre las mediciones acústicas obtenidas en un esquema  $\Phi$ -OTDR. El proceso de desenvolvimiento de fase típicamente se aplica como etapa final al posprocesamiento, ya que los métodos de desenvolvimiento de fase tienden a ser muy sensibles a ruido, por lo que se busca entregarles una señal lo más limpia posible.

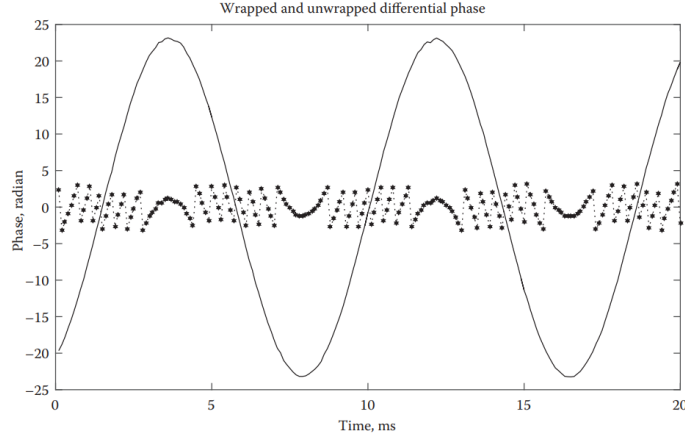


Fig. 2.13: Fase diferencial envuelta y desenvuelta obtenida en una medición con esquema heterodino [5].

El proceso de desenvolvimiento de fase puede formalizarse siguiendo el análisis de Itoh [42]. Para ello, se definen dos operadores básicos: el operador de envolvimiento  $\Phi$  y el operador de diferencia finita  $\Delta$ . El operador de envolvimiento restringe una señal al rango  $[-\pi, \pi]$ :

$$\Phi_l s(n) = s(n) + 2\pi k_l(n) \quad (n = 0, 1, 2, \dots, N) \quad (2.31)$$

Donde  $s(n)$  es una secuencia y  $k_l(n)$  una secuencia de enteros tal que

$$-\pi \leq \Phi_l s(n) < \pi \quad (n = 0, 1, 2, \dots, N) \quad (2.32)$$

Es decir,  $\Phi_l s(n)$  representa la señal envuelta. El operador de diferencia finita  $\Delta$  representa la resta entre muestras consecutivas:

$$\Delta s(n) = s(n) - s(n-1) \quad (n = 0, 1, 2, \dots, N) \quad (2.33)$$

Con estos operadores, el desenvolvimiento de fase puede expresarse como:

$$\sum_{n=1}^m \Phi_2 \Delta \Phi_1 s(n) = \sum_{n=1}^m [\Delta s(n) + 2\pi(\Delta k_1(n) + k_2(n))] \quad (2.34)$$

Si se cumple la condición de Itoh

$$-\pi \leq \Delta s(n) < \pi \quad (2.35)$$

Entonces los términos enteros se anulan:

$$\Delta k_1(n) + k_2(n) = 0 \quad (2.36)$$

De este modo, el desenvolvimiento puede simplificarse a:

$$s(m) = s(0) + \sum_{n=1}^m \Phi_2 \Delta \Phi_l s(n) \quad (2.37)$$

Es decir, se envuelven las diferencias de la fase envuelta original, y luego se suman dichas diferencias para recuperar la señal desenvuelta. Este procedimiento es válido siempre que 2.35 se cumpla. Intuitivamente, esta condición garantiza que no haya aliasing en la fase: si el cambio real entre dos muestras es mayor a  $2\pi$ , el operador de envolvimiento no puede distinguir entre el salto correcto y el múltiplo entero de  $2\pi$ , lo que introduce errores al proceso.

Esta labor se complica cuando hay saltos de fase que no son producto de la obtención de fase en sí, si no que vienen de la medición en forma de ruido o discontinuidades. Los métodos más simples no tienen una forma de discernir entre saltos de fase propias de la generación de la señal aplicando arcotangente y los saltos generados por ruido, aplicando corrección de fase en ambos casos [43]. Dado que la señal de fase es continua, cualquier sobre- o sub- corrección en la fase afecta acumulativamente a la parte de la señal que le sigue temporalmente, propagando errores que se propagan de forma creciente [44].

En la actualidad, existen una variedad de métodos de desenvolvimiento de fase aplicados en un gran rango de aplicaciones en distintas áreas, desde medicina para procesamiento de imágenes de resonancias magnéticas [1] y ultrasonido [45], hasta modelamiento geofísico via interferometría en radares de apertura sintética (InSAR)[2].

Al trabajar con datos DAS obtenidos de un esquema  $\Phi$ -OTDR resulta clave la correcta obtención de la fase desenvuelta generada de la detección del RBS para poder detectar y caracterizar de manera fiel las perturbaciones que afectan a la fibra bajo prueba.

### 2.3 Métodos de desenvolvimiento de fase

En esta sección se describe de forma teórica el funcionamiento de los métodos de desenvolvimiento de fase utilizados en esta memoria.

### 2.3.1 Método *Unwrap* de Matlab

Es el método más simple consistente en detectar saltos de fase mayores a  $\pi$  en vectores unidimensionales consecutivos, y sumar múltiplos de  $\pm 2\pi$  hasta que el salto sea menor a  $\pi$  [3]. El tamaño del salto que induce una corrección de fase es configurable por el usuario a cualquier valor mayor a  $\pi$ .

Esta función acepta de entrada la secuencia de datos de fase a desenvolver, la cual puede ser un arreglo unidimensional o bidimensional (una matriz). En el segundo caso se muestra en la figura 2.14, el desenvolvimiento se realiza iterativamente en cada fila o columna de la matriz.

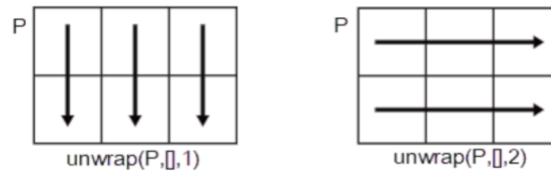


Fig. 2.14: Desenvolvimiento de fase para matrices con función incluida en MATLAB [3].

Por su funcionamiento, es el método con menor robustez al ruido de fase, ya que cualquier salto de fase mayor a la tolerancia se corrige, sin ningún tipo de revisión de si corresponde a un salto de fase creado en el proceso de demodulación o si es un salto producido por ruido en la señal original.

Esta función viene incluida en MATLAB, y su documentación se encuentra en [3]. Es frecuentemente utilizada en la literatura como una base para la comparación al implementar nuevos métodos [46, 47].

### 2.3.2 Método basado en Ecuación de Transporte de Intensidad (TIE)

Este método aborda el problema de desenvolvimiento de fase en dos dimensiones, descrito a continuación:

$$\phi(x, y) = \varphi(x, y) + 2\pi \cdot k(x, y) \quad (2.38)$$

Donde  $\phi(x, y)$  corresponde a la fase envuelta,  $\varphi(x, y)$  a la fase desenvuelta y  $k$  el entero que se debe determinar para resolver el desenvolvimiento.

Este método está diseñado para desenvolver mapas de fase, es decir, matrices bidimensionales de valores de fase y usa un enfoque de mínimos cuadrados para resolver la ecuación de Poisson:

$$\frac{\partial^2 \phi(x, y)}{\partial x^2} + \frac{\partial^2 \phi(x, y)}{\partial y^2} = \nabla^2 \varphi(x, y) \quad (2.39)$$

En este contexto, el método de mínimos cuadrados busca encontrar una función cuya aproximación a las derivadas parciales de la fase envuelta minimice el error en sentido cuadrático. La formulación exacta varía según la implementación [48, 49].

En la propuesta de Zhao [50], el método de mínimos cuadrados se utiliza en conjunto con la ecuación de transporte de intensidad (TIE, por sus siglas en inglés) y la transformada cosenoidal discreta (DCT, por sus siglas en inglés) para la resolución. Para ello, en la ecuación 2.39 el término del lado derecho se reemplaza por la expresión TIE [51]:

$$\frac{\partial^2 \phi(x, y)}{\partial x^2} + \frac{\partial^2 \phi(x, y)}{\partial y^2} = \frac{2\pi}{\lambda} \frac{I(x, y, z + \Delta z) - \lambda I(x, y, z - \Delta z)}{2\Delta z} \quad (2.40)$$

Donde  $I(x, y, z + \Delta z)$  y  $I(x, y, z - \Delta z)$  representan las intensidades correspondientes a dos planos levemente desenfocados (*defocus*) con respecto al plano de referencia  $z$ . Aquí el eje  $z$  no es una tercera dimensión física, sino que corresponde a la dirección de propagación de la onda perpendicular al plano  $(x, y)$ . Este tercer eje puede ser adquirido de manera experimental, pero en el caso de la implementación de Zhao es generado de forma sintética propagando numéricamente la onda desde la fase envuelta.

Sobre esta formulación se propuso en [52] una solución analítica:

$$\frac{\partial^2 \phi(x, y)}{\partial x^2} + \frac{\partial^2 \phi(x, y)}{\partial y^2} = \text{Im}[e^{-i\phi(x, y)} \nabla^2 e^{i\phi(x, y)}] \quad (2.41)$$

La ventaja de esta forma es que se elimina el cálculo de propagación de ondas, reduciendo el costo computacional. El método propuesto por Zhao se aplica sobre esta solución analítica, calculando el lado derecho de la ecuación mediante diferencias finitas y resolviendo con DCT. Además agrega una parte iterativa de corrección de fase, para añadir robustez en casos con señales ruidosas.

La implementación desarrollada por el autor original se encuentra libremente accesible vía un paquete de MATLAB [53].

### 2.3.3 Filtro de Kalman

El filtro de Kalman [54] es un estimador de estado que permite predecir el estado futuro de un sistema (usualmente series temporales) usando mediciones de estados pasados y actuales.

De forma general, la implementación de un filtro de Kalman (en su formulación original de tiempo discreto) consiste en una ecuación de diferencia que describe el proceso y una que describe las mediciones de dicho proceso [55]:

$$x_k = Ax_{k-1} + Bu_k + w_{k-1} \quad (2.42)$$

$$z_k = Hx_k + v_k \quad (2.43)$$

Donde  $x_k$  es el estado a estimar en instante  $k$ , y  $z_k$  la medición de dicho estado. Las variables  $v_k$  y  $w_k$  representan el ruido del proceso y de medición, respectivamente, y corresponden a variables aleatorias independientes entre sí, y que siguen una distribución normal.  $A$  y  $B$  son matrices, donde la matriz  $A$  relaciona el estado actual  $k$  con el anterior  $k - 1$  en un ambiente ideal, y la matriz  $B$  relaciona la entrada  $u_k$  con el estado  $x_k$ . Finalmente, la matriz  $H$  relaciona el estado con la medición.

Su funcionamiento puede dividirse en dos pasos que se realizan de forma alternada: el paso de predicción y el de actualización. En el paso de predicción, se utilizan los estados de intervalos anteriores para estimar el estado actual. En la etapa de actualización, la estimación recién mencionada se mezcla con mediciones actuales para estimar los estados actuales y futuros del sistema [56]. Actualmente el filtro de Kalman es fundamental en distintas disciplinas e industrias relacionadas al control, automatización y diferentes tipos de procesamiento de señales [56]. Requieren de bajos recursos computacionales y de memoria, por lo que son muy utilizados en sistemas embebidos y aplicaciones en tiempo real.

La versión básica del filtro de Kalman es adecuada para sistema lineales, es decir, la ecuación de observación debe ser lineal. Los sistemas reales rara vez suelen ser lineales, por lo que se han desarrollado versiones alternativas del filtro de Kalman que funcionan en sistemas no lineales, como el filtro de Kalman extendido (EKF) y el filtro de Kalman *unscented* (UKF). A grandes rasgos, el funcionamiento del EKF se basa en enfocarse en la expansión de Taylor de primer orden del estado estimado y así transformar el sistema no lineal en uno lineal [57]. En otras palabras, el estado es estimado usando variables aleatorias Gaussianas (GRV) que se propagan analíticamente a través de la linealización de primer orden del sistema [58]. Si bien esto permite trabajar sobre sistemas no lineales, es una aproximación no óptima, y puede llevar a un mal desempeño e incluso la divergencia del filtro [59].

El UKF [60] alivia estos problemas al usar una selección acotada de puntos muestreados de la GRV para representar la distribución de estados. La idea es que estos puntos capturen la verdadera media y covarianza de la GRV, que al propagarse a través del sistema no lineal capturan el sistema aproximado a la serie de Taylor de tercer orden para cualquier no-linealidad [59]. Esto viene de la idea de que es más fácil aproximar una distribución de probabilidad que aproximar una función o transformación no lineal arbitraria [61]. De esta forma se linealiza la función aleatoria no lineal a través de una regresión lineal entre puntos obtenidos de la distribución de

la variable aleatoria no lineal [62]. A estos puntos seleccionados para representar la GRV se les denomina puntos Sigma.

De manera simplificada, el algoritmo del UKF sigue los siguientes pasos: primero se calculan y asignan pesos a los puntos Sigma, los cuales se transforman usando la función no lineal, para luego calcular la distribución Gaussiana de dichos puntos y los estados asociados. Finalmente se calculan la media y la varianza de la distribución obtenida. Este proceso se realiza antes de los procesos de estimación y actualización del filtro de Kalman común.

El UKF es entonces más robusto que el EKF, y tiene la misma complejidad computacional, haciéndolo una mejor alternativa al abordar sistemas no lineales, y de alto interés en este trabajo.

### 2.3.4 Métodos basados en Machine Learning

Los siguientes métodos implementados obtienen la fase desenvuelta usando machine learning, en vez de la forma analítica de los métodos ya mencionados.

De manera general, los métodos de machine learning nos permiten encontrar patrones en datos de forma automática, y usar esos patrones para predecir datos futuros [63]. En [64], Goodfellow lo define como una forma de estadística aplicada con un énfasis en el uso de computadores para estimar funciones complicadas. Este aspecto es el que resulta relevante al caso del desenvolvimiento de fase. Recordando la ecuación 2.38, se está buscando aproximar de la mejor forma la función  $k(x, y)$  que permite llevar la fase envuelta a la desenvuelta. El uso de machine learning nos permite desarrollar modelos que permiten llevar a cabo esta aproximación para desenvolver fases con alta robustez, ya que los modelos aproximan con alto grado de precisión las condiciones reales de las señales con las que se trabajan.

#### Consideraciones generales para métodos de machine learning

De manera general, todos estos métodos funcionan de la misma forma: se construye un modelo compuesto de capas de procesamiento, cambiando la forma de los datos de manera de extraer los patrones o *features* que permiten predecir de mejor manera la fase. Luego los modelos se entrenan a partir de una serie de pares de datos: señales envueltas y sus contrapartes desenvueltas. El modelo encuentra los patrones y formas que se repiten entre estas contrapartes para “aprender”. En el proceso de entrenamiento del modelo también se hacen validaciones con datos nuevos que no ha visto el modelo para verificar el proceso de aprendizaje. Una vez completado el proceso, se usa el modelo para obtener fases desenvueltas. El modelo recibe como entrada un arreglo de fases y tiene por salida otro arreglo con la fase desenvuelta.

Una consideración clave al implementar cualquier modelo de machine learning es el costo computacional asociado a su entrenamiento. Dependiendo de la complejidad del modelo y de

## 2. MARCO TEÓRICO

la capacidad de cómputo disponible, esta etapa puede requerir desde varias horas hasta varios días. Sin embargo, el entrenamiento se realiza solo una vez. En el caso específico del desenvolvimiento de fase, una vez entrenado y validado un modelo, el tiempo de procesamiento de datos es comparable al de los métodos clásicos, e incluso puede ser menor en ciertos escenarios.

### Red U-Net 2D + residual

Este método, presentado por Wang et al [65], propone una red basada en la arquitectura U-Net, combinada con una capa residual. La red U-Net fue propuesta originalmente en 2015 [66] para segmentación de imágenes médicas. Su arquitectura se muestra de manera gráfica en la figura 2.15. La arquitectura típica de la red U-Net corresponde a una red neuronal profunda de convolución (DCNN), compuesta por una serie de bloques de convolución que reducen la dimensionalidad de los datos para extraer las características relevantes, seguido de una serie de convoluciones que aumentan la dimensionalidad para volver a la forma original de los datos.

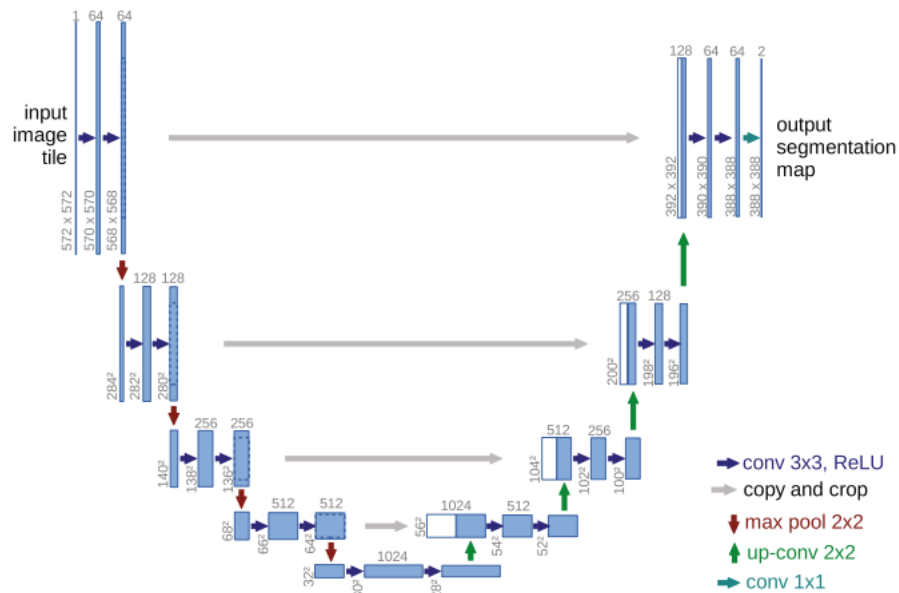


Fig. 2.15: Arquitectura de red U-Net propuesta por Ronneberger, et al [66]. Los números en la parte inferior de cada caja representan el tamaño de la imagen en esa etapa y los de la parte superior indican la cantidad de canales, mientras que las flechas denotan operaciones descritas en la parte inferior derecha.

Esta arquitectura destaca por su capacidad de producir muy buenos resultados con cantidades comparativamente bajas de datos de entrenamiento. Este modelo ha tenido amplia adopción y sigue siendo mejorada por diversas investigaciones [67, 68, 69, 70].

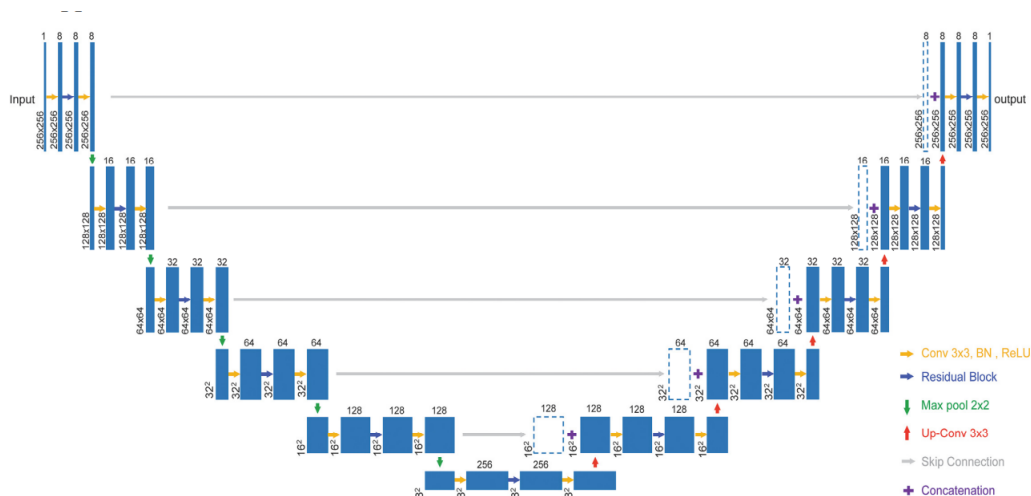


Fig. 2.16: Arquitectura de la red neuronal propuesta por Wang [65]. Los números en la parte inferior de cada caja representan el tamaño de la imagen en esa etapa y los de la parte superior indican la cantidad de canales, mientras que las flechas denotan operaciones descritas en la parte inferior derecha.

La propuesta de Wang agrega sobre la implementación original una capa residual, que permite pasar mayor información entre capas, reduciendo el tiempo de entrenamiento del modelo. La arquitectura completa de la red se observa en la figura 2.16, donde al comparar con la figura 2.15 se pueden observar los cambios propuestos. Se observa la forma característica de la arquitectura U-Net, pero formada por los bloques residuales mencionados. El modelo está diseñado para entradas de tamaño 256x256 píxeles o datos, y llega a 49 capas de convolución para extraer las principales características de los datos. Esta arquitectura modificada fue diseñada para realizar desdoblamiento de fase en imágenes de fase.

### Red U-Net 1D

Este modelo corresponde a la adaptación de la DCNN anterior para su funcionamiento con arreglos unidimensionales. La estructura de capas y bloques se mantiene, pero se deben ajustar las dimensiones de entrada y salida de dichos bloques. Se modifica para ir de entradas matriciales de 256x256, a entradas unidimensionales de 256x1, para que el modelo acepte directamente señales envueltas individuales generadas en las mediciones.

### Red Long Short-Term Memory (LSTM)

El último modelo de machine learning que se ha implementado en este trabajo es el Long Short-Term Memory, un tipo de red neuronal recurrente (RNN) propuesta en 1997 por Hochreiter y Schmidhuber [71], con el objetivo de resolver el problema de desvanecimiento de gradiente que afectaba a muchos modelos de ese tiempo. En pocas palabras, el problema del desvanecimiento

de gradiente describe el efecto por el cual cada capa de una red neuronal se vuelve menos sensible a las entradas anteriores, y “olvida” las primeras entradas [72].

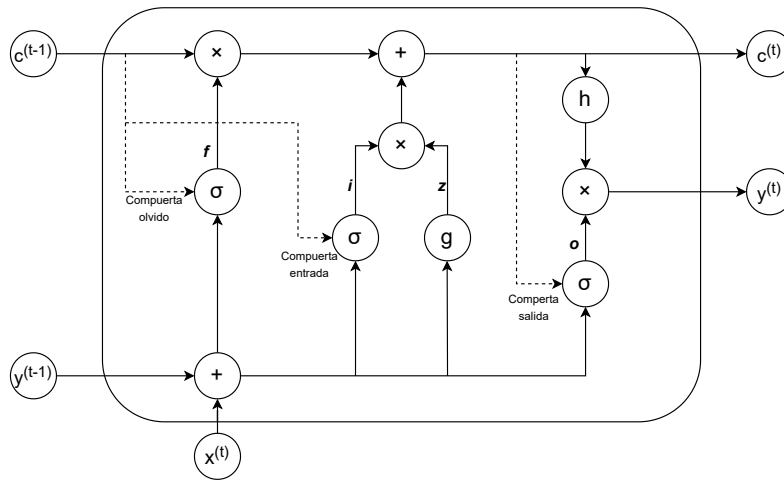


Fig. 2.17: Bloque fundamental LSTM clásico.  $\times$  indica multiplicación,  $+$  suma elemento a elemento,  $\sigma$  la función de activación sigmoide,  $g$  la función de activación de entrada y  $h$  la de salida.

La unidad fundamental de una red LSTM clásica es el bloque de memoria. Estos bloques se conectan de manera secuencial para construir la red. En la figura 2.17 se muestra un bloque de memoria y sus componentes internos. El flujo del bloque se puede describir en 6 pasos [73]:

*Actualización de entradas:* El bloque recibe la entrada actual  $x^{(t)}$  y la salida del mismo bloque en la iteración anterior  $y^{(t-1)}$ .

*Cálculo de la compuerta de entrada:* Se combinan  $x^{(t)}$ ,  $y^{(t-1)}$  y el valor de la celda anterior  $c^{(t-1)}$  ponderados por los pesos y sesgos del modelo para obtener el valor de activación de la compuerta de entrada  $i^{(t)}$  y los valores candidatos a agregarse al estado de la celda  $z^{(t)}$ .

*Cálculo de la compuerta de olvido:* Se decide qué información descartar del estado anterior  $c^{(t-1)}$  calculando la activación de la compuerta de olvido  $f^{(t)}$  a partir de  $x^{(t)}$ ,  $y^{(t-1)}$  y  $c^{(t-1)}$ .

*Actualización del estado de la celda:* El nuevo valor de la celda  $c^{(t)}$  se calcula combinando la información retenida en la compuerta de olvido y la información nueva determinada por la compuerta de entrada y los candidatos  $z^{(t)}$ .

*Cálculo de la compuerta de salida:* Usando  $x^{(t)}$ ,  $y^{(t-1)}$  y el estado de la celda  $c^{(t)}$  se determina la activación de la compuerta de salida  $o^{(t)}$ .

*Cálculo de la salida del bloque:* La salida final  $y^{(t)}$  se obtiene aplicando la compuerta de salida al valor actualizado de la celda.

## 2. MARCO TEÓRICO

---

La arquitectura LSTM destaca en su funcionamiento con arreglos unidimensionales de datos donde el orden de la secuencia es importante, como series de tiempo [74] y procesamiento de lenguaje escrito [75] y hablado [76]. Esto lo hace muy atractivo para su uso en el desenvolvimiento de fase de este trabajo. En [77], se propone una red LSTM para el desenvolvimiento de fase en condiciones sub-óptimas, obteniendo resultados comparables a los de otras redes neuronales profundas, destacando un costo computacional muy reducido.

## 3 Implementación experimental

### 3.1 Métodos de desenvolvimiento de fase

#### 3.1.1 *Unwrap* de MATLAB

Este método está incluido en el software MATLAB, por lo que no requiere de mayor trabajo para implementarlo. Como se mencionó anteriormente, debido a su fácil acceso y su buen rendimiento en una amplia variedad de casos es utilizado en muchos trabajos que requieren de desenvolvimiento de fase, como un método “default”. Se incluye en este trabajo para dar una base a las comparativas que se realizan, y por su relevancia en investigaciones similares.

#### 3.1.2 Basado en ecuación de transporte de intensidad (TIE)

Este método fue propuesto por Zhao et al en [50] como una versión más liviana en términos de cómputo que versiones anteriores del mismo método. Zhao además compartió su implementación del método en MATLAB, encontrándose libremente accesible en *MATLAB file exchange* [53]. Se resalta que a pesar de que este método está hecho para mapas de fase, es decir, datos bidimensionales de fase, realiza el procesamiento por arreglos, es decir, se puede aplicar directamente a casos unidimensionales.

#### 3.1.3 Filtro de Kalman *Unscented*

Para la implementación del UKF se trabaja en base al modelo presentado en [78], el cual utiliza el filtro para desenvolver la fase de *synchronphasors*. En el monitoreo de redes eléctricas se utilizan dispositivos conocidos como PMU para adquirir datos de frecuencia y fase de la red con alta precisión para monitoreo y control [79]. En la publicación recién mencionada, se aplica el UKF a estos datos. Al ser series temporales con datos de fase, habría una alta transferencia al caso de este proyecto.

El filtro se codifica en MATLAB, considerando la relación entre fase envuelta y desenvuelta:

$$\phi_u(k) = \phi_w(k) + 2\pi\gamma(k) \quad (3.1)$$

En la cual  $\phi_u(k)$  y  $\phi_w(k)$  son la fase desenvuelta y envuelta en un instante  $k$  dado, respectivamente. El número que corrige la fase está representado por  $\gamma$ . Las ecuaciones de estado y observación del filtro se muestran a continuación [78]:

$$x(k+1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x(k) + w = f(x(k)) + w \quad (3.2)$$

$$z(k) = \begin{bmatrix} \cos(x_1(k)) \\ \sin(x_1(k)) \end{bmatrix} + v = h(x(k)) + v \quad (3.3)$$

En la ecuación 3.2, el vector de estado en el instante  $k$  está representado por:

$$x(k) = \begin{bmatrix} \phi_u(k) \\ \Delta_u(k) \end{bmatrix} \quad (3.4)$$

Donde  $\phi_u(k)$  es la fase desenvuelta y  $\Delta_u(k)$  su variación temporal, siguiendo el modelo de desenvolvimiento de fase presentado en la sección anterior. El término  $w$  corresponde al ruido de proceso, modelado como ruido gaussiano de media cero. En la ecuación 3.3,  $z(k)$  representa el vector de mediciones en el instante  $k$  y  $v$  es el ruido de medición, también modelado como ruido gaussiano de media cero. La medición de fase envuelta no se usa directamente, sino que se representa en forma seno-coseno:

$$z(k) = \begin{bmatrix} \cos(\phi_w(k)) \\ \sin(\phi_w(k)) \end{bmatrix} \quad (3.5)$$

Esta transformación mapea la fase envuelta a un punto en el círculo unitario, evitando las discontinuidades que ocurren en  $\pm\pi$ . Así se asegura que la función de medición  $h(k)$  sea suave y periódica, lo que a su vez permite que el UKF opere de manera efectiva ante los saltos de  $2\pi$  en la fase envuelta.

Con estas definiciones, el filtro opera ejecutando sus etapas normales: cálculo de puntos sigma, predicción de estado, y corrección en base a las mediciones.

#### 3.1.4 Red Neuronal U-Net

La red U-Net que se implementa para las pruebas en este proyecto corresponde a la anteriormente mencionada U-Net residual propuesta en [65]. El autor incluye en la publicación un repositorio abierto [80] donde se encuentra el código utilizado para entrenar e implementar la red neuronal. Esta implementación se encuentra realizada en Python utilizando la librería Pytorch, ampliamente usada para tareas de machine learning.

La red U-Net se implementa en dos formatos: unidimensional y bidimensional. En el repositorio compartido por Wang, la red está diseñada para imágenes de entrada, por lo que no hay que aplicar cambios a la red para poder utilizarla con las señales DAS. Para el formato unidimensional las dimensiones de los datos de entrada deben ser ajustados para que la red admita señales de series de tiempo individuales. Esto se logra modificando el tamaño de los diferentes bloques de la red, para asegurar que el flujo de datos sea compatible con el nuevo formato de entrada.

Para el entrenamiento de estas redes se utiliza una GPU P100 en un servicio de cómputo en la nube que provee acceso a hardware especializado para tareas de entrenamiento de modelos de manera remota, para disminuir el tiempo total de entrenamiento. Luego, los archivos con los pesos de la red se obtienen para poder procesar datos de manera local.

#### 3.1.5 Red Neuronal LSTM

Actualmente, existen librerías que permiten implementar el modelo LSTM en una red de manera muy eficiente, incluso con una sola línea de código. Para poder extraer las características de los datos y potenciar el funcionamiento de la capa LSTM, se incluyen capas para cambiar la forma y extraer relaciones de los datos previa o posteriormente a su paso por la capa LSTM. Esto permite un aprendizaje más profundo y asegura que la forma de los datos a la salida del modelo tenga sentido y sea consistente con la entrada.

En trabajos similares [81, 82], se utilizan capas de convolución, pooling y densas para extraer características locales, reducir parámetros y luego la clasificación/regresión [83]. Para este caso, se implementa la red con capas de pooling, convolución y densas. La red resultante posee 228576 parámetros tuneables. Se entrena con 40000 datos de fase en la ya mencionada GPU P100 en la nube, con batch size de 64 y 100 épocas.

#### 3.1.6 Datos de entrenamiento para métodos de machine learning

Para generar datos de entrenamiento para los distintos modelos se usa el *script* de MATLAB descrito en [65], e implementado por el mismo autor en [80]. Este programa fue creado originalmente en el contexto del entrenamiento de una red neuronal para el desenvolvimiento de fase

en dos dimensiones.

En el caso de las señales DAS, la fase envuelta proviene directamente de las mediciones interferométricas, pero la fase desenvuelta no puede obtenerse de manera directa ni con precisión absoluta. Esto significa que no existen bases de datos que contengan fase envuelta y su correspondiente fase desenvuelta como “verdad de terreno”.

Por esta razón, el uso de datos sintéticos resulta fundamental: permite disponer de la fase desenvuelta exacta para cada señal envuelta, y además se pueden controlar parámetros como el nivel de ruido o la estructura espacial de la fase. El programa que se utiliza genera señales de fase envuelta junto con su fase desenvuelta a partir de matrices “base”, que se transforman en mapas de fase ajustando distintos parámetros como el ruido blanco y la amplitud, entre otros.

Para los métodos de desenvolvimiento 2D, esta salida se puede usar directamente, configurando los parámetros de manera que las características de las señales sean lo más similares posible a las esperadas en las mediciones DAS. En el caso de los métodos unidimensionales, se realizan las modificaciones necesarias para que las matrices de fase sean simplemente señales con forma de series de tiempo.

#### **3.1.7 Estrategias de desenvolvimiento de fase**

Los métodos de desenvolvimiento de fase se dividen en dos categorías: los unidimensionales y los bidimensionales. En general, el desenvolvimiento de fase para datos obtenidos con detección  $\Phi$ -OTDR se desenvuelven en una dirección: temporal, desenvolviendo un punto de la fibra a través del tiempo para caracterizar una perturbación, o espacial, desenvolviendo la fase de la traza obtenida de la señal de RBS generada por un solo pulso incidente. El desenvolvimiento bidimensional es usado generalmente para mapas de fase, como los generados en inSAR [2], donde se aplica el desenvolvimiento de fase a una imagen o mapa de fases.

En el siguiente capítulo se compara el rendimiento de los métodos unidimensionales en el tiempo y espacio por separado, y luego el rendimiento del método bidimensional. Se propone realizar un desenvolvimiento “pseudo-bidimensional”, desenvolviendo la fase en el espacio y luego en el tiempo con cada método unidimensional. La comparativa se hace tanto con las fases experimentales obtenidas en el laboratorio como con fases artificiales generadas para poner a prueba los métodos.

Para aplicar el desenvolvimiento de fase para cada método, se crea un script en MATLAB que permite llamar a cada método y así realizar el procesamiento. El flujo de este programa sigue los pasos descritos a continuación.

Primero, se importa el archivo de datos preprocesado, extrayendo la matriz de fases envueltas. Luego, dependiendo del desenvolvimiento a realizar, se extrae una fila o columna, y se le aplica la función correspondiente al método a usar. Esto devuelve un arreglo con las fases desenvueltas,

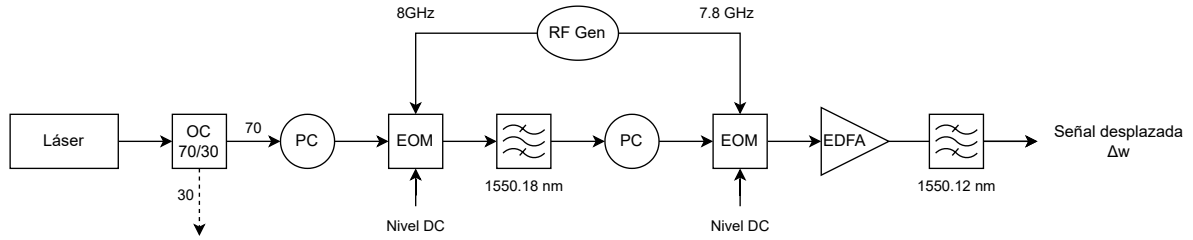


Fig. 3.1: Oscilador local del esquema  $\Phi$ -OTDR heterodino.

las cuales se pueden guardar, graficar o volver a procesar.

En el caso de las pruebas de métodos unidimensionales, se realiza la prueba sobre una sola traza o punto temporal según sea el caso, para facilitar la visualización de los datos y la comparación de resultados.

## 3.2 Esquema $\Phi$ -OTDR heterodino

Para la obtención de datos experimentales se emplea un esquema  $\Phi$ -OTDR heterodino, con algunos cambios debido a disponibilidad de equipos en laboratorio. Los cambios principales se deben a la no disponibilidad de un AOM, por lo que el corrimiento de frecuencia necesario para el funcionamiento del esquema se realiza en el oscilador local mediante una serie de modulaciones ópticas y filtrados. A continuación se describe con más detalle cada sección del esquema.

### 3.2.1 Oscilador local

Como se mencionó en el capítulo 2, para la detección heterodina se necesita un corrimiento de frecuencia entre el pulso incidente y el oscilador local. Al no contar con un AOM, que permite modular y generar esta diferencia de frecuencia en un solo paso, se utilizan una serie de moduladores electro-ópticos (EOM por sus siglas en inglés) y filtrados, junto con amplificación mediante EDFAs para lograr el corrimiento de frecuencia. El esquema del oscilador local se muestra en la figura 3.1, y la evolución de los espectros en cada paso se puede visualizar en la figura 3.2.

### 3. IMPLEMENTACIÓN EXPERIMENTAL

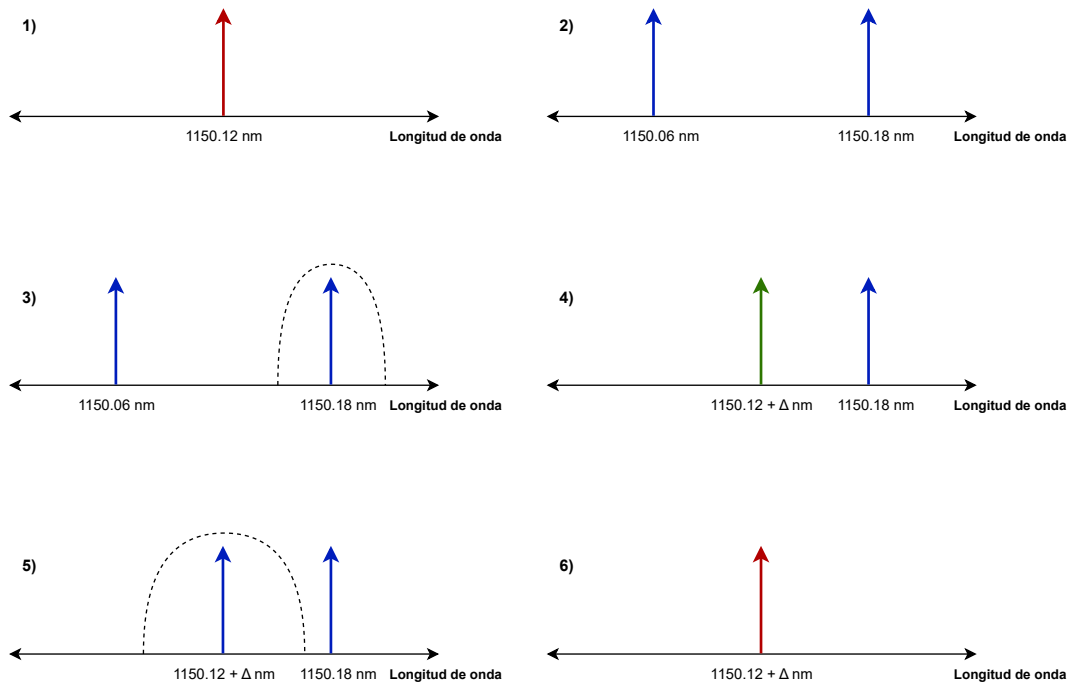


Fig. 3.2: Evolución del espectro de la señal óptica a medida que pasa por el oscilador local. Las flechas corresponden a los espectros ideales, y las líneas punteadas indican filtrado.

El fotodetector utilizado tiene un ancho de banda de 300 MHz, por lo que el espectro de la señal batida debe ser menor a este ancho de banda. Es por esto que el corrimiento de frecuencia que se usa en el oscilador local es de 200 MHz. Para lograr esto se utiliza el esquema de oscilador local mostrado en la figura 3.1. Primero, la luz pasa por un controlador de polarización. Se utilizan varios de éstos a través del esquema completo para poder ajustar la polarización y así maximizar la señal eléctrica al momento de realizar las mediciones. Luego se realiza la primera modulación óptica mediante un EOM. Este EOM modula la señal con una sinusoidal a 8 GHz generada por un generador de radiofrecuencia, llevando la longitud de onda del láser de 1550.12 nm a 1550.18 nm. Es importante notar que al modular mediante un EOM, se generan dos bandas laterales, en este caso a 1550.18 y 1550.6 nm. La existencia de bandas laterales en la señal batida afecta de manera negativa la detección coherente, por lo que luego de cada modulación con EOM la señal debe ser filtrada para dejar solo una de las bandas laterales. Entonces, luego de la modulación sigue un filtro de ancho 32 pm centrado en 1550.18 nm para eliminar la banda lateral generada en 1550.06 nm. Recordando que la meta es generar un corrimiento de frecuencia de 200 MHz, nuevamente se modula la señal con un EOM y señal sinusoidal, esta vez con frecuencia de 7.8 GHz. Antes de realizar el filtrado para eliminar las nuevas bandas laterales, se usa un EDFA para amplificar la señal óptica, y luego se filtra con un filtro de ancho 50 pm y una frecuencia central de 1550.12 nm. El filtrado se hace después de la amplificación para eliminar el ruido generado

al utilizar el EDFA. Finalmente, antes de entrar al detector balanceado, la señal pasa por otro controlador de polarización.

#### 3.2.2 Generación de pulsos

Los pulsos que se usan para interrogar la fibra son pulsos rectangulares simples, generados mediante el uso de un amplificador óptico de semiconductor (SOA) y un EOM en cascada. En esta configuración, el SOA genera un pulso óptico más ancho dentro del cual se genera el pulso modulado por el EOM. El pulso utilizado en el SOA es de 140 ns, mientras que el del EOM es de 50 ns. El SOA tiene una tasa de extinción de aproximadamente 50 dB, mientras que la del EOM es de aproximadamente 40 dB. Al usarlos en cascada se busca que la señal que entre a la fibra fuera del pulso interrogador sea lo más baja posible. La sincronización de los pulsos se logra alimentando ambos desde un equipo generador de señales, ajustando la fase de las señales para lograr el efecto buscado. En la figura 3.3 se muestra la configuración del brazo del esquema correspondiente a la generación de pulsos, y en la figura 3.4 se muestra un diagrama del funcionamiento de los pulsos en cascada del SOA y EOM.

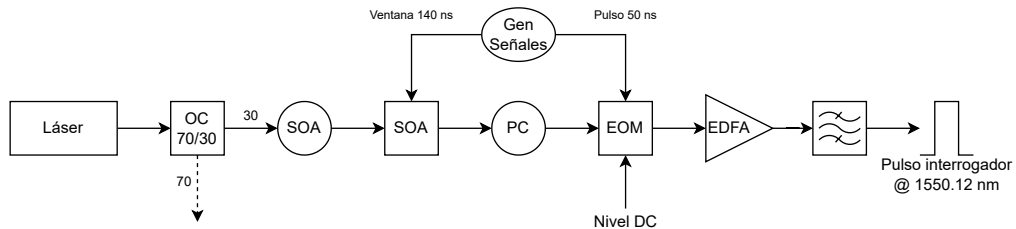


Fig. 3.3: Sección del esquema para generar pulsos interrogadores.

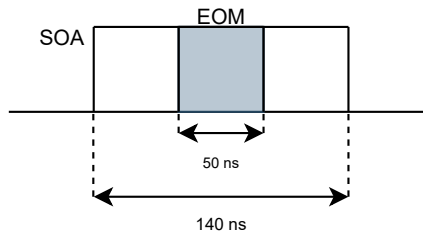


Fig. 3.4: Sincronización de pulsos del SOA y EOM. El EOM modula el pulso ya modulado por el SOA para mejorar la razón de extinción.

### 3.2.3 Esquema implementado

Con las secciones de esquema mostradas anteriormente se tiene el esquema completo para realizar la interrogación de la fibra y la detección. El esquema completo se muestra en la figura 3.5. La luz de un láser coherente opera en 1550.12 nm, y pasa por un acoplador óptico 70/30, donde el 70 % de la luz va al oscilador local, y el 30 % a la generación de pulso. Al final del oscilador local se encuentra un controlador de polarización para ajustar la polarización. Luego de la generación del pulso hay un acoplador 99/1, donde el 1 % de la luz va a un atenuador y luego a un fotodetector para monitorear el pulso. El 99 % restante va al circulador que permite la entrada del pulso a la fibra y la salida de la señal de RBS, la cual se amplifica con un EDFA, posteriormente filtrándose para eliminar el ruido óptico producido por la amplificación y finalmente a un controlador de polarización.

Ambas señales se baten en el fotodetector balanceado, donde se convierte la señal óptica en una señal eléctrica para su posterior muestreo.

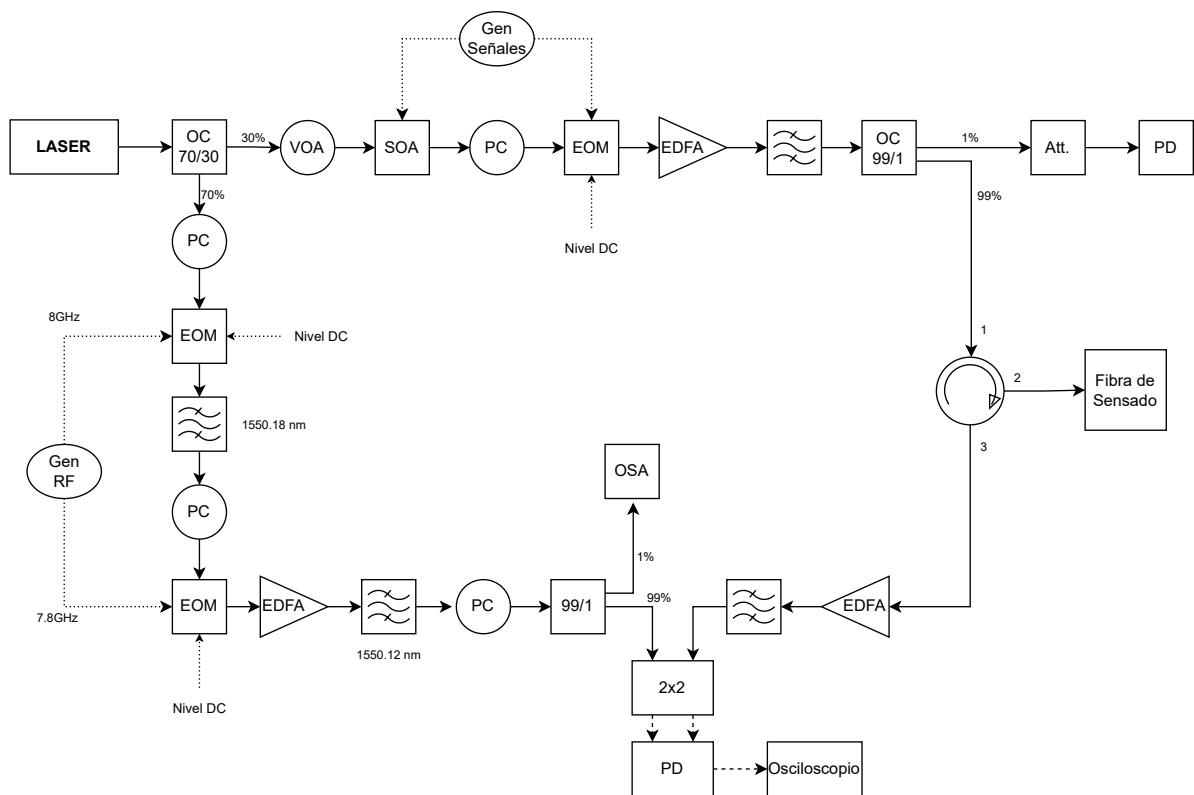


Fig. 3.5: Esquema completo de detección  $\Phi$ -OTDR heterodino. Las líneas sólidas indican señales ópticas, y las punteadas señales eléctricas.

## 3.3 Mediciones experimentales

Como se mencionó anteriormente, para interrogar la fibra se usan pulsos rectangulares simples, de ancho de 50 ns, entregando una resolución espacial de 5 m sobre una fibra de 25 km de largo. Dichos pulsos se generan como se describió en la sección anterior, interrogando la fibra y generando la señal de RBS, la cual es amplificada y filtrada, para luego ser batida con la señal del oscilador local en un fotodetector balanceado.

Para generar trazas de diferentes calidades, se varía la potencia del pulso incidente entre 5 dBm y 20 dBm, midiendo la potencia de entrada a la fibra en ese rango. El límite inferior de 5 dBm se establece porque, por debajo de este valor, los pulsos no pueden distinguirse del piso de ruido del sistema, lo que impide asegurar un nivel de potencia confiable. El límite superior de 20 dBm, en cambio, se fija para no salir de la zona de respuesta lineal de la fibra. La señal es captada por el fotodetector ya mencionado y la señal eléctrica resultante registrada con un osciloscopio a 1 Gbps, realizando 200 capturas para cada nivel de potencia.

Las potencias de pulso entregada a la fibra fueron incrementando de a 1 dB, entre 5 y 20 dB. La potencia de entrada se conoce gracias a la previa caracterización del esquema y fotodetector utilizando un *power meter*.

## 3.4 Procesamiento de datos

Una vez obtenidas las mediciones de las trazas, estas deben ser procesadas para extraer la información de fase. El procedimiento comienza con una etapa de filtrado para reducir el ruido y aislar la señal de interés. A continuación, se realiza la demodulación mediante el esquema IQ, a partir del cual se calcula la fase utilizando la función arcotangente. Finalmente, antes de aplicar los métodos de desenvolvimiento, se incorpora una etapa de preprocesamiento con el algoritmo SERVS para mejorar la calidad de la señal, descrito en el capítulo 2. El flujo completo de este procesamiento se resume en el diagrama de la figura 3.6.

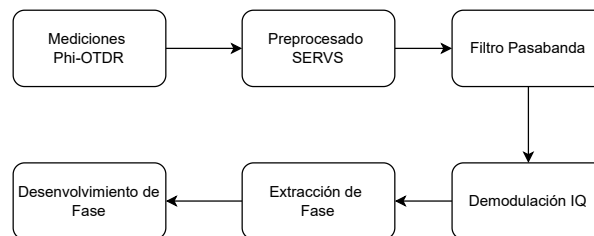
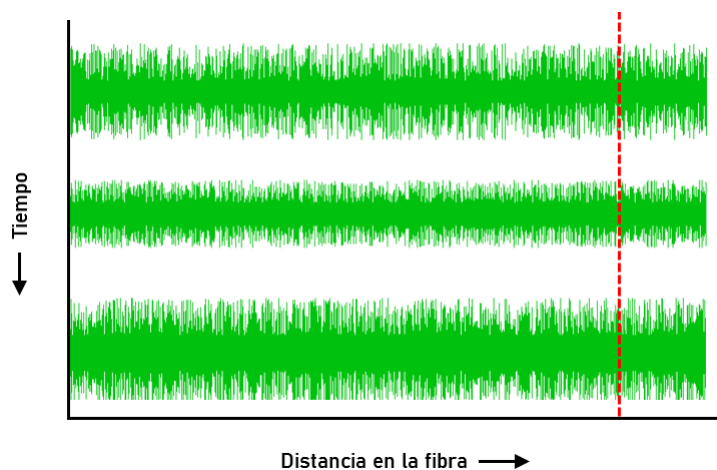


Fig. 3.6: Flujo de procesamiento de datos adquiridos.

## 4 Resultados

### 4.1 Datos Obtenidos

Los datos obtenidos a través del esquema implementado en el laboratorio se organizan en muestras consecutivas a lo largo de la fibra de sensado. El procesamiento de éstos se realiza en dos sentidos: espacial y temporal. Para aclarar lo que esto significa se presentan ambos casos en la figura 4.1. La medición consiste en una serie de capturas del RBS de la fibra en un instante de tiempo. Organizando estas capturas de forma consecutiva en una matriz se obtiene la evolución temporal de la fibra completa a través del tiempo en el que se muestrea.



*Fig. 4.1:* Organización temporal y espacial de las mediciones. Las trazas verdes representan una traza a lo largo de la fibra, y los puntos que pasan por la línea roja punteada representan los datos asociados a una misma posición a través del tiempo.

En la gran mayoría de los métodos clásicos de desenvolvimiento de fase se aplica el desenvolvimiento en una de las dos direcciones: temporal o espacial. En este trabajo se aplicarán los diferentes métodos en ambas direcciones por sí solas, además de aplicarlas en ambas direcciones de forma consecutiva, evaluando el desempeño en cada caso, para cada método.

### 4.1.1 Traza sin perturbación

Primero se obtienen 200 trazas para cada nivel de potencia de entrada a la fibra. Antes de realizar los desenvolvimientos de fase se realiza una caracterización de dichas trazas, calculando la SNR para cada nivel de potencia. Esta SNR se calcula en la traza de intensidad, antes del paso de extracción de fase. En la figura 4.2 se muestran las SNR de las trazas de potencias de entrada 5, 10, 15, y 20 dBm. Para la mayor potencia de entrada, se obtuvo una SNR máxima de  $\approx 45$  dBm, mientras que para la menor fue de  $\approx 30$  dBm. También se puede observar cómo a medida que se avanza en la fibra, la SNR decae, bajando entre 7 y 9 dB en los 25 km de fibra. Cabe notar que la relación entre SNR y potencia de entrada a la fibra es lineal: la diferencia de SNR para entradas de potencia de 20 dBm y 5 dBm fue de unos 15 dBm.

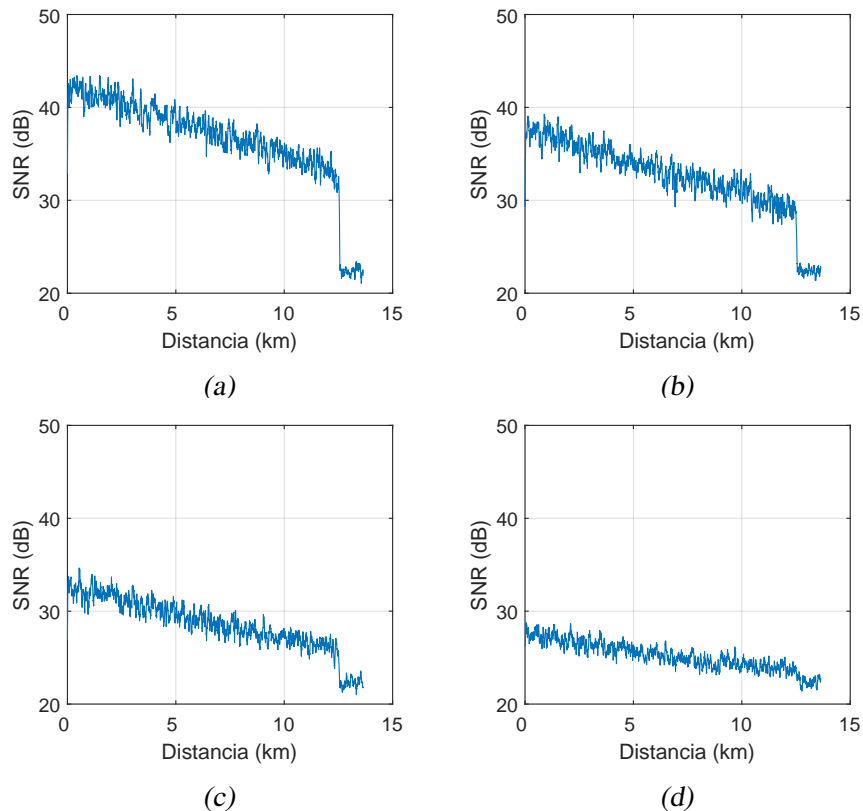


Fig. 4.2: SNR's eléctricas para potencias ópticas de entrada a) 20 dB, b) 15 dB, c) 10 dB, d) 5 dB

En estas gráficas también se observa cómo el preprocesamiento con SERVS elimina los puntos de desvanecimiento a lo largo de la traza, ayudando a mantener un nivel de SNR más constante a lo largo de la fibra.

### 4.1.2 Traza con perturbación

Además de las mediciones anteriormente descritas, se obtienen trazas adicionales para distintos niveles de potencia aplicando una perturbación al final de la fibra mediante un estirador de fibra. Este induce estiramientos en una sección de fibra que se encuentra envuelta alrededor de un piezoeléctrico. La perturbación se define con un generador de señales. El esquema completo de la fibra con el estirador se muestra en la figura 4.3.

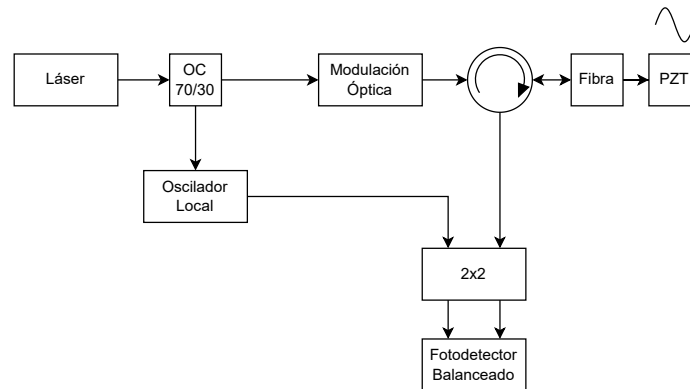


Fig. 4.3: Diagrama simplificado del esquema utilizado para realizar mediciones con perturbación en la fibra.

Se inducen perturbaciones sinusoidales de 20 y 200 Hz, y se mide al igual que en el caso sin perturbación, en una fibra de 25 km a distintos niveles de potencia a la entrada de la fibra, muestreando a 1 GSps.

La diferencia clave entre el caso sin perturbación y con perturbación es que en el segundo caso se tiene una señal de referencia conocida que se busca extraer de los datos desenvueltos. Esto permite comparar de forma directa los resultados obtenidos con cada método, y compararlos de manera cuantitativa.

## 4.2 Desenvolvimiento de fase con datos experimentales sin perturbación

Es importante recalcar que al tratarse de fases medidas experimentalmente sin perturbación, no se tiene una apreciación cuantitativa de la precisión de cada método, sino que se realiza un análisis cualitativo comparando los métodos entre sí, y considerando formas típicas de fases desenvueltas.

## 4. RESULTADOS

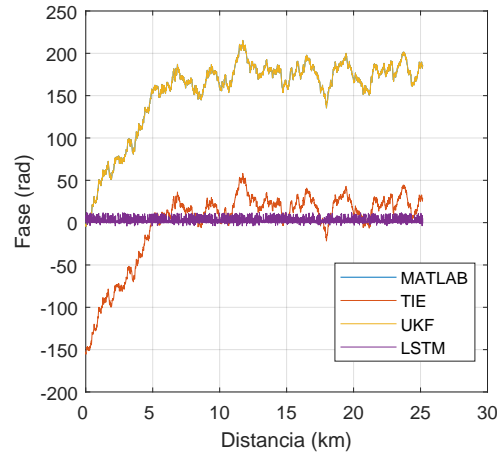


Fig. 4.4: Desenvolvimiento de fase en sentido espacial para señal de pulso 20 dBm.

En la figura 4.4 se muestra el resultado de algunos métodos desarrollando la misma fase en el sentido espacial, es decir, la traza resultante de un pulso incidente en la fibra. Como se puede ver, los métodos unwrap y UKF obtienen virtualmente los mismos resultados, mientras que el método TIE sigue la misma forma, pero con un desfase producto de un error en el cálculo de la fase inicial. Para cuantificar la diferencia entre la señal desenvuelta por la función de MATLAB y el UKF se calcula el error cuadrático medio entre ambas señales, obteniendo un RMSE de 0,0241 radianes, confirmando la alta similitud entre ambas señales.

Notar que el método U-Net no se incluye en los resultados mostrados en este capítulo. Aunque este método fue implementado y evaluado, sus resultados fueron significativamente inferiores a los de los demás métodos. Es por esto que no se incluyen en los gráficos y errores presentados, ya que solo entorpecería la comparación de los demás datos sin aportar información relevante.

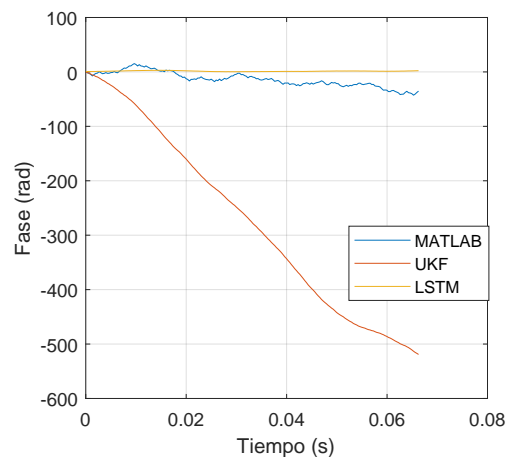


Fig. 4.5: Desenvolvimiento de fase en sentido temporal para señal con pulso de 20 dBm.

## 4. RESULTADOS

Pasando al unwrapping temporal, realizado en la misma posición de la fibra a lo largo de las 200 mediciones, se obtienen los resultados mostrados en la figura 4.5. Se observa cómo el método de UKF falla en esta instancia, sobrecorrectiendo saltos de fase y generando una fase desenvuelta errónea. Para el método de MATLAB se obtiene una señal que mantiene una fase baja, mientras el LSTM oscila alrededor de cero radianes. Esto ocurre ya que uno de los métodos determina que un salto es detectado como un salto de fase real, mientras que el otro no lo corrige al determinar que se trata de ruido. Se destaca que el método de TIE no se incluye en las estimaciones de fase en sentido temporal ya que el algoritmo deja de funcionar en este sentido. Al iterar puede deberse a una divergencia de la estimación al tener menor relación entre datos consecutivos.

Para el pulso de 10 dBm, se observa una tendencia similar, tal como lo muestran las figuras 4.6 y 4.7.

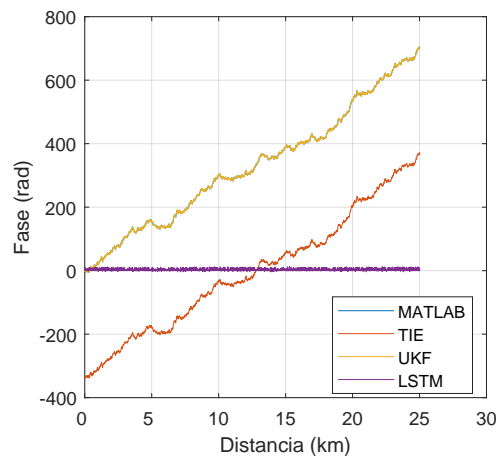


Fig. 4.6: Desenvolvimiento de fase en sentido espacial para señal de pulso 10 dBm.

Los métodos de MATLAB y UKF obtienen prácticamente los mismos resultados, mientras que el método TIE obtiene una traza similar pero con un desfase de  $\approx 390$  radianes. Este desfase no implica una falla en el método en sí, puesto que se puede observar que la fase desenvuelta sigue la misma tendencia que los métodos de MATLAB y UKF. El método de machine learning nuevamente obtiene una fase que rodea los cero radianes.

En el caso de DAS, la información importante se obtiene de las fases diferenciales, es decir, de las diferencias de las fases consecutivas en el tiempo por lo que el valor exacto de la fase inicial estimada no impacta en el resultado final mientras este sea consistente en el tiempo.

## 4. RESULTADOS

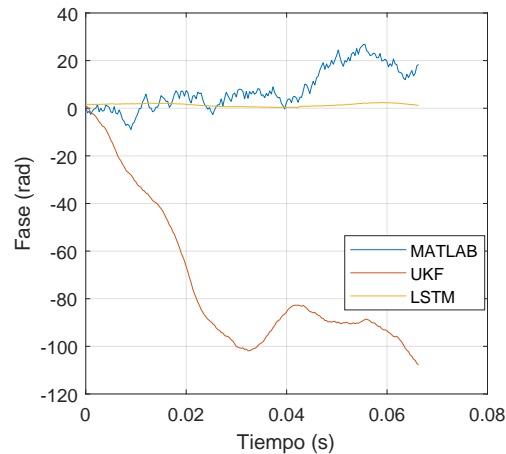


Fig. 4.7: Desenvolvimiento de fase en sentido temporal para señal de pulso 10 dBm.

En la señal temporal de la figura 4.7, nuevamente el UKF diverge en su estimación de la fase, mientras que MATLAB y LSTM siguen produciendo resultados similares a los casos anteriores. Se observa en general para los métodos de MATLAB y UKF que se están sobrecompensando saltos de fase, y este error se está acumulando a lo largo de la señal.

Es relevante notar que para señales obtenidas con esquemas  $\Phi$ -OTDR, las trazas espaciales desenvueltas obtenidas siguen la forma general de trazas similares en la literatura, indicando a primera vista que los métodos dan resultados válidos para datos experimentales.

Luego se realiza el desenvolvimiento de fase en ambas direcciones para cada método: primero en dirección temporal y luego en dirección espacial. Para el pulso de 20 dBm y 10 dBm se grafica una traza temporal y espacial. Primero se realiza con el método unwrap de MATLAB:

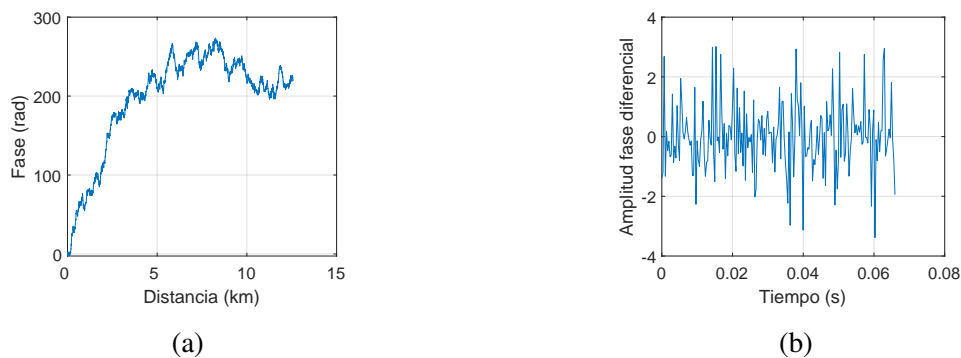
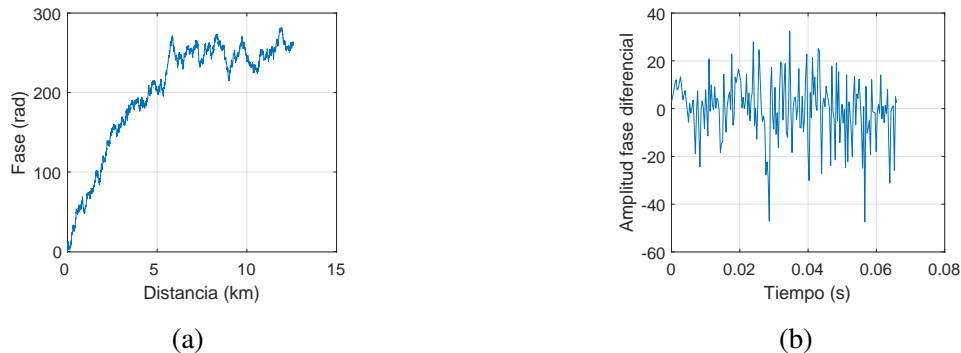


Fig. 4.8: Resultados de desenvolvimiento cuasi-bidimensional usando unwrap de MATLAB sobre la señal con pulso de 20 dBm. a) corresponde a una traza desenvuelta, mientras que b) a un punto en la fibra a través del tiempo.

## 4. RESULTADOS

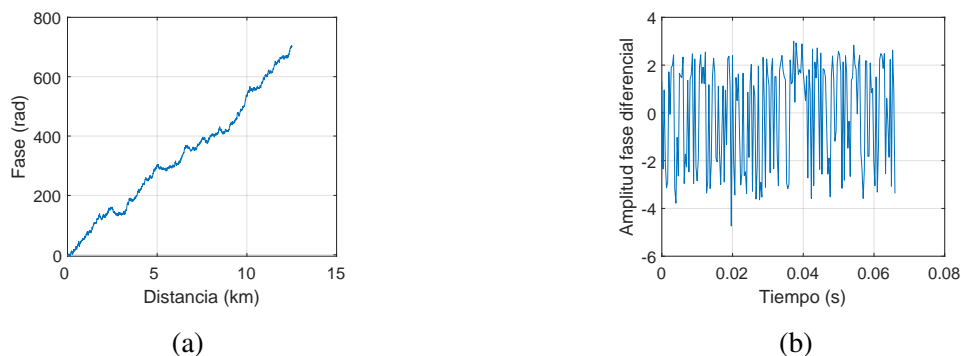
Y luego con el método UKF:



*Fig. 4.9:* Resultados de desenvolvimiento cuasi-bidimensional usando UKF sobre la señal con pulso de 20 dBm. a) corresponde a una traza desenvuelta, mientras que b) a un punto en la fibra a través del tiempo.

Como se puede ver, la traza espacial obtenida para ambos métodos es muy similar, y mantiene la forma del desenvolvimiento hecho en una dimensión anteriormente, pero con una fase acumulada que llega hasta aproximadamente 50 radianes, dado que la fase inicial estimada por el método de MATLAB es cercano a -200 radianes. Esto resulta relevante ya que el método TIE estimó esa misma fase en su versión unidimensional, en la figura 4.4, lo que indicaría que ahora el método de MATLAB se acerca al TIE, que sería el método más robusto. Esto a su vez significaría que el desenvolvimiento en dos sentidos mejoró el desempeño de unwrap de MATLAB. Además, se mantiene el bajo rendimiento del UKF a la hora de desenvolver las fases de un punto de la fibra a través del tiempo, a pesar de haber desenvuelto en dos direcciones.

El proceso se repite para la señal con pulso de entrada de 10 dBm:



*Fig. 4.10:* Resultados de desenvolvimiento cuasi-bidimensional usando unwrap de MATLAB sobre la señal con pulso de 10 dBm. a) corresponde a una traza desenvuelta, mientras que b) a un punto en la fibra a través del tiempo.

## 4. RESULTADOS

---

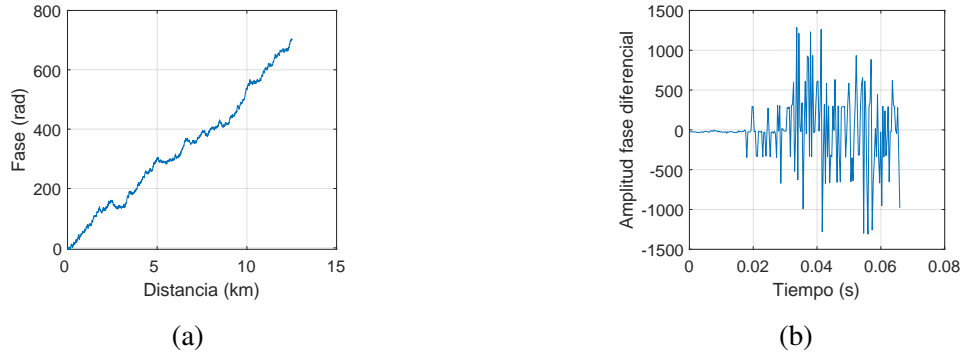


Fig. 4.11: Resultados de desenvolvimiento cuasi-bidimensional usando UFK sobre la señal con pulso de 10 dBm. a) corresponde a una traza desenvuelta, mientras que b) a un punto en la fibra a través del tiempo.

En este caso, se obtienen resultados similares en las dos dimensiones, con fases espaciales que llegan al orden de 700 radianes, indicando un bajo rendimiento al estar muy por encima de las fases estimadas en el caso ideal del pulso de 20 dBm.

Es importante notar que el método TIE no se incluye en las pruebas cuasi-2D, ya que al correrlo con la matriz completa de datos en ambas direcciones, detiene su funcionamiento dentro de los primeros desenvolvimientos. Este método es iterativo, por lo que es probable esto se deba a un problema de convergencia resultante de realizar desenvolvimientos sobre fases ya procesadas.

### 4.3 Desenvolvimiento de fase con datos experimentales con perturbación

El siguiente paso consiste en aplicar los métodos de desenvolvimiento al conjunto de datos de la fibra bajo perturbación. Como se mencionó anteriormente, la señal de referencia es una senoide de 20 Hz o 200 Hz, según el caso. En la sección anterior, los gráficos mostrados correspondían a una traza individual de la fibra, presentando la fase a lo largo de toda su extensión. En esta sección, en cambio, los resultados corresponden a una zona específica de la fibra observada en el tiempo, ubicada hacia el extremo donde se encuentra el piezoeléctrico que introduce la perturbación. Se comienza nuevamente con el uso de la función *unwrap()* de MATLAB, aplicada al caso de una perturbación de 20 Hz con un pulso de 20 dBm:

## 4. RESULTADOS

---

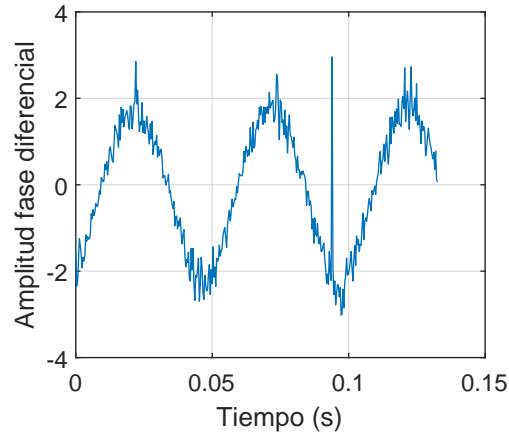


Fig. 4.12: Señal temporal rescatada utilizando *unwrap()* en sentido espacial para la señal de 20Hz a 20 dBm de potencia de entrada a la fibra.

Como se puede ver, la forma de onda se rescata de muy buena manera, siguiendo la forma esperada manteniendo la frecuencia de la perturbación. Para el mismo método pero en sentido temporal se tiene:

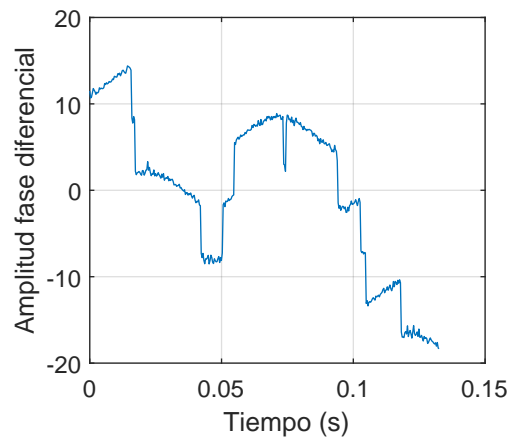
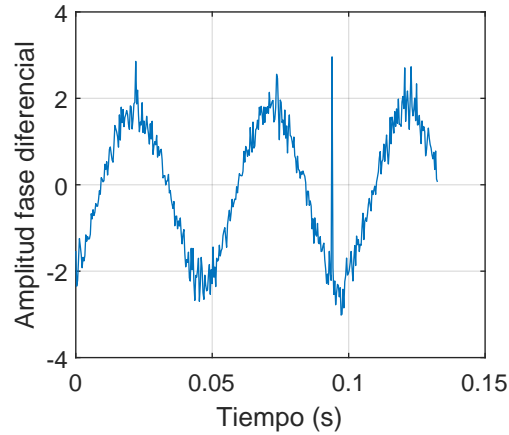


Fig. 4.13: Señal temporal rescatada utilizando *unwrap()* en sentido temporal para la señal de 20 Hz a 20 dBm de potencia de entrada a la fibra.

En este caso se sigue observando el fenómeno visto en las trazas sin perturbación: al aplicar el desenrollamiento en sentido temporal (siguiendo un punto en el tiempo en vez de una traza en el espacio) la calidad de la señal final baja considerablemente. En este caso se pueden ver los cambios de dirección asociados a la oscilación de la senoide, pero para efectos prácticos es una señal inutilizable. Siguiendo al método TIE:

## 4. RESULTADOS

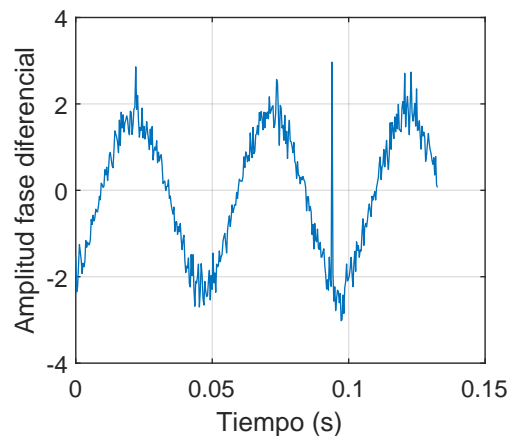
---



*Fig. 4.14:* Señal rescatada utilizando TIE en sentido espacial para la señal de 20 Hz a 20 dBm de potencia de entrada a la fibra.

Para el caso de TIE, en el desenvolvimiento espacial se obtienen resultados muy similares a la función base de MATLAB, destacando el punto entre 0.08 y 1 segundo donde se produce un salto considerable de fase el cual se corrige en seguida. Al igual que en la sección anterior, al realizar el desenvolvimiento de fase de manera temporal con este método se produce un error donde el desenvolvimiento se detiene durante el proceso. Como se mencionó, se especula que se debe a una divergencia en el cómputo de la fase, provocando un estado indefinido en el algoritmo. Es por esto que este método no será mencionado más adelante en esta sección.

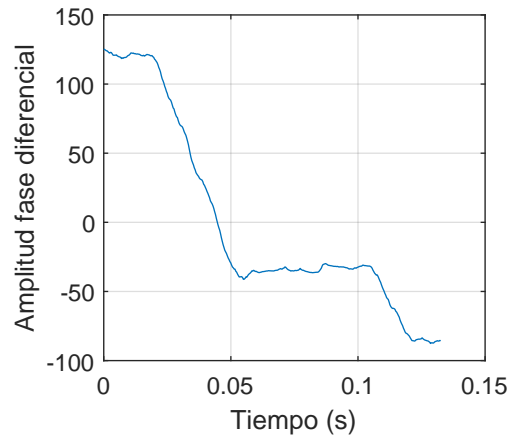
Para el filtro de Kalman los primeros resultados son:



*Fig. 4.15:* Señal rescatada utilizando UKF en sentido espacial para la señal de 20 Hz a 20 dBm de potencia de entrada a la fibra.

## 4. RESULTADOS

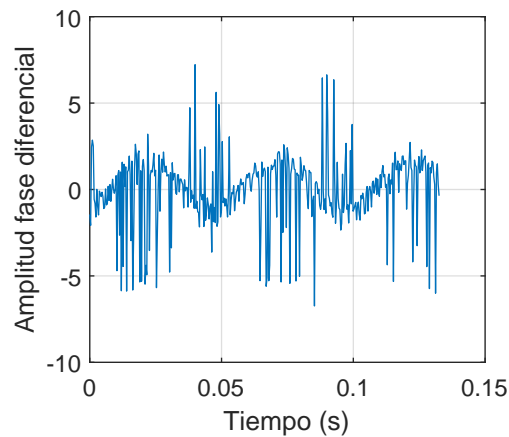
---



*Fig. 4.16:* Señal rescatada utilizando UKF en sentido temporal para la señal de 20 Hz a 20 dBm de potencia de entrada a la fibra.

Nuevamente en el caso espacial es rescatada la sinusoide de buena forma, mostrando que en condiciones favorables estos tres métodos obtienen resultados similares. Para el caso temporal se observa un resultado completamente alejado de lo esperado, generando una curva de fase que aumenta de manera continua sin oscilaciones.

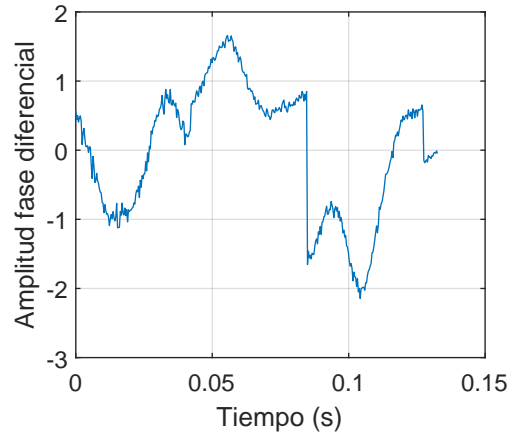
Finalmente, se realiza con el método LSTM:



*Fig. 4.17:* Señal rescatada utilizando LSTM en sentido espacial para la señal de 20 Hz a 20 dBm de potencia de entrada a la fibra.

## 4. RESULTADOS

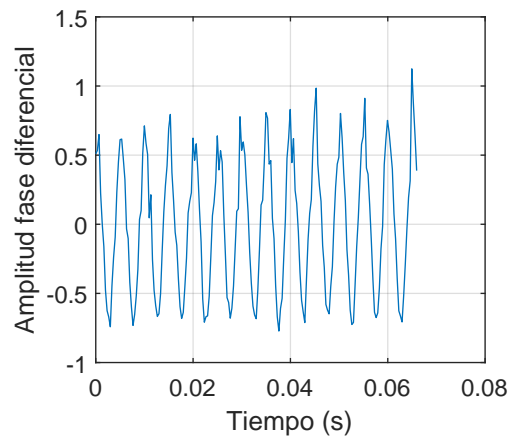
---



*Fig. 4.18:* Señal rescatada utilizando LSTM en sentido temporal para la señal de 20 Hz a 20 dBm de potencia de entrada a la fibra.

Se observa una pérdida de la continuidad de la fase en varios puntos de la fase desenvuelta en sentido espacial, mientras que en sentido temporal se obtiene una señal más fidedigna que los demás casos temporales, pero donde la señal sigue alejada de una buena representación debido a la presencia de saltos de fase.

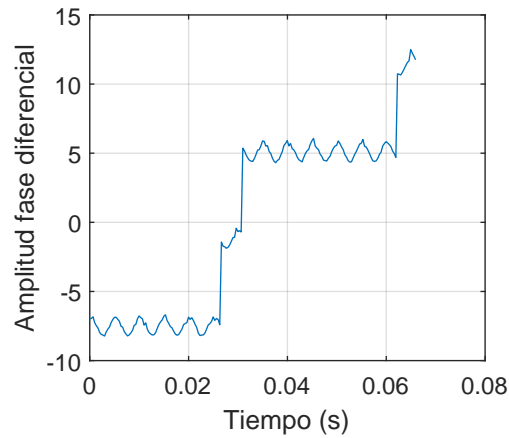
Se realiza el mismo análisis para la señal de 200 Hz, con potencia 20 dBm a la entrada de la fibra. Para `unwrap()` de MATLAB:



*Fig. 4.19:* Señal rescatada utilizando `unwrap()` en sentido espacial para la señal de 200 Hz a 20 dBm de potencia de entrada a la fibra.

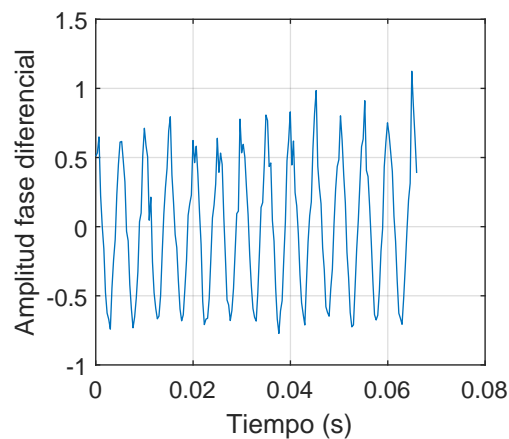
## 4. RESULTADOS

---



*Fig. 4.20:* Señal rescatada utilizando `unwrap()` en sentido temporal para la señal de 200 Hz a 20 dBm de potencia de entrada a la fibra.

Para la señal de mayor frecuencia se destaca que el desenvolvimiento temporal logra obtener la forma de onda de la perturbación a pesar de tener algunas sobrecompensaciones que se propagan, y sigue con peor desempeño que el desenvolvimiento espacial. El método TIE también rescata la perturbación:

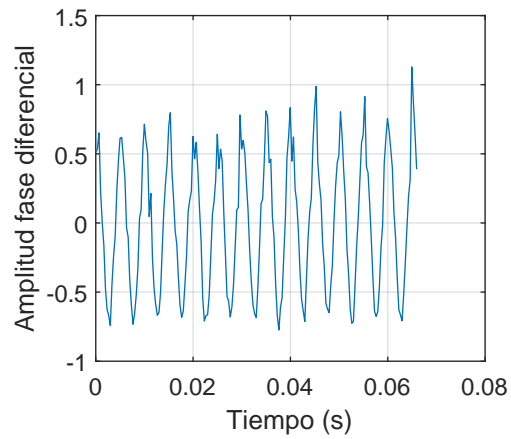


*Fig. 4.21:* Señal rescatada utilizando TIE en sentido espacial para la señal de 200 Hz a 20 dBm de potencia de entrada a la fibra.

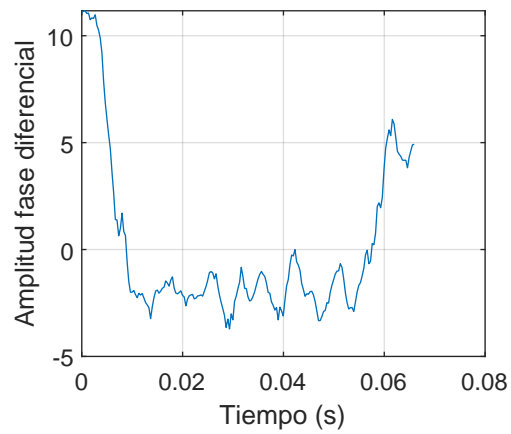
## 4. RESULTADOS

---

Al igual que el UKF espacial, mientras que el temporal falla:



*Fig. 4.22:* Señal rescatada utilizando UKF en sentido espacial para la señal de 200 Hz a 20 dBm de potencia de entrada a la fibra.

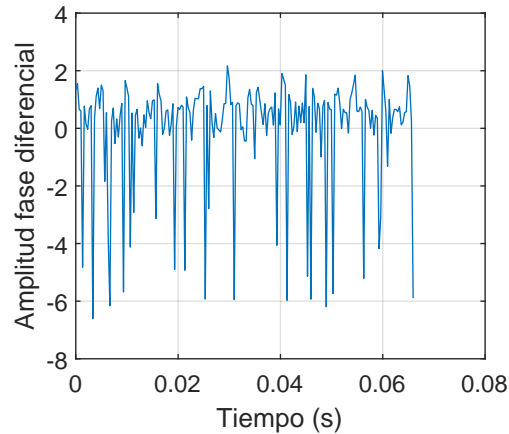


*Fig. 4.23:* Señal rescatada utilizando UKF en sentido temporal para la señal de 200 Hz a 20 dBm de potencia de entrada a la fibra.

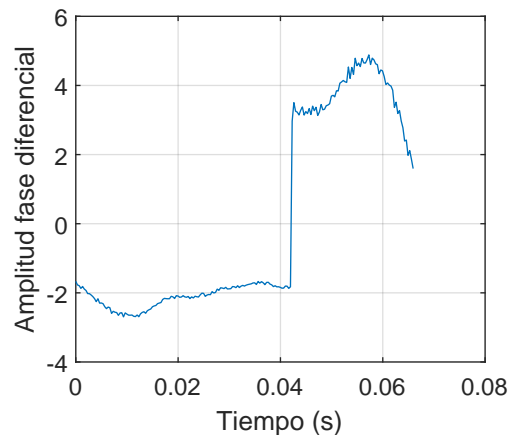
## 4. RESULTADOS

---

Para el método LSTM se tiene:



*Fig. 4.24:* Señal rescatada utilizando LSTM en sentido espacial para la señal de 200 Hz a 20 dBm de potencia de entrada a la fibra.



*Fig. 4.25:* Señal rescatada utilizando LSTM en sentido temporal para la señal de 200 Hz a 20 dBm de potencia de entrada a la fibra.

Para el método LSTM se ve que para la señal desenvuelta en sentido espacial de mayor frecuencia hay muchos más saltos que en el caso de 20 Hz del mismo método. En sentido temporal se rescata una sinusoide de baja frecuencia, lo que puede deberse a que en el entrenamiento del modelo los datos tendían a ser de menor frecuencia, lo que reduciría la robustez del LSTM ante señales de mayor frecuencia.

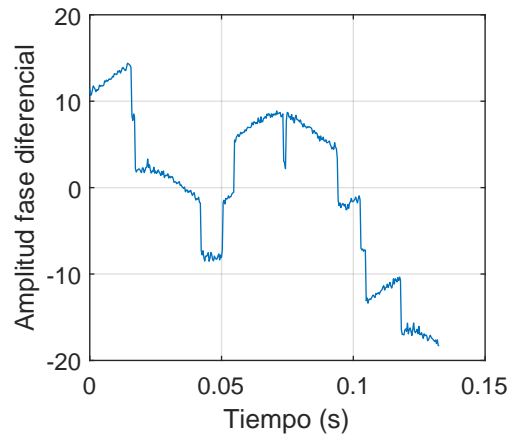
Se destaca que el filtro de Kalman logra extraer la forma general de la sinusoide parcialmente en el desenvolvimiento en sentido temporal, al igual que `unwrap()`. Esto podría ser indicativo de que usar la dirección temporal entrega mejores resultados en casos de perturbaciones de mayor frecuencia. La relevancia de este uso podría ser menor, ya que la gran mayoría de las

## 4. RESULTADOS

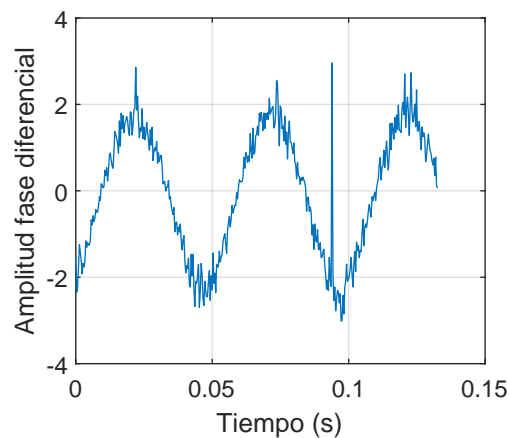
---

aplicaciones actuales de esquemas DAS se usan para detectar señales que se propagan a través de medios como edificaciones o el suelo, por lo que se las perturbaciones que se miden son de baja frecuencia.

Tal como en la sección anterior, se realiza el desenvolvimiento en dos direcciones, una después de la otra. Para `unwrap()` se obtiene:



*Fig. 4.26:* Señal rescatada utilizando `unwrap()`, primero en sentido espacial y luego temporal para la señal de 20 Hz a 20 dBm de potencia de entrada a la fibra.



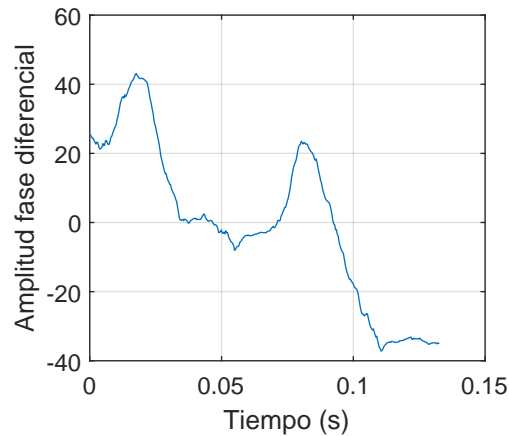
*Fig. 4.27:* Señal rescatada utilizando `unwrap()`, primero en sentido temporal y luego espacial para la señal de 20 Hz a 20 dBm de potencia de entrada a la fibra.

Para ambos casos no se nota mayor diferencia al aplicar una capa extra de desenvolvimiento.

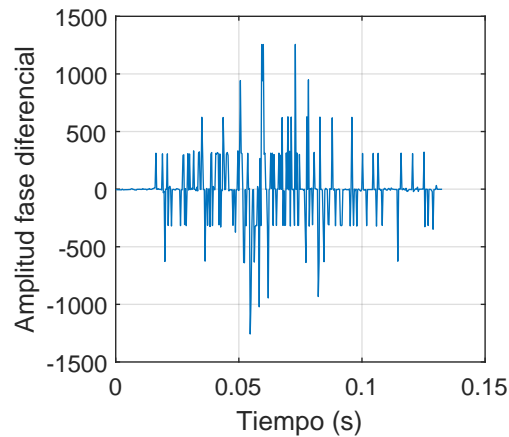
## 4. RESULTADOS

---

Para UKF se tiene:



*Fig. 4.28:* Señal rescatada utilizando UKF, primero en sentido espacial y luego temporal para la señal de 20 Hz a 20 dBm de potencia de entrada a la fibra.



*Fig. 4.29:* Señal rescatada utilizando UKF, primero en sentido temporal y luego espacial para la señal de 20 Hz a 20 dBm de potencia de entrada a la fibra.

En ambos casos para el UKF la calidad de la señal desenvuelta decae considerablemente al aplicarle un segundo desenvolvimiento, a diferencia de `unwrap()` que no tuvo mayores cambios. Esto puede deberse a que el UKF está diseñado para seguir una señal con cierta lógica temporal, donde no se deberían presentar grandes saltos en datos consecutivos. Pero al realizar el desenvolvimiento en sentido temporal cada punto en la fibra presenta grandes variaciones entre muestras.

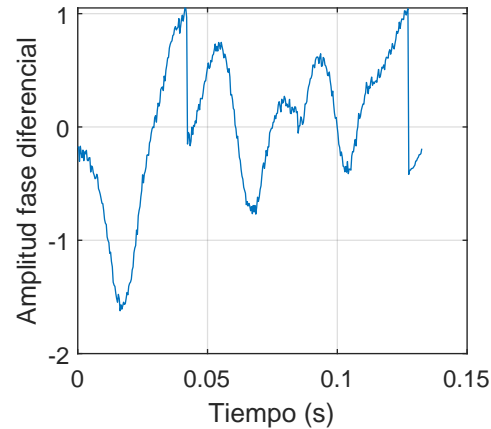


Fig. 4.30: Señal rescatada utilizando LSTM, primero en sentido espacial y luego temporal para la señal de 20 Hz a 20 dBm de potencia de entrada a la fibra.

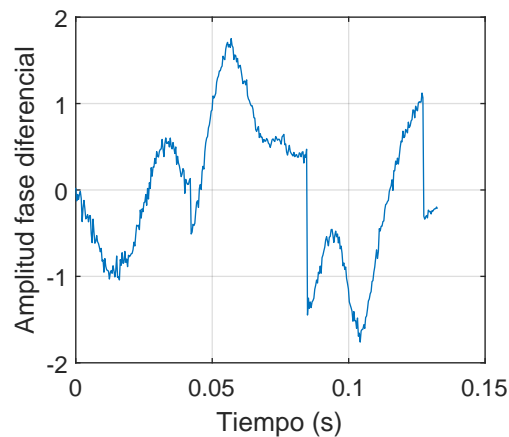


Fig. 4.31: Señal rescatada utilizando LSTM, primero en sentido temporal y luego espacial para la señal de 20 Hz a 20 dBm de potencia de entrada a la fibra.

A diferencia de los otros métodos, en el caso del LSTM se observan formas de onda similares para ambas permutaciones del orden de desenvolvimiento. Sin embargo, en ambos casos se obtiene un resultado peor que el uso de la dirección espacial por si sola.

Además del uso de distintos métodos de desenvolvimiento, existen distintas técnicas de post-procesado para mejorar los resultados y obtener fases más suaves y naturales. Uno de estos es el uso de medias móviles para disminuir el efecto de puntos que sean *outliers* extremos, producidos generalmente por ruido. Como se aprecia en los casos de perturbación de 20 Hz, hay un dato que genera un salto de fase extremo, el cual se podría suavizar con esta técnica. Se aplica una media móvil de ancho de 20 puntos para evitar distorsiones excesivas.

## 4. RESULTADOS

Aplicado en la señal de la figura 4.12, se obtiene:

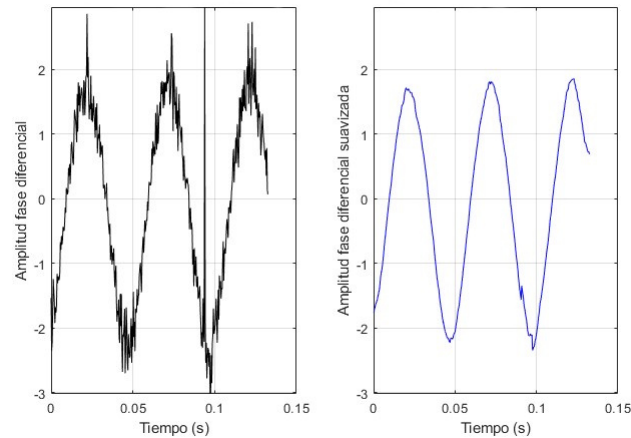


Fig. 4.32: Fase diferencial desenvuelta con MATLAB y la señal resultante al suavizar con media móvil de tamaño 20.

Esto es de especial utilidad en las señales de mayor frecuencia. Por ejemplo, en la señal de 200 Hz obtenida con UKF (figura 4.22):

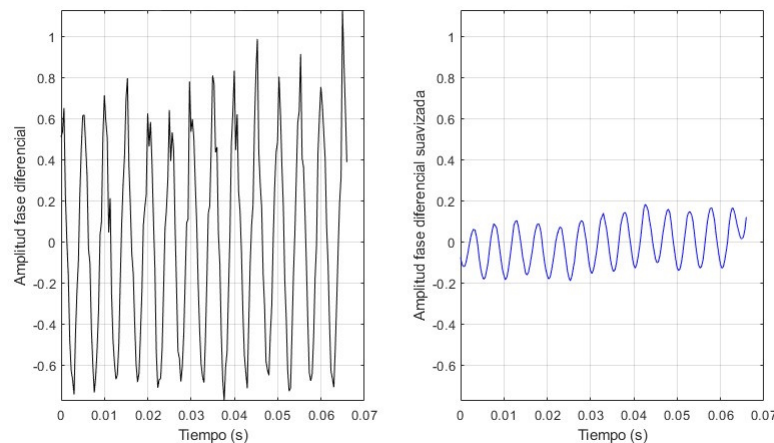


Fig. 4.33: Fase diferencial desenvuelta con UKF y la señal resultante al suavizar con media móvil de tamaño 20.

Se logra observar de manera mucho más limpia la señal sinusoidal rescatada, a cambio de una reducción de la amplitud de ésta. Esto es un *tradeoff* entre atenuación de la señal y rescatar una señal más suave y fácil de analizar a simple vista. Estos dos factores se deben considerar según el caso en el que se esté aplicando el promediado, pensando en la importancia de la fidelidad con la señal original y la obtención de una señal más suavizada.

## 4. RESULTADOS

---

Ajustando el ancho de la ventana se puede tener una señal suavizada pero con mayor amplitud:

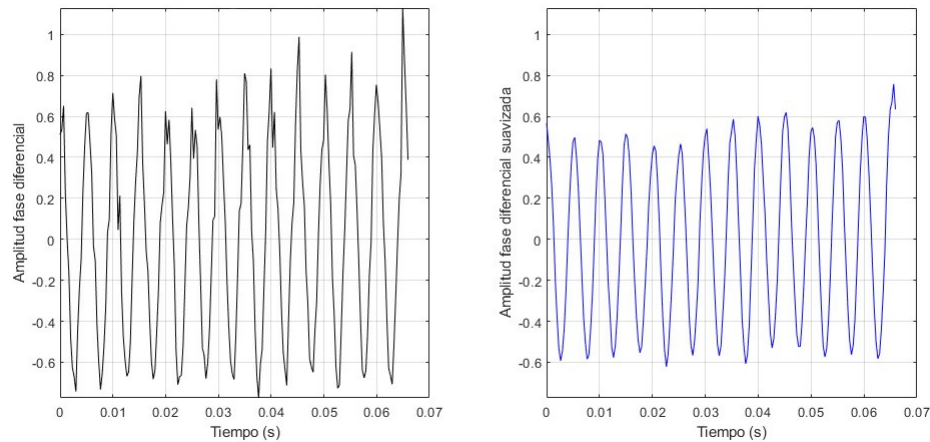


Fig. 4.34: Fase diferencial desenvuelta con UKF y la señal resultante al suavizar con media móvil de tamaño 5.

En el caso de la señal desenvuelta con LSTM, que posee mucho ruido aún siendo desenvuelta, el suavizado mejora levemente la señal:

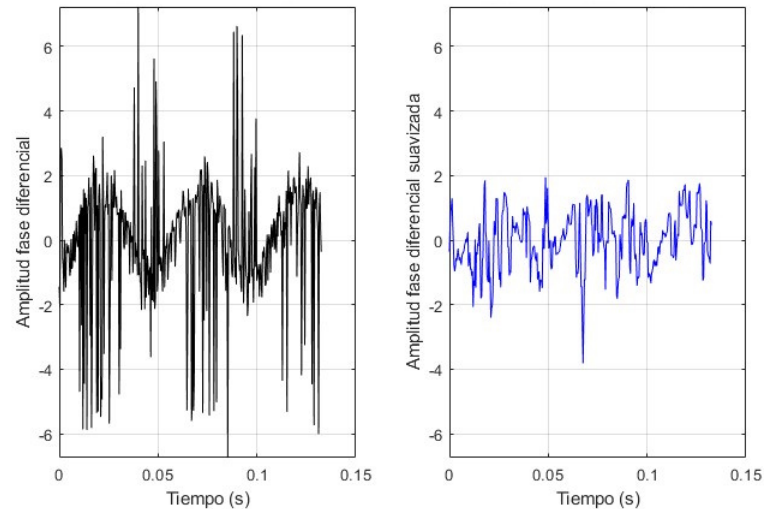


Fig. 4.35: Fase diferencial desenvuelta con LSTM y la señal resultante al suavizar con media móvil de tamaño 5.

### 4.4 Downsampling/Dropout

Una forma de medir la robustez de cualquier algoritmo que realiza tareas de regresión o similares es bajar la calidad de los datos y observar como baja la calidad de los resultados. El uso de

#### 4. RESULTADOS

---

dropout, o pérdida de datos es popular en distintas áreas [84], y consiste en eliminar datos de forma aleatoria o sistemática para simular errores en las mediciones.

Para realizar esta prueba, se realiza dropout de forma aleatoria de un cierto porcentaje de los datos de fase envuelta de las señales con perturbación, tanto de 20 Hz como de 200 Hz, con 20 dBm potencia de entrada a la fibra. El proceso consiste en seleccionar el porcentaje de datos a eliminar de cada traza, y generar números aleatorios dentro del rango de datos en las mediciones. Los datos cuyo índice coincide con los números generados son eliminados de la traza.

En estudios donde se trabajan con este tipo de datos, es muy común que parte del preprocesamiento de los datos incluya reparar datos faltantes en la series [85, 86]. Esto puede ser repitiendo el dato anterior, promediando los datos adyacentes, o el uso de algún tipo de regresión. En este caso, se usa una regresión lineal simple para rellenar los datos faltantes. Esto es un paso necesario ya que los algoritmos de desenvolvimiento de fase necesitan de datos continuos, y un dato clasificado como vacío o NaN (*not a number*) provocará un error en su ejecución.

Es importante notar que el dropout se realiza sobre los datos de fase envueltos, es decir, luego de demodulados de la señal de backscatter. De esta forma se aísla el rendimiento del desenvolvimiento, ya que si el dropout se hiciera sobre los datos de backscatter se estaría poniendo a prueba también el flujo de preprocesamiento, lo cual no está bajo prueba en este trabajo.

Una vez interpolados los datos que fueron eliminados, se procede a realizar el desenvolvimiento con normalidad para los distintos métodos. Para esta sección se realiza en sentido espacial, en vista de los resultados de las secciones anteriores.

Para cada método se realiza el desenvolvimiento con un dropout de 10 %, 30 % y 50 %, comparando los resultados con la señal desenvuelta sin dropout.

El punto de comparación de los diferentes casos será la perturbación obtenida con potencia de entrada 20 dBm, y desenvuelta con unwrap de MATLAB. Estas señales se muestran en la figura 4.36.

## 4. RESULTADOS

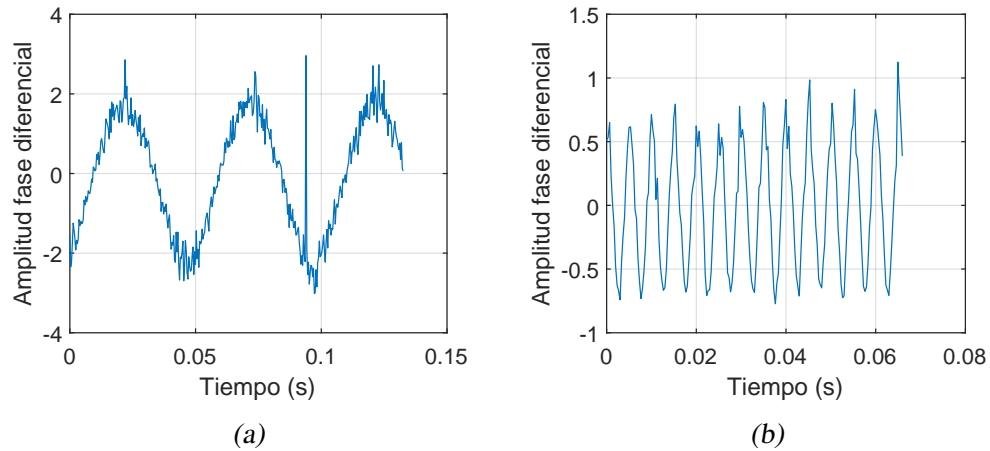


Fig. 4.36: Señales de referencia (sin dropout) para comparación de dropout. Perturbaciones obtenidas para potencia de entrada 20 dBm, con frecuencia a) 20 Hz y b) 200 Hz.

Comenzando con 10 % de dropout:

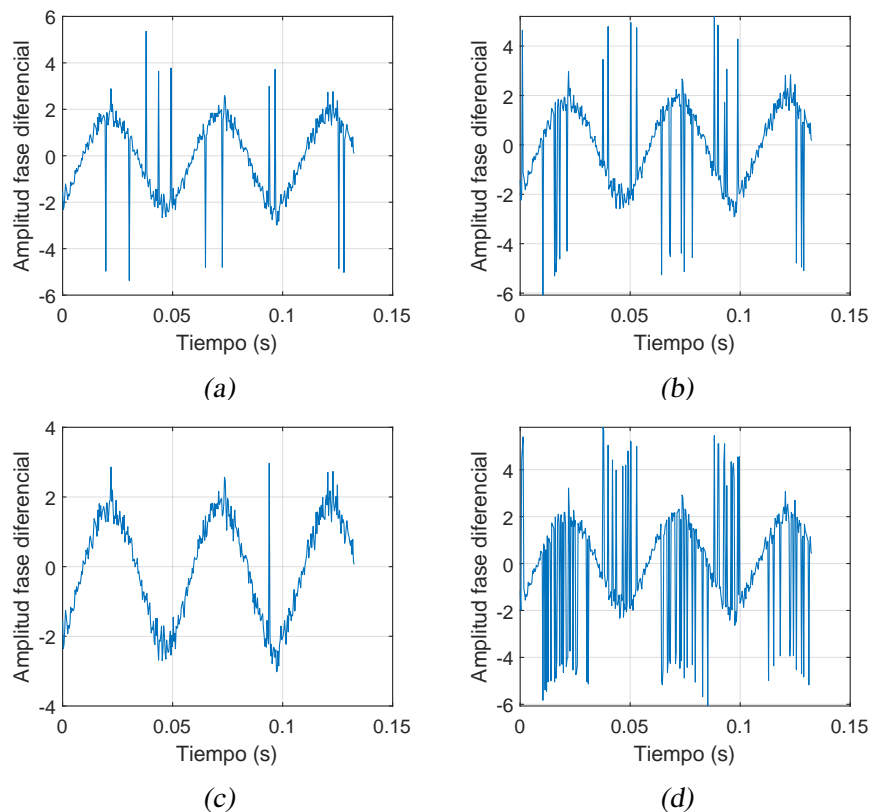


Fig. 4.37: Resultados de desenvolvimiento con perturbación de 20Hz y 10% dropout. a) MATLAB, b) TIE, c) UKF, d) LSTM.

Se observa que los métodos MATLAB y TIE obtienen resultados similares, con saltos considerables de fase a través de la forma de onda, debido a los puntos interpolados. Por otra parte, el

## 4. RESULTADOS

método UKF recrea la señal con un alto nivel de precisión. El método LSTM muestra las carencias que presentó en el desenvolvimiento original, pero es destacable que la calidad no baja de manera considerable sobre eso observando la forma de la onda a simple vista, es decir, a pesar de hacer un peor trabajo al desenvolver a primera vista es bastante robusto al dropout.

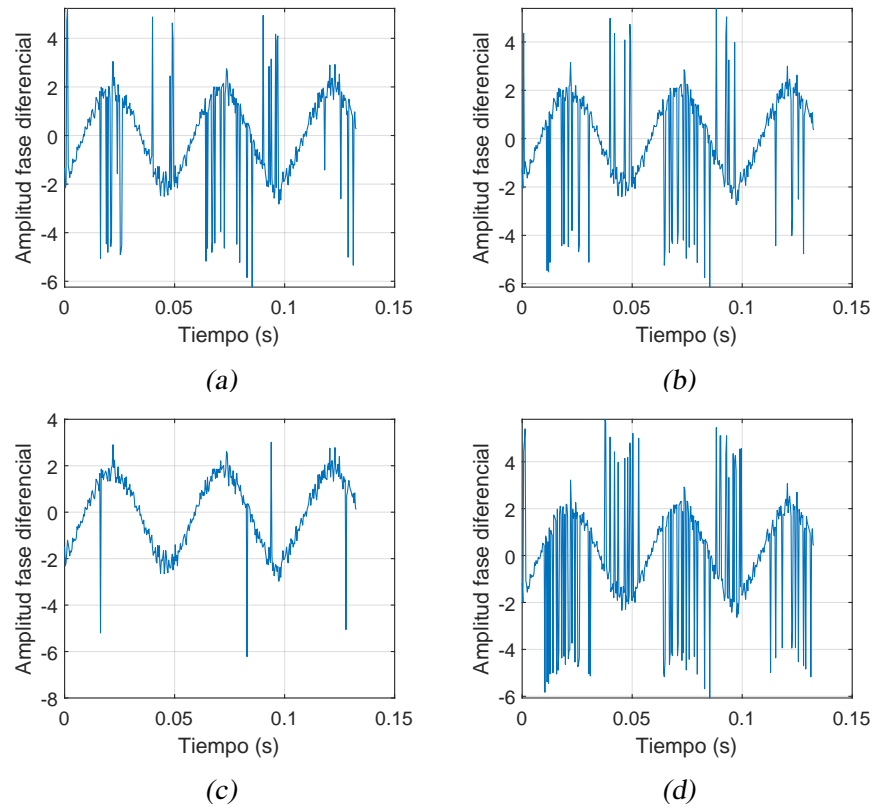
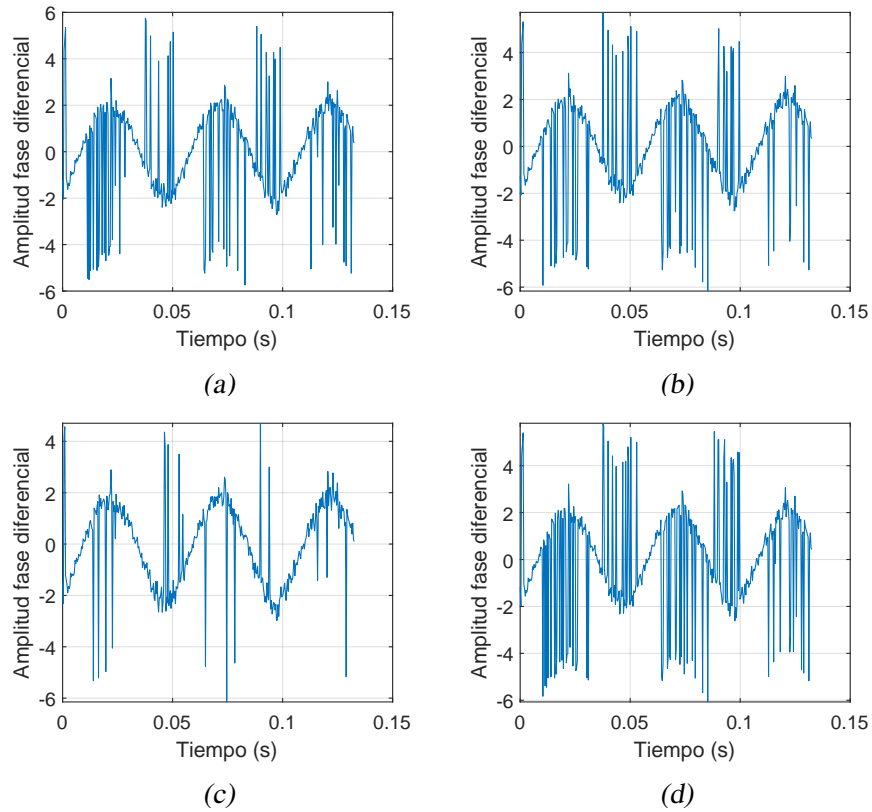


Fig. 4.38: Resultados de desenvolvimiento con perturbación de 20Hz y 30 % dropout. a) MATLAB, b) TIE, c) UKF, d) LSTM.

La dinámica continúa en el caso de 30 % de dropout, donde UFK obtiene resultados mucho mejores a MATLAB y TIE, entre los cuales, a simple vista, se obtienen resultados levemente mejores en TIE. LSTM nuevamente presenta muchos puntos fuera de fase, pero en una cantidad similar al caso de 10 %.

## 4. RESULTADOS

---



*Fig. 4.39:* Resultados de desenvolvimiento con perturbación de 20Hz y 50% dropout. a) MATLAB, b) TIE, c) UKF, d) LSTM.

En el último caso nuevamente destaca el filtro de Kalman, con una cantidad de puntos perdidos mucho menor a los demás métodos, los cuales muestran resultados similares entre ellos.

## 4. RESULTADOS

Se realiza el mismo análisis con la señal de 200 Hz:

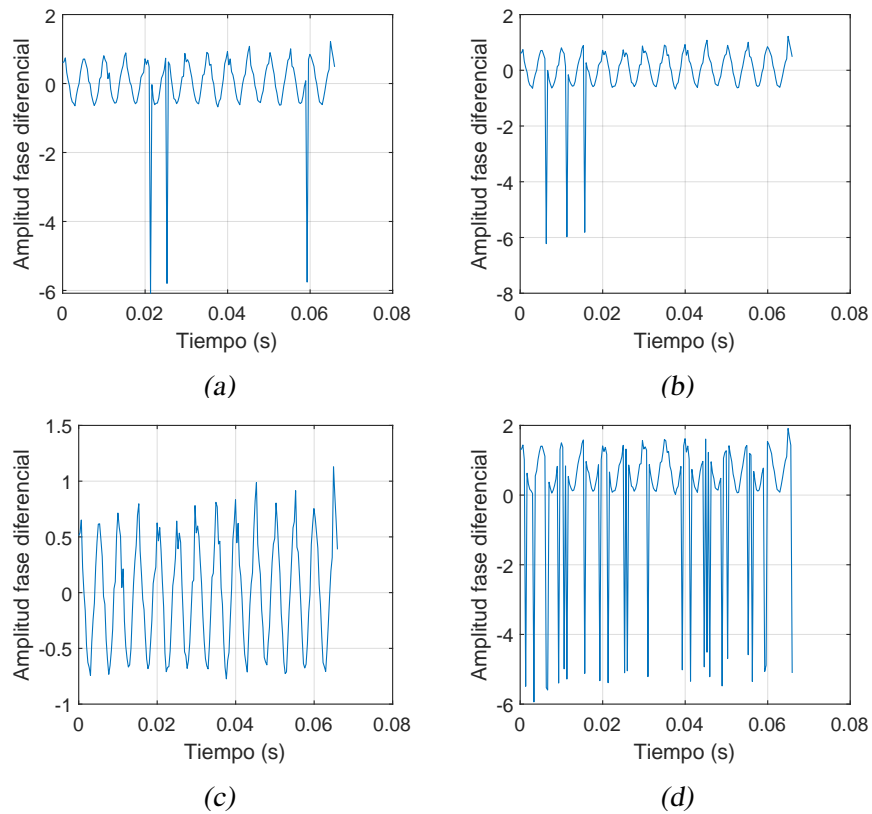
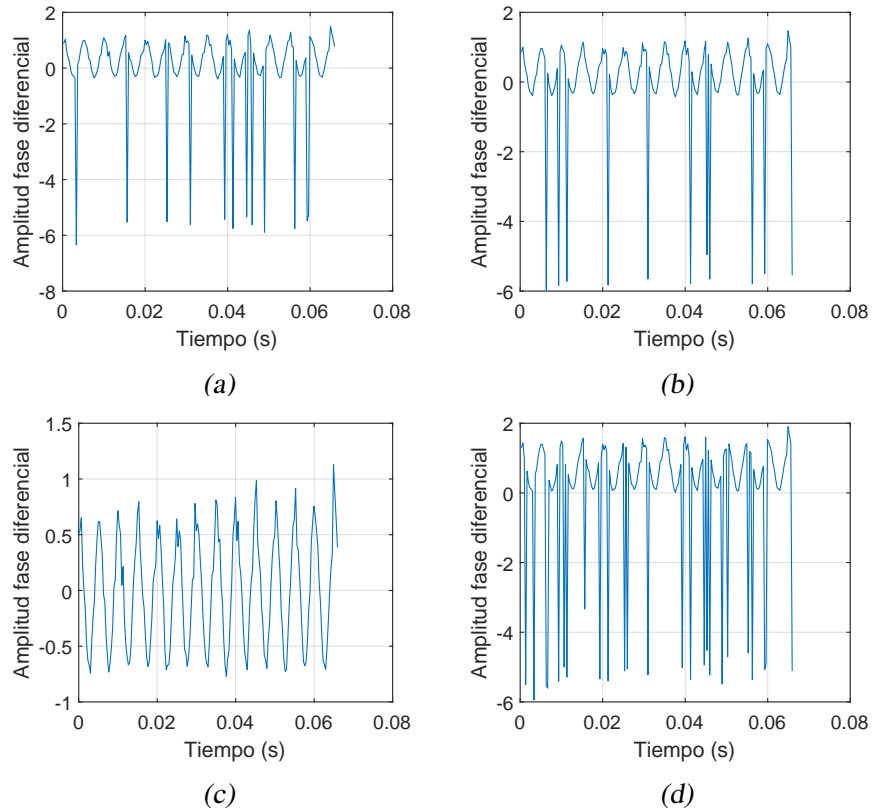


Fig. 4.40: Resultados de desenvolvimiento con perturbación de 20Hz y 10 % dropout. a) MATLAB, b) TIE, c) UKF, d) LSTM.

MATLAB y TIE obtienen resultados similares, con tres saltos de fase notorios cada uno, mientras que UKF visualmente recrea la perturbación de buena forma. El método de LSTM presenta muchos errores.

## 4. RESULTADOS

---



*Fig. 4.41:* Resultados de desenvolvimiento con perturbación de 20Hz y 30% dropout. a) MATLAB, b) TIE, c) UKF, d) LSTM.

Nuevamente MATLAB y TIE presentan resultados similares, mientras que UKF sigue sin presentar grandes desviaciones de la señal original. LSTM sigue presentando un error considerable.

## 4. RESULTADOS

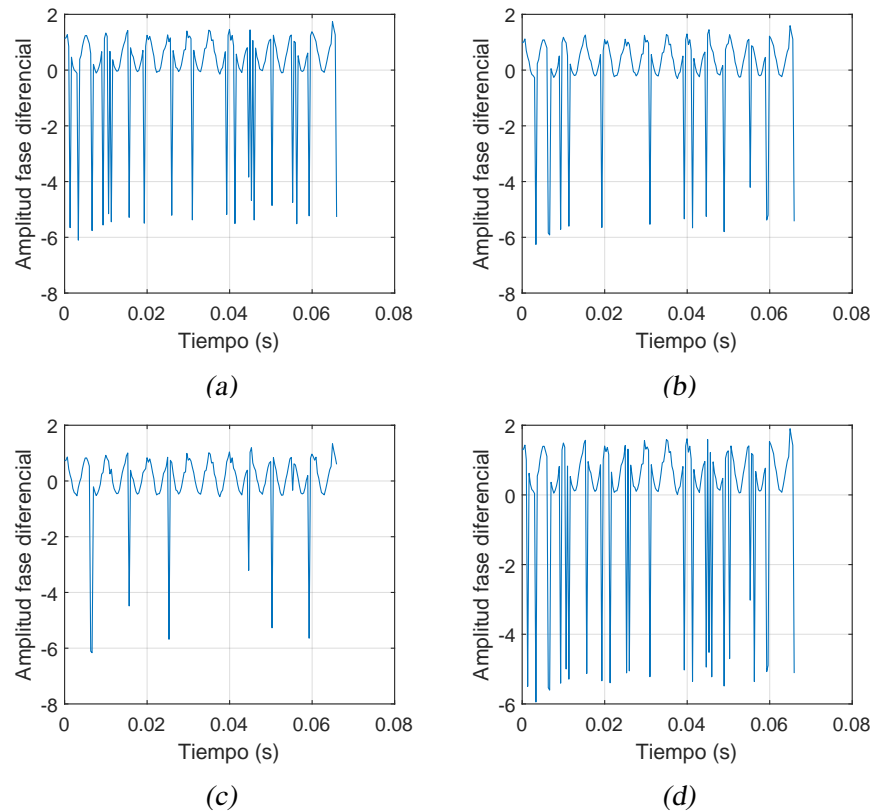


Fig. 4.42: Resultados de desenvolvimiento con perturbación de 20Hz y 50% dropout. a) MATLAB, b) TIE, c) UKF, d) LSTM.

En cuanto a MATLAB y TIE se obtienen resultados similares, reconstruyendo la señal de manera adecuada para 10% y 30% de dropout. UKF sobresale nuevamente, reconstruyendo mejor que los demás métodos la forma de onda original, y presentando pocos desvíos en la fase. Para LSTM se destaca también la consistencia de los resultados para los distintos niveles de dropout, manteniendo en cuenta la diferencia de calidad en el desenvolvimiento original.

Para comparar de manera más cuantitativa los resultados, se calcula el RMSE de cada señal respecto a la señal desenvuelta original. Los resultados se muestran en la tabla 4.1.

Método	MATLAB		TIE		UKF		LSTM	
	20 Hz	200 Hz	20 Hz	200 Hz	20 Hz	200 Hz	20 Hz	200 Hz
10 %	0,99	0,77	1,49	0,77	0,0051	0,0014	1,18	0,85
30 %	1,77	1,50	1,93	1,44	0,55	0,0039	1,19	0,84
50 %	2,23	1,87	2,31	1,65	1,24	1,08	1,17	0,81

Tab. 4.1: Error cuadrático medio (en radianes) para cada método y tasa de dropout.

Como se puede ver, el método UKF es el que mejor se desempeña en todos los niveles, con errores menores por ordenes de magnitud en los casos de 10% y 30% de dropout. Los demás

métodos mantienen niveles de error comparables entre ellos, aunque en general MATLAB supera levemente el rendimiento de TIE. El rendimiento del método LSTM muestra alto nivel de consistencia, con un margen de error que es elevado comparado a los demás para niveles menores de dropout, pero en niveles mayores resulta comparable o mejor que los demás métodos.

## 4.5 Rendimiento computacional

Otro aspecto importante al comparar métodos de procesamiento de datos es el costo computacional que implican. En este trabajo, el análisis se realiza de manera *offline*, lo que permite aplicar distintos enfoques sin una restricción inmediata de tiempo. Sin embargo, en aplicaciones reales de DAS puede ser necesario alcanzar procesamiento en tiempo real, especialmente cuando se requiere una respuesta rápida frente a eventos detectados. Por ello, aunque aquí el desempeño en tiempo no es prioritario, resulta relevante considerar la velocidad de los distintos métodos como referencia tanto para el análisis *offline* como para futuros escenarios en los que el poder de cómputo y el esquema de detección permitan implementar procesamiento en línea.

En las tareas de desenvolvimiento realizadas anteriormente, se usaron las funciones incluidas en los distintos lenguajes de programación para medir el tiempo de ejecución de cada función. Los tiempos que se usarán para comparar consisten en el tiempo total de desenvolvimiento de fase para la matriz de datos asociada a las señales de 20 Hz, puesto que son las matrices más grandes (aproximadamente  $400 \times 250000$ ), y por lo tanto corresponde a un caso más representativa y consistente del proceso. En la siguiente tabla se muestran los tiempos promedios de cómputo para cada método, donde el promedio se calcula considerando el desenvolvimiento en sentido temporal y espacial.

Método	Tiempo de cómputo medio (s)
unwrap()	10.25
TIE	174.02 (*)
UKF	2093.71
LSTM	921.99

Tab. 4.2: Tiempo medio de cómputo de fase desvuelta para los distintos métodos. (\*: Para TIE solo se consideran el cómputo en sentido espacial)

El método de MATLAB es el más rápido por un orden de magnitud comparado con el segundo lugar, el método TIE. Le sigue LSTM, y finalmente UKF, el cual es el método más lento por un margen importante. Es por esto que el uso práctico de un método como LSTM o UKF (con la implementación en este trabajo) debe ser enfocado a procesamiento *offline*.

## 5 Conclusiones

En este trabajo se implementaron y probaron una serie de métodos de desenvolvimiento de fase con datos reales obtenidos de un esquema heterodino  $\Phi$ -OTDR, obteniendo resultados positivos en términos de aplicabilidad de métodos que originalmente no estaban pensados para ser utilizados en datos DAS.

En general, todos los métodos logran fases desenvueltas utilizables en para mediciones obtenidas en condiciones favorables. La función `unwrap` de MATLAB destaca obteniendo resultados regulares en las distintas pruebas, a pesar de no tener pasos adicionales para añadir robustez ante errores de medición. El método TIE obtiene resultados similares a `unwrap`, pero mostrando mayor robustez en algunos casos, como el de dropout. El filtro de Kalman Unscented destaca de manera constante como el método con mayor tolerancia en condiciones desfavorables, resaltando su robustez ante casos de datos perdidos, reconstruyendo las señales de manera excelente.

La aplicación del desenvolvimiento en sentido temporal fue completamente desfavorable, empeorando la calidad de la fase al utilizarlo en conjunto con el desenvolvimiento espacial, e incapaz de generar fases válidas al ser utilizado por si solo.

El uso de los diferentes métodos que se probaron tiene asociado un costo computacional que no deja de ser relevante. El método de MATLAB presenta un costo computacional muy bajo, al tratarse de una serie de comparaciones, mientras que los demás métodos aumentan en complejidad para aumentar la robustez. TIE y UKF tienen un costo computacional alto al momento de ejecutar, y el LSTM a la hora de ser entrenado.

De manera práctica, el método default de MATLAB resulta adecuado para casos generales, donde las condiciones no son desfavorables y el poder de cómputo es más limitado. Por otro lado UKF resulta el claro ganador a la hora de tratarse de escenarios más desfavorables en calidad de datos y mediciones, a cambio de un mayor tiempo de cómputo. El método TIE termina al medio de ambos, mostrando robustez regular y una buena velocidad de cómputo.

La implementación de enfoques basados en machine learning requiere un trabajo más profundo en la generación de bases de datos representativas y de mayor tamaño para el caso DAS. Los resultados obtenidos que se muestran en este trabajo sugieren un alto potencial de dichos métodos, siempre que se optimicen los procesos de entrenamiento, lo que podría traducirse en resultados

## 5. CONCLUSIONES

---

más robustos.

Finalmente, avanzar hacia la implementación de estos métodos en sistemas capaces de operar en tiempo real abre un campo de gran interés. Esto implica no solo optimizar los algoritmos para reducir su costo computacional y hacerlos viables en, por ejemplo, sistemas embebidos, sino también integrarlos con el procesamiento necesario para obtener fases diferenciales y, a partir de ellas, las perturbaciones que afectan la fibra en aplicaciones DAS de manera mucho más eficiente.

# Bibliografía

- [1] Mark Jenkinson. “Fast, automated, N-dimensional phase-unwrapping algorithm”. En: *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine* 49.1 (2003), págs. 193-197.
- [2] Hanwen Yu et al. “Phase Unwrapping in InSAR : A Review”. En: *IEEE Geoscience and Remote Sensing Magazine* 7.1 (2019), págs. 40-58. DOI: 10 . 1109 / MGRS . 2018 . 2873644.
- [3] The MathWorks Inc. *MATLAB Documentation*. Natick, Massachusetts, United States, 2025. URL: <https://www.mathworks.com/help/matlab/ref/unwrap.html>.
- [4] G. E. Spoorthi, Subrahmanyam Gorthi y Rama Krishna Sai Subrahmanyam Gorthi. “PhaseNet: A Deep Convolutional Neural Network for Two-Dimensional Phase Unwrapping”. En: *IEEE Signal Processing Letters* 26.1 (2019), págs. 54-58. DOI: 10 . 1109 / LSP . 2018 . 2879184.
- [5] Arthur H Hartog. *An introduction to distributed optical fibre sensors*. CRC press, 2017.
- [6] Juan Pastor Graells. “Chirped-pulse phase-sensitive optical time domain reflectometry”. Tesis doct. Universidad de Alcalá, 2018.
- [7] Xiaoyi Bao y Yuan Wang. “Recent advancements in Rayleigh scattering-based distributed fiber sensors”. En: *Advanced devices & instrumentation* (2021).
- [8] Koh-Ichi Aoyama, Kiyoshi Nakagawa y Takeshi Itoh. “Optical time domain reflectometry in a single-mode fiber”. En: *IEEE Journal of Quantum Electronics* 17.6 (1981), págs. 862-868.
- [9] M\_K Barnoski et al. “Optical time domain reflectometer”. En: *Applied optics* 16.9 (1977), págs. 2375-2379.
- [10] Quan Chai et al. “Review on fiber-optic sensing in health monitoring of power grids”. En: *Optical Engineering* 58.7 (2019), págs. 072007-072007.

- [11] Li Zhang. *Distributed optical fiber sensing based on frequency-scanned phase-sensitive optical time-domain reflectometry*. Inf. téc. EPFL, 2020.
- [12] John G Proakis. *Digital Signal Processing*. Pearson, 2007.
- [13] Yuanyuan Shan et al. “A Broadband Distributed Vibration Sensing System Assisted by a Distributed Feedback Interferometer”. En: *IEEE Photonics Journal* 10.1 (2018), págs. 1-10. DOI: 10.1109/JPHOT.2017.2776919.
- [14] Yuelan Lu et al. “Distributed vibration sensor based on coherent detection of phase-OTDR”. En: *Journal of lightwave Technology* 28.22 (2010), págs. 3243-3249.
- [15] Fei Peng et al. “Ultra-long high-sensitivity  $\Phi$ -OTDR for high spatial resolution intrusion detection of pipelines”. En: *Opt. Express* 22.11 (jun. de 2014), págs. 13804-13810. DOI: 10.1364/OE.22.013804. URL: <https://opg.optica.org/oe/abstract.cfm?URI=oe-22-11-13804>.
- [16] Zinan Wang et al. “Coherent  $\Phi$ -OTDR based on I/Q demodulation and homodyne detection”. En: *Optics express* 24.2 (2016), págs. 853-858.
- [17] Shuaiqi Liu et al. “Advances in phase-sensitive optical time-domain reflectometry”. En: *Opto-Electronic Advances* 5.3 (2022), págs. 200078-1.
- [18] Govind P Agrawal. “Nonlinear fiber optics”. En: *Nonlinear Science at the Dawn of the 21st Century*. Springer, 2000, págs. 195-211.
- [19] Yun-Jiang Rao et al. “Long-distance fiber-optic  $\Phi$ -OTDR intrusion sensing system”. En: *20th international conference on optical fibre sensors*. Vol. 7503. SPIE. 2009, págs. 250-253.
- [20] Hugo Fidalgo Martins et al. “Comparison of the use of first and second-order Raman amplification to assist a phase-sensitive optical time domain reflectometer in distributed vibration sensing over 125 km”. En: *23rd International Conference on Optical Fibre Sensors*. Vol. 9157. SPIE. 2014, págs. 933-936.
- [21] ZN Wang et al. “Phase-sensitive optical time-domain reflectometry with Brillouin amplification”. En: *Optics letters* 39.15 (2014), págs. 4313-4316.
- [22] ZN Wang et al. “Ultra-long phase-sensitive OTDR with hybrid distributed amplification”. En: *Optics letters* 39.20 (2014), págs. 5866-5869.
- [23] Muping Song et al. “151-km single-end phase-sensitive optical time-domain reflectometer assisted by optical repeater”. En: *Optical Engineering* 57.2 (2018), págs. 027104-027104.
- [24] Faruk Uyar et al. “94.8 km-range direct detection fiber optic distributed acoustic sensor”. En: *2019 Conference on Lasers and Electro-Optics (CLEO)*. IEEE. 2019, págs. 1-2.

- [25] Qian Zhang et al. “Analytical model on the sensitivity in heterodyne-detection phase-sensitive optical time-domain reflectometry”. En: *Optical Fiber Sensors*. Optica Publishing Group. 2023, W4-29.
- [26] Govind P Agrawal. *Fiber-optic communication systems*. John Wiley & Sons, 2012.
- [27] Meiqi Ren et al. “Influence of finite extinction ratio on performance of phase-sensitive optical time-domain reflectometry”. En: *Optics express* 24.12 (2016), págs. 13325-13333.
- [28] R. Paschotta. *Acousto-optic Modulators*. RP Photonics Encyclopedia. URL: [https://www.rp-photonics.com/acousto%5C\\_optic%5C\\_modulators.html](https://www.rp-photonics.com/acousto%5C_optic%5C_modulators.html).
- [29] INPHENIX. *Semiconductor optical amplifier (soa)*. URL: <https://www.inphenix.com/en/semiconductor-optical-amplifiers/>.
- [30] AeroDIODE. *Semiconductor optical amplifier (soa)*. URL: <https://www.aerodiode.com/product/semiconductor-optical-amplifier/>.
- [31] Guangyao Yang et al. “Long-range distributed vibration sensing based on phase extraction from phase-sensitive OTDR”. En: *IEEE Photonics Journal* 8.3 (2016), págs. 1-12.
- [32] Haijun He et al. “SNR enhancement in phase-sensitive OTDR with adaptive 2-D bilateral filtering algorithm”. En: *IEEE Photonics journal* 9.3 (2017), págs. 1-10.
- [33] Zengguang Qin, Hui Chen y Jun Chang. “Detection performance improvement of distributed vibration sensor based on curvelet denoising method”. En: *Sensors* 17.6 (2017), pág. 1380.
- [34] Zengguang Qin, Hui Chen y Jun Chang. “Signal-to-noise ratio enhancement based on empirical mode decomposition in phase-sensitive optical time domain reflectometry systems”. En: *Sensors* 17.8 (2017), pág. 1870.
- [35] P Healey. “Fading in heterodyne OTDR”. En: *Electronics letters* 20.1 (1984), págs. 30-32.
- [36] Zhengqing Pan et al. “Interference-fading-free phase-demodulated OTDR system”. En: *OFS2012 22nd International Conference on Optical Fiber Sensors*. Vol. 8421. SPIE. 2012, págs. 418-421.
- [37] Jingdong Zhang et al. “80 km fading free phase-sensitive reflectometry based on multi-carrier NLFM pulse without distributed amplification”. En: *Journal of Lightwave Technology* 37.18 (2019), págs. 4748-4754.
- [38] Kexin Cui et al. “Interference-Fading-Suppressed Pulse-Coding Phi-OTDR Using Spectrum Extraction and Rotated-Vector-Sum Method”. En: *IEEE Photonics Journal* 13.6 (2021), págs. 1-6. DOI: 10.1109/JPHOT.2021.3121064.

- [39] Yue Wu et al. “Interference fading elimination with single rectangular pulse in-OTDR”. En: *Journal of Lightwave Technology* 37.13 (2019), págs. 3381-3387.
- [40] Dian Chen, Qingwen Liu y Zuyuan He. “Phase-detection distributed fiber-optic vibration sensor without fading-noise based on time-gated digital OFDR”. En: *Optics express* 25.7 (2017), págs. 8315-8325.
- [41] Qian Zhang et al. “Modelling the Demodulated Phase Precision of Standard Heterodyne-detection Phase-sensitive Optical Time-domain Reflectometry”. En: *Journal of Lightwave Technology* (2024), págs. 1-10. DOI: 10.1109/JLT.2024.3486818.
- [42] Kazuyoshi Itoh. “Analysis of the phase unwrapping algorithm”. En: *Applied optics* 21.14 (1982), págs. 2470-2470.
- [43] J. M. Huntley y H. Saldner. “Temporal phase-unwrapping algorithm for automated interferogram analysis”. En: *Appl. Opt.* 32.17 (jun. de 1993), págs. 3047-3052. DOI: 10.1364/AO.32.003047. URL: <https://opg.optica.org/ao/abstract.cfm?URI=ao-32-17-3047>.
- [44] H. O. Saldner y J. M. Huntley. “Temporal phase unwrapping: application to surface profiling of discontinuous objects”. En: *Appl. Opt.* 36.13 (mayo de 1997), págs. 2770-2775. DOI: 10.1364/AO.36.002770. URL: <https://opg.optica.org/ao/abstract.cfm?URI=ao-36-13-2770>.
- [45] O. Michailovich y D. Adam. “Phase unwrapping for 2-D blind deconvolution of ultrasound images”. En: *IEEE Transactions on Medical Imaging* 23.1 (2004), págs. 7-25. DOI: 10.1109/TMI.2003.819932.
- [46] GE Spoorthi, Rama Krishna Sai Subrahmanyam Gorthi y Subrahmanyam Gorthi. “PhaseNet 2.0: Phase unwrapping of noisy data based on deep learning approach”. En: *IEEE transactions on image processing* 29 (2020), págs. 4862-4872.
- [47] Jean-Christophe Kucwaj et al. “A model-based angle unwrapper: Application to the GNSS-R Interference Pattern Technique”. En: *2016 19th International Conference on Information Fusion (FUSION)*. 2016, págs. 2095-2100.
- [48] Dennis C Ghiglia y Louis A Romero. “Robust two-dimensional weighted and unweighted phase unwrapping that uses fast transforms and iterative methods”. En: *Journal of the Optical Society of America A* 11.1 (1994), págs. 107-117.
- [49] Zhi-Pei Liang. “A model-based method for phase unwrapping”. En: *IEEE Transactions on Medical Imaging* 15.6 (1996), págs. 893-897. DOI: 10.1109/42.544507.

- [50] Zixin Zhao et al. “Robust 2D phase unwrapping algorithm based on the transport of intensity equation”. En: *Measurement Science and Technology* 30.1 (nov. de 2018), pág. 015201. DOI: 10.1088/1361-6501/aaec5c. URL: <https://dx.doi.org/10.1088/1361-6501/aaec5c>.
- [51] Neeraj Pandey, Amitava Ghosh y Kedar Khare. “Two-dimensional phase unwrapping using the transport of intensity equation”. En: *Applied optics* 55.9 (2016), págs. 2418-2425.
- [52] Juan Martinez-Carranza, Konstantinos Falaggis y Tomasz Kozacki. “Fast and accurate phase-unwrapping algorithm based on the transport of intensity equation”. En: *Applied Optics* 56.25 (2017), págs. 7079-7088.
- [53] Zixin Zhao (2025). *Robust 2D phase unwrapping algorithm*. 2025. URL: <https://www.mathworks.com/matlabcentral/fileexchange/68493-robust-2d-phase-unwrapping-algorithm>.
- [54] Rudolph Emil Kalman. “A new approach to linear filtering and prediction problems”. En: (1960).
- [55] Gary Bishop, Greg Welch et al. “An introduction to the kalman filter”. En: *Proc of SIGGRAPH, Course 8.27599-23175* (2001), pág. 41.
- [56] Masoud Khodarahmi y Vafa Maihami. “A review on Kalman filter models”. En: *Archives of Computational Methods in Engineering* 30.1 (2023), págs. 727-747.
- [57] Qiang Li et al. “Kalman filter and its application”. En: *2015 8th international conference on intelligent networks and intelligent systems (ICINIS)*. IEEE. 2015, págs. 74-77.
- [58] Eric A Wan y Rudolph Van Der Merwe. “The unscented Kalman filter”. En: *Kalman filtering and neural networks* (2001), págs. 221-280.
- [59] E.A. Wan y R. Van Der Merwe. “The unscented Kalman filter for nonlinear estimation”. En: *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373)*. 2000, págs. 153-158. DOI: 10.1109/ASSPCC.2000.882463.
- [60] Simon J Julier y Jeffrey K Uhlmann. “New extension of the Kalman filter to nonlinear systems”. En: *Signal processing, sensor fusion, and target recognition VI*. Vol. 3068. Spie. 1997, págs. 182-193.
- [61] Jeffrey K Uhlmann. *Simultaneous map building and localization for real time applications*. Inf. téc. Technical report, University of Oxford, 1994. Transfer thesis, 1994.
- [62] Rudolph Van Der Merwe. *Sigma-point Kalman filters for probabilistic inference in dynamic state-space models*. Oregon Health & Science University, 2004.

- [63] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.
- [64] Ian Goodfellow et al. *Deep learning*. Vol. 1. 2. MIT press Cambridge, 2016.
- [65] Kaiqiang Wang et al. “One-step robust deep learning phase unwrapping”. En: *Optics express* 27.10 (2019), págs. 15100-15115.
- [66] Olaf Ronneberger, Philipp Fischer y Thomas Brox. “U-net: Convolutional networks for biomedical image segmentation”. En: *Medical image computing and computer-assisted intervention—MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III* 18. Springer. 2015, págs. 234-241.
- [67] Nahian Siddique et al. “U-net and its variants for medical image segmentation: A review of theory and applications”. En: *IEEE access* 9 (2021), págs. 82031-82057.
- [68] Zongwei Zhou et al. “Unet++: A nested u-net architecture for medical image segmentation”. En: *Deep learning in medical image analysis and multimodal learning for clinical decision support: 4th international workshop, DLMIA 2018, and 8th international workshop, ML-CDS 2018, held in conjunction with MICCAI 2018, Granada, Spain, September 20, 2018, proceedings 4*. Springer. 2018, págs. 3-11.
- [69] Ozan Oktay et al. “Attention u-net: Learning where to look for the pancreas”. En: *arXiv preprint arXiv:1804.03999* (2018).
- [70] Reza Azad et al. “Medical image segmentation review: The success of u-net”. En: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2024).
- [71] Sepp Hochreiter y Jürgen Schmidhuber. “Long short-term memory”. En: *Neural computation* 9.8 (1997), págs. 1735-1780.
- [72] Alex Graves y Alex Graves. “Long short-term memory”. En: *Supervised sequence labeling with recurrent neural networks* (2012), págs. 37-45.
- [73] Greg Van Houdt, Carlos Mosquera y Gonzalo Nápoles. “A review on the long short-term memory model”. En: *Artificial Intelligence Review* 53.8 (2020), págs. 5929-5955.
- [74] Thomas Fischer y Christopher Krauss. “Deep learning with long short-term memory networks for financial market predictions”. En: *European Journal of Operational Research* 270.2 (2018), págs. 654-669. ISSN: 0377-2217. DOI: <https://doi.org/10.1016/j.ejor.2017.11.054>. URL: <https://www.sciencedirect.com/science/article/pii/S0377221717310652>.
- [75] Kaisheng Yao et al. “Spoken language understanding using long short-term memory neural networks”. En: *2014 IEEE spoken language technology workshop (SLT)*. IEEE. 2014, págs. 189-194.

- [76] Alex Graves, Abdel-rahman Mohamed y Geoffrey Hinton. “Speech recognition with deep recurrent neural networks”. En: *2013 IEEE international conference on acoustics, speech and signal processing*. Ieee. 2013, págs. 6645-6649.
- [77] Lei Kong et al. “Robust one-dimensional phase unwrapping algorithm based on lstm network with reduced parameter number”. En: *Journal of Lightwave Technology* 40.19 (2022), págs. 6560-6567.
- [78] Chetan Mishra, Luigi Vanfretti y Kevin D Jones. “Synchrophasor phase angle data unwrapping using an unscented Kalman filter”. En: *IEEE Transactions on Power Systems* 36.5 (2021), págs. 4868-4871.
- [79] Muhammad Usama Usman y M Omar Faruque. “Applications of synchrophasor technologies in power systems”. En: *Journal of Modern Power Systems and Clean Energy* 7.2 (2019), págs. 211-226.
- [80] Kaiqiang Wang. *Phase unwrapping by U-net*. [https://github.com/kqwang/Phase\\_unwrapping\\_by\\_U-Net](https://github.com/kqwang/Phase_unwrapping_by_U-Net). 2022.
- [81] Shuai Wang, Bo Chen y Jing Wang. “Phase unwrapping with the convolutional neural network”. En: *Holography, Diffractive Optics, and Applications X*. Vol. 11551. SPIE. 2020, págs. 239-244.
- [82] Malsha V. Perera y Ashwin De Silva. *A Joint Convolutional and Spatial Quad-Directional LSTM Network for Phase Unwrapping*. 2020. arXiv: 2010.13268 [cs.LG]. URL: <https://arxiv.org/abs/2010.13268>.
- [83] Keiron O’shea y Ryan Nash. “An introduction to convolutional neural networks”. En: *arXiv preprint arXiv:1511.08458* (2015).
- [84] David L Donoho y Jared Tanner. “Precise undersampling theorems”. En: *Proceedings of the IEEE* 98.6 (2010), págs. 913-924.
- [85] Mathieu Lepot, Jean-Baptiste Aubin y François HLR Clemens. “Interpolation in time series: An introductory overview of existing methods, their performance criteria and uncertainty assessment”. En: *Water* 9.10 (2017), pág. 796.
- [86] MN Noor et al. “Filling missing data using interpolation methods: Study on the effect of fitting distribution”. En: *Key Engineering Materials* 594 (2014), págs. 889-895.