



Instituto **Tecnológico**[®]
de Aguascalientes

Instituto Tecnológico de Aguascalientes

Desarrollo de un espectrómetro de absorción VIS para la clasificación de materiales translúcidos

TESIS

Para obtener el grado de Maestro en Ciencias de la Ingeniería

PRESENTA: Erick Hernández Torres

Director de tesis: Dr. Julio Cesar Martínez Romo

Co-director de tesis: Dr. Francisco Javier Luna Rosas

Aguascalientes, Ags., Agosto del 2025



Educación
Secretaría de Educación Pública



TECNOLÓGICO
NACIONAL DE MÉXICO



Instituto Tecnológico de Aguascalientes
División de Estudios de Posgrado e Investigación

Aguascalientes, Ags., **5/septiembre/2025**
Oficio No. DEPI 594/2025
Asunto: Autorización impresión tesis

C. ERICK HERNÁNDEZ TORRES
PRESENTE

Los abajo firmantes, Miembros del Jurado para Examen de Grado de Maestría, hacemos **constar** que, habiendo revisado el Trabajo de Tesis desarrollado por Usted, bajo el título: **"Desarrollo de un espectrómetro de absorción VIS para la clasificación de materiales translúcidos"**, hemos dictaminado que éste es de aceptarse, por lo que se autoriza su impresión.

ATENTAMENTE

Excelencia en Educación Tecnológica®
Ingenio, Cultura y Saber, que conducen a la Excelencia®

PRESIDENTE


DR. JULIO CÉSAR MARTÍNEZ ROMO

SECRETARIO


DR. FRANCISCO JAVIER LUNA ROSAS

VOCAL


DR. MARCO ANTONIO HERNÁNDEZ VARGAS

ccp. Archivo
JFLH/yevo



2025
Año de
La Mujer
Indígena

Av. Adolfo López Mateos #1801. Fracc. Bona Gens
C.P. 20256 Aguascalientes, Ags. Tel. 449 910 5002 ext. 127
e-mail: jefatura.depi@aguascalientes.tecnm.mx
tecnm.mx | www.aguascalientes.tecnm.mx



AUTORIZACIÓN DE USO DE DERECHOS DE AUTOR OTORGADO POR

Erick Hernández Torres, mayor de edad, con domicilio ubicado en Diego Rivera #240, Lomas de Santa Anita, en mi calidad de titular y autor de la tesis denominada **Desarrollo de un espectrómetro de absorción VIS para la clasificación de materiales translúcidos** quien para todos los fines del presente documento se denominará **EL AUTOR Y/O TITULAR**, suscribo el presente documento de autorización de uso de derechos patrimoniales de autor a favor del Tecnológico Nacional de México/Instituto Tecnológico de Aguascalientes el cual se registrá por las clausulas siguientes:

PRIMERA – AUTORIZACIÓN: **EL AUTOR Y/O TITULAR**, mediante el presente documento autoriza la utilización de los derechos patrimoniales de autor al Tecnológico Nacional de México/Instituto Tecnológico de Aguascalientes, de la tesis denominada **Desarrollo de un espectrómetro de absorción VIS para la clasificación de materiales translúcidos**, a través del Repositorio Institucional del Tecnológico Nacional de México (en lo sucesivo TecNM) y en el Repositorio Nacional, que puede ser consultado en la liga electrónica: (www.repositorionacionalcti.mx/).

SEGUNDA – OBJETO: Por medio del presente escrito, **EL AUTOR Y/O TITULAR** autoriza al Tecnológico Nacional de México/Instituto Tecnológico de Aguascalientes, a través del Repositorio Institucional del TecNM y en el Repositorio Nacional para que de conformidad con la Ley Federal del Derecho de Autor y la Ley de la Propiedad Industrial, use los derechos del documento antes referido, con fines exclusivamente académicos.

TERCERA – TERRITORIO: Los derechos aquí autorizados se dan sin limitación geográfica o territorial alguna.

CUARTA – ALCANCE: La presente autorización se da tanto para formato o soporte material, y se extiende a la utilización en medio óptico, magnético, electrónico, en red, mensajes de datos o similar conocido o por conocer, del ejemplar o número respectivo de la publicación.

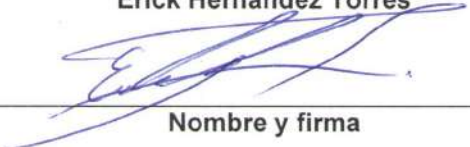
QUINTA – EXCLUSIVIDAD: La autorización de uso aquí establecida no implica exclusividad en favor del Tecnológico Nacional de México/Instituto Tecnológico de Aguascalientes. Por lo tanto, **EL AUTOR Y/O TITULAR** en su carácter de autor de la obra objeto del presente documento se reserva el derecho de publicar directamente, u otorgar a cualquier tercero, autorizaciones de uso similares o en los mismos términos aquí acordados.

SEXTA – DERECHOS MORALES (Créditos y mención): La Autorización de los derechos antes mencionados no implica la cesión de los derechos morales sobre los mismos por cuanto en conformidad con lo establecido en los artículos 18, 19, 20, 21, 22 y 23 de la Ley Federal de Derechos de Autor, dada la cuenta que estos derechos son inalienables, imprescriptibles, irrenunciables e inembargables. Por lo tanto, los mencionados derechos seguirán radicados en cabeza de **EL AUTOR Y/O TITULAR**, y siempre deberá mencionarse su nombre cuando se utilice la obra.

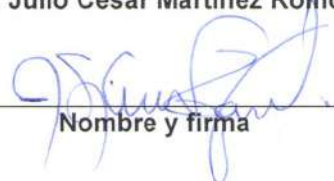
SÉPTIMA – AUTORÍA: **EL AUTOR Y/O TITULAR**, declara y ratifica que el material objeto de la presente fue realizada por él o por ella sin violar o usurpar derechos de Propiedad Intelectual de terceros.

Aguascalientes, Aguascalientes, a 08 días del mes de septiembre de 2025.

Autor de la Tesis
Erick Hernández Torres


Nombre y firma

Director de Tesis
Dr. Julio César Martínez Romo


Nombre y firma



Educación
Secretaría de Educación Pública



TECNOLÓGICO
NACIONAL DE MÉXICO



Instituto Tecnológico de Aguascalientes
División de Estudios de Posgrado e Investigación

Aguascalientes, Ags., **21/octubre/2025**
Asunto: Constancia de Originalidad

**CONSEJO DE POSGRADO
MAESTRÍA EN CIENCIAS DE LA INGENIERÍA
PRESENTE**

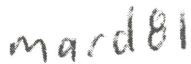
El director de tesis de **Erick Hernández Torres**, estudiante de **Maestría en Ciencias de la Ingeniería** con número de control **G23153208**, hace constar que, después de haber realizado la revisión del informe de originalidad generado con la herramienta Turnitin, el trabajo **"Desarrollo de un espectrómetro de absorción VIS para la clasificación de materiales translúcidos"** es original conforme a los criterios establecidos. El porcentaje de similaridad es de **18%** y se adjunta resumen del reporte.

ATENTAMENTE

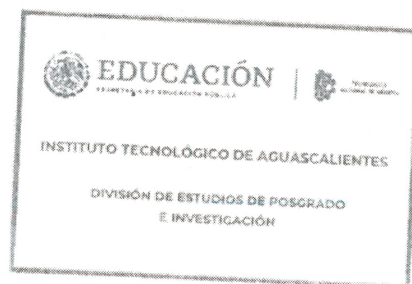
*Excelencia en Educación Tecnológica®
Ingenio, Cultura y Saber, que conducen a la Excelencia*

Vo.Bo.


DR. JULIO CÉSAR MARTÍNEZ ROMO
DIRECTOR DE TESIS


DR. MARIO ALBERTO RODRÍGUEZ DÍAZ
COORDINADOR DE LA MAESTRÍA EN CIENCIAS
DE LA INGENIERÍA

ccp. Archivo



2025
Año de
La Mujer
Indígena

Av. Adolfo López Mateos #1801. Fracc. Bona Gens
C.P. 20256 Aguascalientes, Ags. Tel. 449 910 5002 ext. 127
e-mail: jefatura.depi@aguascalientes.tecnm.mx
tecnm.mx | www.aguascalientes.tecnm.mx



Tesis_MCI_Erick Hernández Torres.pdf

-  Erick Hernández Torres
-  Erick Hernández Torres
-  Tecnológico Nacional de Mexico

Detalles del documento

Identificador de la entrega

trn:oid:::20755:516134062

Fecha de entrega

21 oct 2025, 12:57 p.m. GMT-6

Fecha de descarga

21 oct 2025, 1:09 p.m. GMT-6

Nombre del archivo

Tesis_MCI_Erick Hernández Torres.pdf

Tamaño del archivo

9.1 MB

125 páginas

31.391 palabras

162.501 caracteres

18% Similitud general

El total combinado de todas las coincidencias, incluidas las fuentes superpuestas, para ca...

Filtrado desde el informe

- Bibliografía
- Texto citado
- Texto mencionado
- Coincidencias menores (menos de 8 palabras)

Fuentes principales

- 16%  Fuentes de Internet
- 7%  Publicaciones
- 11%  Trabajos entregados (trabajos del estudiante)

Marcas de integridad

N.º de alerta de integridad para revisión

-  **Caracteres reemplazados**
51 caracteres sospechosos en N.º de páginas
Las letras son intercambiadas por caracteres similares de otro alfabeto.

Los algoritmos de nuestro sistema analizan un documento en profundidad para buscar inconsistencias que permitirían distinguirlo de una entrega normal. Si advertimos algo extraño, lo marcamos como una alerta para que pueda revisarlo.

Una marca de alerta no es necesariamente un indicador de problemas. Sin embargo, recomendamos que preste atención y la revise.

Agradecimientos

Este trabajo no hubiera sido posible gracias al apoyo de muchas personas.

Agradezco al doctor Julio Cesar Martínez Romo por orientarme y encaminarme durante todo el tiempo de realización de este trabajo, además por resolver mis dudas y aportar propuestas e información de forma continua, sin lo cual este trabajo no hubiera podido realizarse. De igual forma agradezco a todos mis profesores que son sus materias me dotaron del conocimiento y de las herramientas necesarias para llevar a cabo este trabajo. Agradezco también a todo el personal del Tecnológico Nacional de México campus Aguascalientes que me apoyaron con todos los trámites necesarios para mi estancia en el programa de Maestría en ciencias de la ingeniería.

Agradezco a todo el pueblo de México, quién a través de SECIHTI, y por medio de su programa de becas, puede recibir apoyo económico para concluir mis estudios de maestría.

Agradezco enormemente a mis padres, Rubén y Gaby, por apoyarme en continuar con mis estudios, y de igual forma agradezco a mis hermanos, Rubén y Aléxis por siempre apoyarme durante este proceso y ayudarme en todo lo que les era posible.

Resumen

Este trabajo de tesis se centra en el desarrollo de un espectrómetro de bajo costo que opere en el rango visible del espectro electromagnético, con el objetivo de implementar técnicas de filtrado de ruido durante la integración del espectro, y no después como se hace comúnmente. La investigación surge a partir de la necesidad de presentar una alternativa a los espectrómetros profesionales, los cuales suelen ser costosos, voluminosos y complejos, y a la manera en que se elimina el ruido de los espectros, lo cual se hace posterior a su integración total.

El estudio considera las bases teóricas de técnicas espectroscópicas como la UV-Vis y la Raman, ambas fundamentales para el análisis de materiales, y plantea el desarrollo de un sistema completo (hardware y software) capaz de capturar espectros de absorción en materiales translúcidos. Además se evalúa la efectividad del filtrado durante la integración del espectro, comparándola con el filtrado posterior a la integración, obteniendo la relación señal-ruido.

El objetivo final no es solo comprobar la viabilidad del sistema propuesto, sino también sentar las bases para que pueda extenderse a otras espectroscopías, abriendo el camino a desarrollos más accesibles y portables para el análisis espectroscópico.

Índice general

1. Introducción	15
1.1. Antecedentes	15
1.2. Problemática	16
1.3. Planteamiento del problema	16
1.4. Justificación	16
1.5. Objetivos	17
1.5.1. Objetivo general	17
1.5.2. Objetivos particulares	17
1.6. Hipótesis	18
1.6.1. Hipótesis de trabajo	18
1.6.2. Hipótesis nula	18
1.7. Estado del arte	18
1.8. Conclusiones	19
2. Marco Teórico	21
2.1. Introducción	21
2.2. Radiación electromagnética	21
2.3. Polarización	22
2.4. Interacción de la luz con la materia	24
2.4.1. Reflexión	24
2.4.2. Transmisión	24
2.4.3. Absorción	25
2.4.4. Luminiscencia	25
2.5. Espectroscopía	26
2.5.1. ¿Qué es la espectroscopía?	26
2.5.2. Ruido en la espectroscopía	27
2.5.3. Reducción de ruido en espectroscopía	28
2.6. Espectroscopía Raman	30
2.6.1. Dispersión Raman	30
2.6.2. Ruido en el espectro Raman	32
2.7. Espectrómetros	32
2.7.1. Espectrómetros de emisión y absorción	32
2.7.2. Monocromador	34
2.7.3. Sistemas ópticos	35

2.7.4.	Fuentes de luz	40
2.7.5.	Espectrógrafo Czerny-Turner	41
2.7.6.	Detectores	42
2.7.7.	Espectrómetros Raman	42
2.8.	Procesamiento de señales	44
2.8.1.	¿Qué es el procesamiento de señales?	44
2.8.2.	Procesamiento digital de señales	45
2.8.3.	Convertidor Anológico Digital	46
2.8.4.	Teorema de muestreo	48
2.8.5.	Filtros digitales	48
2.8.6.	Savitzky-Golay	51
2.8.7.	Wavelets	52
2.9.	Interpolación lineal	54
2.10.	Python	55
2.11.	Conclusiones	55
3.	Desarrollo del prototipo	56
3.1.	Introducción	56
3.2.	Hardware	56
3.2.1.	Optical bench	56
3.2.2.	Detector	58
3.2.3.	Microcontrolador	61
3.2.4.	Circuito de Control	64
3.2.5.	Arreglo óptico	71
3.2.6.	Fuente de luz	73
3.2.7.	Captura de datos	75
3.3.	Software	81
3.3.1.	Envío y recepción de datos	81
3.3.2.	Visualización y almacenamiento de espectros	83
3.3.3.	Calibración del equipo	84
3.4.	Conclusiones	87
4.	Captura de espectros	88
4.1.	Introducción	88
4.2.	Espectros de muestras capturados	88
4.3.	Filtrado de ruido en espectros	93
4.3.1.	Filtrado con wavelets	94
4.3.2.	Filtrado con Savitzky-Goaly	95
4.4.	Implementación de técnicas de filtrado durante la integración del espectro	96
4.5.	Conclusiones	98

5. Interpretación de resultados	99
5.1. Introducción	99
5.2. Obtención del SNR	99
5.3. Obtención del PCA	101
5.4. Conclusiones	103
6. Comentarios Finales	105
6.1. Introducción	105
6.2. Cumplimiento de los objetivos	105
6.3. Dificultades presentadas	105
6.4. Trabajo futuro	106
6.5. Conclusión	107
A. Código fuente STM32	111
B. Código fuente interpolación lineal en python	122
C. Código fuente del receptor en Python	124
D. Código fuente PCA en python	127
E. Código fuente SNR en python	129

Índice de figuras

2.1. Amplitud del vector eléctrico de la radiación electromagnética en función del tiempo. Tomado de [1]	22
2.2. Absorción de radiación electromagnética. Tomado de [1]	22
2.3. Definiciones de luz polarizada s y p, que se definen en términos del plano de incidencia. Tomado de [2]	23
2.4. Espectroscopio Bunsen y Kirchhoff. Tomado de [2]	26
2.5. Espectro visible del Hidrógeno. Tomado de [2]	27
2.6. Espectro efecto Raman. Tomado de [1]	30
2.7. Espectro Raman Cloroformo. Tomado de [3]	31
2.8. Esquema de espectrómetro de absorción. Tomado de [2]	33
2.9. Esquema de espectrómetro de emisión	34
2.10. Rejilla de difracción	35
2.11. Comportamiento de una lente	36
2.12. Tipos de lentes	37
2.13. Bloqueadores	38
2.14. Espejos esféricos	39
2.15. Selección varios tipos de prismas	39
2.16. Rayos en una fibra óptica revestida	40
2.17. Diagrama espectrógrafo Czerny-Turner	41
2.18. Plano focal ideal espectrógrafo Czerny-Turner	41
2.19. Ejemplo de la orientación de un CCD en el plano focal de un espectrógrafo, mostrando las imágenes de la rendija de entrada para la luz incidente de tres longitudes de onda	42
2.20. Espectrómetro Raman genérico	43
2.21. Espectrómetro Raman genérico	44
2.22. Diferencia señal en tiempo continuo y en tiempo discreto	45
2.23. Proceso de digitalización de una señal	47
2.24. Respuesta dominio del tiempo y frecuencia	50
2.25. Respuesta en frecuencia filtros	51
2.26. Wavelet Coiflet	53
2.27. Algoritmo Mallat	54
3.1. Optical bench Czerny-Turner	57
3.2. Curva de eficiencia optical bench Czerny-Turner	58
3.3. Sensor Sony ILX511	59

3.4. Gráfica de sensibilidad sensor Sony ILX511	59
3.5. Sensor Hamamatsu S10122-512Q-01	60
3.6. Respuesta espectral S10122-512Q-01	60
3.7. Sensor Toshiba TCD1304DG	61
3.8. Respuesta espectral TCD1304DG	61
3.9. Características STM32F401CC6	63
3.10. Encapsulados microcontroladores STM32F401	63
3.11. BlackPill	64
3.12. Circuito de control típico TCD1304DG	65
3.13. Amplificador no inversor para la salida del sensor	66
3.14. Circuito de control implementado en protoboard	67
3.15. Esquemático primera PCB circuito de control	68
3.16. Primera PCB circuito de control terminada	69
3.17. Esquemático segunda PCB circuito de control	69
3.18. Segunda PCB circuito de control terminada montada sobre el optical bench	70
3.19. Óptico bench con circuito de control implementado	71
3.20. Cubeta para muestras	72
3.21. Lentes plano convexas para arreglo óptico	72
3.22. Espectrómetro construido	73
3.23. Lámpara halógena de tungsteno	74
3.24. Respuesta espectral lámpara halógena de tungsteno	75
3.25. Software STM32Cube IDE para la programación del microcon- trolador	75
3.26. Interface gráfica para la configuración del microcontrolador	76
3.27. Requerimientos de señales de control de sensor CCD	77
3.28. Señal oM captada por osciloscopio	78
3.29. Generación de retraso señal ICG en código de programación	78
3.30. Señal ICG y SH captadas por osciloscopio	79
3.31. Representación ideal de la señales de control y de salida del sensor CCD	79
3.32. Señales ICG y OS captados por osciloscopio	80
3.33. Función para captura de la señal OS	81
3.34. Función para envío de datos por USB	82
3.35. Configuración en Python de USB y variables para captura de espectros en el receptor	82
3.36. Recepción y conversión de los datos recibidos	83
3.37. Graficar los datos	83
3.38. Graficar y guardar los datos	84
3.39. Espectro luz blanca Led	84
3.40. Código interpolación lineal para obtener longitudes de onda	86
3.41. Interpolación lineal para obtener longitudes de onda	86
3.42. Espectro luz blanca Led con equipo calibrado	87
4.1. Espectros aceite de oliva, licor de amaranto, brandy y café soluble	89
4.2. Espectros clorofila, detergente de ropa, enjuague bucal y jamaica	90

4.3.	Espectros leche, anticongelante, líquido limpia vidrios y mertiolate	91
4.4.	Espectros naranja, pepto bismol, pinol y shampoo	91
4.5.	Espectros whisky, láser rojo y láser verde	92
4.6.	Espectros colores Led RGB	93
4.7.	Espectros selenita y fluorita	93
4.8.	Código de filtrado con transformada wavelet	94
4.9.	Espectros clorofila antes y después de ser filtrada con transformada wavelet	95
4.10.	Código de filtrado con Savitzky-Golay	95
4.11.	Espectros clorofila antes y después de ser filtrada con Savitzky-Golay	96
4.12.	Espectros de clorofila filtrados con valores de ventana de 21, 51, 71 y 91	96
4.13.	Selección del tipo de filtro a utilizar	97
4.14.	Clorofila filtrada con wavelets después y durante la integración del espectro	98
4.15.	Clorofila filtrada con Savitzky-Golay después y durante la integración del espectro	98
5.1.	Código de python para obtención del SNR	100
5.2.	PCA sin filtrar	101
5.3.	PCA Wavelets	102
5.4.	PCA Savitzky-Golay	102
5.5.	PCA global	103

Índice de tablas

5.1. Resultados SNR filtros aplicados	100
---	-----

Capítulo 1

Introducción

En este capítulo se describen los antecedentes necesarios, los cuales llevan a entender de forma coherente la problemática y el planteamiento del problema, y como estos se justifican, lo que dá a lugar a plantear los objetivos que se busca cumplir en este proyecto, y la manera en que se pretende cumplirlos, exponiendo la metodología utilizada y las conclusiones de lo que se pretende hacer.

1.1. Antecedentes

En la actualidad, el análisis de la materia mediante el uso de la espectroscopía es bastante común. Existen muchas tipos de espectroscopías, y estas se clasifican principalmente según la región del espectro electromagnético que utilizan, el tipo de interacción de la radiación con la materia y la información que proporcionan. De todos los tipos de espectroscopía que existen, la espectroscopía Raman y la espectroscopía ultravioleta-visible (UV-Vis), resultan ser unas de las mas útiles para analizar materia, debido a la información que puede obtenerse acerca de la muestra a analizar, teniendo aplicaciones forenses, en la industria farmacéuticas, análisis de semiconductores, etc.

Cuando un haz de luz blanca pasa a través de una sustancia coloreada, una porción de las longitudes de onda de este rango es absorbida, mientras que el restante es reflejado. Este es el fenómeno que origina el color de los cuerpos, y en el que se basa la espectroscopía UV-Vis, la cual se encarga de medir la atenuación del haz de luz después de su paso a través de la muestra. Esta técnica es comúnmente usada para la detección de grupos funcionales, detección de impurezas, análisis cualitativo y cuantitativo de analitos, medicamentos con grupos cromóforos entre otras aplicaciones [4].

Por su parte, la espectroscopía Raman se basa en el fenómeno analizado por el físico Chandrasekhara Venkata Raman en el año 1928, el cual plantea que al incidir un haz de luz monocromático a través de una sustancia, una fracción de la luz cambia de dirección respecto de la que provenía debido a la interacción con las moléculas de la sustancia, y mediante la recolección de esta reemisión

de luz en se puede describir la composición química de la sustancia, a esto se le conoce como efecto Raman [5].

Para poder captar estos fenómenos espectrales se requiere de un equipo especial. El desarrollo actual de la instrumentación para la espectroscopía ha permitido introducir los métodos espectrales en diversas áreas, sin embargo la implementación de dichos métodos espectrales posee un costo elevado, y los análisis cuantitativos son complejos de implementar por la falta de modelos analíticos que describan la complejidad del fenómeno espectral. El instrumento más asociado con las técnicas espectroscópicas es el espectrómetro óptico, el cual aprovecha el fenómeno de la dispersión de la luz para determinar las longitudes de onda presentes en la radiación bajo estudio [6].

1.2. Problemática

Como se mencionó anteriormente, en cualquier tipo de espectroscopía se necesita de un equipo especializado para poder detectar las longitudes de onda características generadas por la interacción de la radiación electromagnética con la materia, siendo el espectrómetro el instrumento encargado de realizar dicha tarea. Sin embargo, el disponer de un instrumento así resulta ser bastante costoso, pesado y restrictivo para la implementación de técnicas de filtrado de ruido en los espectros obtenidos, problema que también esta presente en alternativas de equipos de bajo costo y mas ligeras que las opciones profesionales, lo cual termina por complicar la interpretación de los espectros obtenidos. Cabe mencionar que existen diversos trabajos donde se emplean diferentes técnicas de filtrado para la eliminación del ruido aplicado a datos obtenidos de espectrómetro, sin embargo, esta siempre se realiza posterior a la integración total del espectro, como se puede ver en la sección 1.7.

1.3. Planteamiento del problema

Tomado en cuenta lo mencionado en la sección anterior, y con el propósito de explorar nuevas alternativas respecto a la construcción de espectrómetros y el filtrado del ruido en la espectroscopía, surgen las siguientes preguntas:

¿Es posible desarrollar el equipo con el hardware y software necesarios para la construcción un espectrómetro que trabaje en el rango de luz visible? ¿Será posible filtrar el ruido en los espectros captados por este equipo, durante la integración de los mismos, de manera que se puedan obtener espectros aceptables dentro del rango de luz visible con un equipo que no fue construido de manera profesional?

1.4. Justificación

La presencia del ruido en los espectros electromagnéticos siempre ha resultado ser un problema para la correcta interpretación de los datos obtenidos.

A pesar de que se han probado exitosamente diferentes técnicas de filtrado en los espectros, todas estas siempre son aplicadas aplicadas posterior a la integración total del espectro. El explorar nuevas formas de implementar técnicas de filtrado de ruido en los espectrómetros profesionales resulta ser una tarea complicada, debido a las restricciones de construcción del equipo, como son el conocer como está diseñado, como es que opera, tener acceso a poder realizar modificaciones en el hardware y software en caso de ser requerido, etc. También es importante mencionar conseguir un espectrómetro profesional resulta es bastante caro, sin mencionar que estos equipos son voluminosos, pesados y difíciles de transportar, características que pueden incrementarse dependiendo del tipo de espectroscopía para la cual fue diseñado, siendo los del tipo UV-Vis de absorción los más comunes y fáciles de conseguir.

Es por todo lo anterior, que a través de este trabajo, se busca desarrollar un espectrómetro que trabaje en el rango de luz visible con el hardware y algoritmos de filtrado necesarios para la eliminación del ruido de alta frecuencia durante la integración de espectros de absorción aplicado a materiales translúcidos con fines de clasificación automática.

1.5. Objetivos

1.5.1. Objetivo general

Desarrollar un espectrómetro de rango de luz visible con el hardware y algoritmos de filtrado necesarios para la eliminación del ruido de alta frecuencia durante la integración de espectros de absorción aplicado a materiales translúcidos con fines de clasificación automática.

1.5.2. Objetivos particulares

- Desarrollar el hardware y software adecuado para capturar los espectros en el rango visible.
- Implementar el equipo desarrollado para la construir un espectrómetro que trabaje en el rango visible.
- Implementar técnicas de filtrado para eliminar el ruido de los espectros durante la integración de los mismos.
- Capturar espectros para pruebas iniciales de hardware y software.
- Determinar el punto de implementación en el que las técnicas de filtrado entrega mejores resultados midiendo el nivel de ruido presente en los espectros y comparando los resultados de realizar el filtrado durante la integración del espectro y posterior a la integración total.
- Captura de base de datos con suficientes ejemplares para su clasificación.
- Implementar un clasificador básico de materiales translúcidos.

1.6. Hipótesis

1.6.1. Hipótesis de trabajo

Desarrollando el hardware y software adecuados para la construcción de un espectrómetro que trabaje en el rango de luz visible, e implementado la limpieza de los espectros durante la integración de los mismos, es posible obtener resultados similares a los obtenidos con un equipo comercial, haciendo uso de la instrumentación y del procesamiento digital de señales.

1.6.2. Hipótesis nula

Desarrollando el hardware y software adecuados para la construcción de un espectrómetro que trabaje en el rango de luz visible, e implementado la limpieza de los espectros durante la integración de los mismos, NO será posible obtener resultados similares a los obtenidos con un equipo comercial, haciendo uso de la instrumentación y del procesamiento digital de señales.

1.7. Estado del arte

El ruido en la espectroscopía siempre ha sido un tema de investigación, esto debido a los diferentes tipos de ruido presentes en los espectros, y la determinación de los investigadores de encontrar nuevas maneras de eliminarlo, con el objetivo de lograr una correcta interpretación de los datos obtenidos. Ejemplo de esto está en [7], donde mediante el uso de métodos de regresión polinomial, filtro Savitzky-Golay y mínimos cuadrados penalizados con ponderación iterativamente adaptable, se desarrolló tres prototipos computacionales que procesan los espectros para una mejor identificación de la composición, eliminando enmascaramiento provocado por el efecto fluorescencia en espectroscopía Raman.

En [8] se realizó un análisis de la composición de los asfaltenos mediante espectroscopía UV-Vis, los cuales se procesaron con filtros Savitzky-Golay y Transformada de Fourier (FFT), lo cual mejoró la calidad de las señales, incrementando su nitidez, lo que permitió la identificación de los grupos funcionales aromáticos, sus componentes y estructuras presentes.

[9] Es otro ejemplo de la eliminación del ruido en la espectroscopía, específicamente Raman. En este trabajo, mediante el uso de la transformada Wavelet, se desarrolló un algoritmo de filtrado para los espectros obtenidos por el instrumento SUPERCAM de la misión MARS 2020, donde los resultados obtenidos sugieren que la transformada Wavelet puede mejorar significativamente el aspecto de los espectros Raman, dependiendo de los umbrales elegidos, obteniendo una señal reconstruida que imita la forma de los picos de la señal original en gran medida.

En [10] se presentó una aproximación alternativa y novedosa basada en redes neuronales profundas para intentar extraer información presente en una señal ruidosa mediante el aprendizaje basado en modelos especializados en el filtrado de ruido.

En [11] se propuso un método de eliminación de ruido en el espectro Raman, basado en el principio de integración espectral seguido de la estimación de Wiener utilizando un conjunto de datos de calibración numérica, eliminando la necesidad de mediciones experimentales para la calibración, reduciendo la interacción del usuario para optimizar su rendimiento en una variedad de relaciones señal ruido.

El desarrollo de la instrumentación de un espectrómetro también ha sido un tema que ha motivado bastante al desarrollo de diversos trabajos de investigación, principalmente por buscar alternativas que entregue datos aceptables y al mismo tiempo no implique un gran costo al llevarlo a cabo. Un ejemplo de esto está en [12], en donde se propuso un diseño para un espectrofotómetro en que se utilizan materiales de bajo costo para construir todos los componentes necesarios del equipo, y de donde se destaca el uso del sensor CCD, TCD1304DG para captar los espectros y el uso de la tarjeta STM32 Núcleo F401RE, la cual implementa un microcontrolador de la familia STM32 para poder manejar el sensor.

Otro ejemplo de diseño y construcción de un espectrómetro está en [13], en donde se presentó una propuesta de diseño con la tarjeta de desarrollo Arduino UNO, ya que este permite controlar con facilidad el ángulo de inclinación de una rejilla de difracción montada sobre un servo motor, permitiendo acceder a diferentes longitudes de onda, y de la misma forma, recibir los datos de un sensor espectral, además de ser una opción flexible para adaptarse a las necesidades particulares del usuario, debido a la fácil modificación del código de programación.

En [14], se fabricó un espectrómetro Raman de doble láser, usando dos arreglos ópticos Czerny-Turner que se fusionan simétricamente y un único sensor CCD, el cual tiene dos entradas de luz, cada una de ellas concentrado la luz por separado, creando un diseño más versátil para la recopilación de datos.

En [15], se llevó a cabo una revisión de diversos proyectos de desarrollo de dispositivos de espectrofotometría que van desde prototipos académicos hasta productos industriales, utilizando distintas estrategias de implementación con el fin de conocer la factibilidad de crear dispositivos portátiles y de amplio espectro, o productos específicos para un sector en particular, pero con la intención de lograr una alta precisión y funcionalidad.

[16], se propuso un espectrómetro casero, utilizando materiales de fácil acceso y de bajo costo, que permite observar la descomposición de la luz, ayudando al análisis de la intensidad de la luz para generar diferentes espectros, utilizando 6 fuentes emisoras de luz.

1.8. Conclusiones

En este capítulo se dio un primer acercamiento hacia lo que se realizó en este trabajo, dando varios puntos clave respecto a la espectroscopía, como es que se capturan los espectros y el ruido presente en estos y los métodos de filtrado utilizado para eliminarlo, además de proporcionar un contexto teórico acerca de

la investigaciones que son realizadas respecto a estos temas en la actualidad.

Capítulo 2

Marco Teórico

2.1. Introducción

En este capítulo se desarrollan los fundamentos necesarios, para que el lector tenga un buen contexto teórico respecto a los temas abordados y trabajados en este proyecto, con el fin de entender y dar una correcta interpretación a los resultados obtenidos.

2.2. Radiación electromagnética

Toda la luz, incluida la infrarroja, se clasifica como radiación electromagnética, y consiste en campos eléctricos y magnéticos alterados, y se describe clásicamente por un movimiento continuo, similar a una onda sinusoidal, de los campos eléctrico y magnético. La figura 2.1 muestra la amplitud del campo eléctrico de la luz en función del tiempo, siendo los parámetros importantes, la longitud de onda (λ , longitud de una onda), la frecuencia ($\bar{\nu}$, número de ciclos por unidad de tiempo) y los números de onda (ν , números de onda por unidad de longitud), y estos se relacionan por la expresión 2.1

$$\bar{\nu} = \frac{v}{(c/n)} = \frac{1}{\lambda} \quad (2.1)$$

donde c es la velocidad de la luz y n el índice de refracción del medio que atraviesa. En teoría cuántica, la radiación se emite desde una fuente en unidades discretas llamadas fotones, donde la frecuencia del fotón, ν , y la energía del fotón, E_p , están relacionadas por la expresión 2.2

$$E_p = h\nu \quad (2.2)$$

donde h es la constante de Planck ($6,6256 \times 10^{-27}$ erg seg.). Los fotones de energía específica pueden ser absorbidos (o emitidos) por una molécula, lo que resulta en una transferencia de energía. En la copia de espectros de absorción,

esto resultará en el aumento de la energía de la molécula desde el suelo a un estado de excitación específico, como se ve en la figura 2.2 [1].

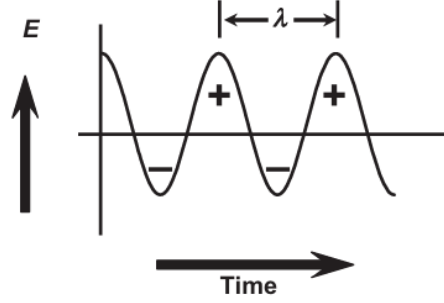


Figura 2.1: Amplitud del vector eléctrico de la radiación electromagnética en función del tiempo. Tomado de [1]

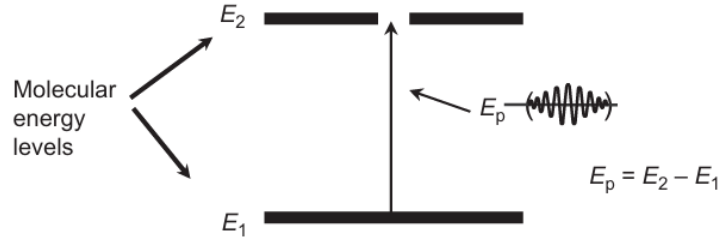


Figura 2.2: Absorción de radiación electromagnética. Tomado de [1]

Típicamente, el rotacional (E_{rot}), vibracional (E_{vib}), o electrónico (E_{el}) de la molécula es cambiado por ΔE , como se muestra en la ecuación 2.3. [1]

$$\Delta E = E_p = h\nu = hc\bar{\nu} \quad (2.3)$$

En la absorción de un fotón, la energía de la molécula aumenta y la ΔE es positiva. En una primera aproximación, las energías rotacional, vibracional y electrónica son aditivas, como se ve en la ecuación 2.4. La luz de mayor energía da lugar a transiciones electrónicas (E_{el}), y luz de menor energía da lugar a transiciones rotacionales (E_{rot}) [1].

$$E_T = E_{el} + E_{vib} + E_{rot} \quad (2.4)$$

2.3. Polarización

Al ser la luz tratada como un campo electromagnético transversal, el campo eléctrico de esta puede tomar diferentes direcciones al oscilar en un plano espe-

cífico. A esto se le conoce como polarización. Los materiales pueden absorber, reflejar o transmitir la luz de diferente manera dependiendo de su polarización. Los tipos de polarización son:

- Polarización lineal: El campo eléctrico oscila en una sola dirección.
- Polarización circular: El campo eléctrico rota en un círculo mientras la luz se propaga.
- Polarización elíptica: El campo eléctrico describe una elipse mientras la luz se propaga.

La luz polarizada aleatoriamente puede describirse en términos de polarización perpendicular a una superficie reflectante y polarización paralela a una superficie reflectante, como se ve en la figura 2.3 ambas direcciones. El plano de incidencia es el plano marcado por la trayectoria de los haces entrantes y reflejados y que es perpendicular a la superficie reflectante. Es con respecto a este plano que se definen las polarizaciones s y p. La polarización S es el componente del campo eléctrico que es perpendicular a este plano, y la polarización P es el componente del campo eléctrico que es paralelo a este plano [2] [17].

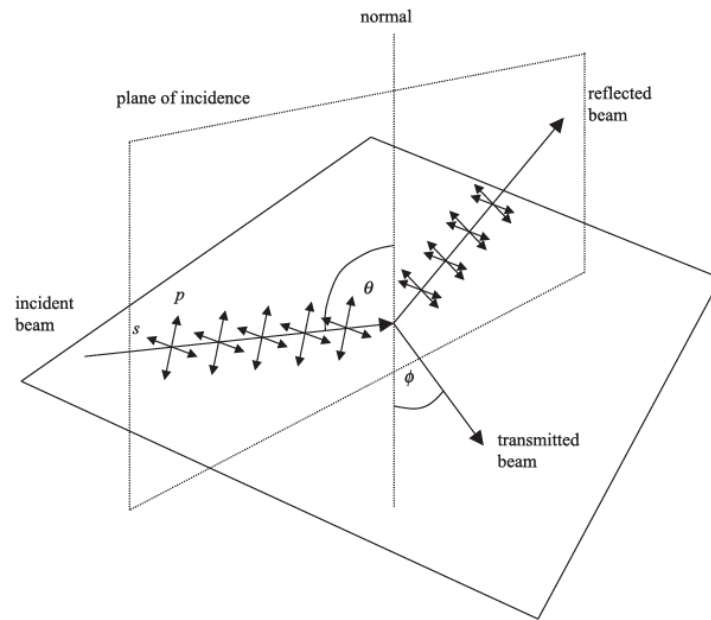


Figura 2.3: Definiciones de luz polarizada s y p, que se definen en términos del plano de incidencia. Tomado de [2]

2.4. Interacción de la luz con la materia

La luz interactúa con la materia de tres maneras: puede ser reflejada, puede ser transmitida o puede ser absorbida, por lo que la intensidad original de la luz puede separarse solo en tres partes como en la ecuación 2.5

$$I_o = I_r + I_t + I_a \quad (2.5)$$

donde las tres partes son las intensidades que se reflejan, transmiten y absorben, respectivamente. Los tres procesos suelen ocurrir simultáneamente hasta cierto punto, por lo que, aunque pueden discutirse de forma independiente, hay que tener en cuenta que los tres procesos tienen lugar al mismo tiempo [2].

2.4.1. Reflexión

La reflexión se produce cuando una onda de luz rebota en la superficie. Aunque el proceso general puede interpretarse como la absorción completa de un fotón y luego la reemisión de un fotón de exactamente la misma longitud de onda en un ángulo reflejado igual al ángulo de incidencia, es más fácil pensar en la reflexión simplemente como el fotón que rebota en la superficie. Sin embargo, la reflexión no es tan sencilla. Aunque el ángulo de incidencia sea igual al ángulo de reflexión y la longitud de onda no sea afectada, las propiedades de polarización del haz reflejado pueden modificarse. La luz no polarizada puede considerarse una combinación aleatoria de la luz polarizada perpendicular al plano de la superficie reflectante (luz polarizada S) y la luz polarizada paralela al plano de la superficie reflectante (luz polarizada P). Sin embargo, debido a que una superficie tiene diferentes relaciones reflectivas para las dos polarizaciones de la luz, el haz reflejado puede tener propiedades de polarización diferentes del haz incidente. Esta propiedad es importante para las superficies que también pueden transmitir a la longitud de onda particular de la luz [2].

2.4.2. Transmisión

Aunque la luz viaja a una velocidad característica a través del vacío, la luz viaja a diferentes velocidades en diferentes medios, como consecuencia, cuando la luz entra en el nuevo medio en ángulo (en lugar de frente), su trayectoria se dobla un poco; esta idea se llama refracción, siendo cuantificada por Snell, llamándose ley de Snell y generando la ecuación 2.6.

$$n_i \sin \theta = n_r \sin \phi \quad (2.6)$$

El índice de refracción n se define como el cociente de la velocidad de la luz en ese medio, dividido por la velocidad de la luz en el vacío, como se ve en la ecuación 2.7,

$$n = \frac{c}{v} \quad (2.7)$$

donde v es la velocidad de la luz en ese medio. Cada fase tiene su propio índice característico de refracción; por lo tanto, tenemos n_i para el índice de refracción

del medio incidente, y n_r para el índice de refracción del medio de refracción en la ecuación 2.6. Las diferentes polarizaciones de la luz tienen diferentes índices de refracción, y estos también dependen en gran medida de la longitud de onda de la luz. Si se reorganiza la ley de Snell, suponiendo que la luz incidente está en el medio de mayor índice de refracción, para dos medios cuales quiera hay un ángulo de reflexión ϕ que llega a ser de 90 grados, siendo que en ese punto la luz no se transmite al medio, sino que pasa paralela a la superficie, y los ángulos de incidencia mayores simplemente se reflejan en la superficie del medio transmisor. Este ángulo, θ_c , se denomina ángulo crítico, dado por la expresión 2.8, el cual es independiente a la longitud de onda y a la polarización [2].

$$\theta_c = \sin^{-1} \frac{n_r}{n_i} \quad (2.8)$$

2.4.3. Absorción

A determinadas longitudes de onda, un medio de transmisión normal absorberá la luz, y en tal caso el fotón se destruye y su energía se convierte en un proceso atómico o molecular, como el cambio de orbita de un electrón, la vibración o giro diferente de una molécula, o algún otro efecto. Por lo general, la(s) longitud(es) de onda de la luz absorbida es característico de la especie absorbente, o cromóforo. Para la mayoría de los procesos espectroscópicos simples, se involucran dos niveles de energía diferentes, de modo que la diferencia entre los niveles de energía, ΔE , está relacionada con la frecuencia de la luz absorbida por la condición de frecuencia de Bohr, como se ve en la ecuación 2.9, donde h es la constante de Planck y ν es la frecuencia de la luz absorbida.

$$\Delta E = h\nu \quad (2.9)$$

En el caso del medio absorbente, el índice de refracción es complejo y se reescribe como una expresión compleja dependiente de la longitud de onda, como se ve en 2.10, donde $\hat{n}$ es el índice complejo de refracción y k el factor de atenuación relacionado con un coeficiente de absorción, α [2].

$$\hat{n} = n - ik \quad (2.10)$$

2.4.4. Luminiscencia

La luminiscencia es la emisión de luz por alguna sustancia, y ocurre por los estados excitados electrónicamente, lo que quiere decir que ocurre después de absorber energía. La luminiscencia está formalmente dividida en dos categorías, la fluorescencia y fosforescencia, dependiendo de la naturaleza del estado excitado [18].

- **Fluorescencia:** En este proceso, se excita un electrón a estado excitado singlete debido a la absorción de un fotón, en consecuencia el retorno al estado fundamental produce rápidamente la emisión de un fotón, teniendo un tiempo de vida típico cercano a 10ns [18].

- **Fosforescencia:** Similar a la fluorescencia, siendo en este caso la emisión de luz desde estados excitados triplete, teniendo una duración mayor para regresar al estado fundamental, provocando que se emita luz por varios minutos [18].

2.5. Espectroscopía

2.5.1. ¿Qué es la espectroscopía?

La espectroscopía es el estudio de la materia mediante radiación electromagnética, o en términos más simples, por lo descrito en la sección 2.2, podemos decir que la espectroscopía es el estudio de la materia mediante el uso de la luz. Fue Isaac Newton el primero en demostrar que la luz blanca se produce mediante la combinación de luces de varios colores, y fue quien propuso el nombre de espectro para la banda fantasmal de colores que se forma cuando una rendija de luz blanca pasa a través de un prisma.

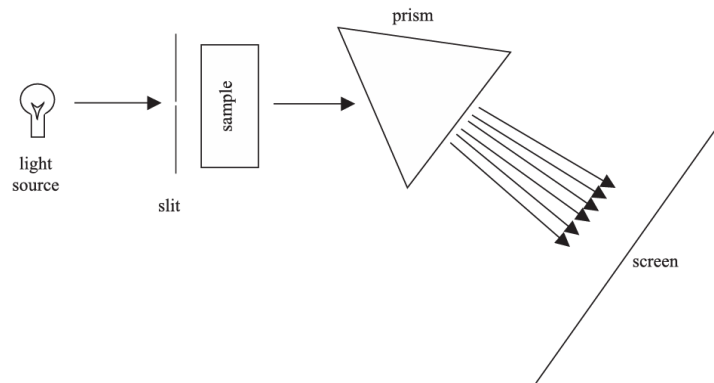


Figura 2.4: Espectroscopio Bunsen y Kirchhoff. Tomado de [2]

Con el tiempo, esto motivó a mas investigadores a realizar experimentos con el espectro de la luz, prueba de esto está el esquema del espectroscopio de Bunsen y Kirchhoff, como se ve en la figura 2.4, donde la luz de alguna fuente pasa a través de una rendija, una muestra y un prisma antes de ser proyectada en una pantalla, la luz absorbida está representada por líneas oscuras superpuestas en un espectro de tipo de arco de lluvia. Si la muestra se calienta y emite luz, entonces la fuente se omite y el espectro consiste en líneas brillantes de luz en longitudes de onda que son emitidas por la muestra. Bunsen y Kirchhoff demostraron que estas longitudes de onda absorbidas o emitidas son características de los elementos de la muestra. Ejemplo de esto es el espectro del Hidrógeno, como se ve en la figura 2.5, el cual consta de cuatro líneas en la región visible [2].

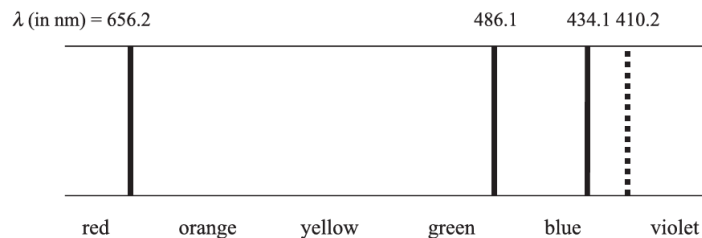


Figura 2.5: Espectro visible del Hidrógeno. Tomado de [2]

Actualmente, la espectroscopía es ampliamente utilizada para realizar análisis de muestras desconocidas, teniendo la ventaja que muchas de las pruebas realizadas con esta no son destructivas. Además, existen varios tipos de espectroscopía, según el tipo de estudio que se desea realizar, algunas variantes de esta son: ultravioleta-visible, infrarroja, Raman, de rayos X, etc., donde su principal diferencia está en el tipo de luz utilizada y la forma en que se trabaja con esta, pero todas parten de la misma idea base vista en esta sección.

2.5.2. Ruido en la espectroscopía

En la espectroscopía, al igual que en cualquier técnica de medición, el concepto de ruido tiene un impacto importante en lo que realmente se está midiendo. En casi todas las técnicas espectroscópicas, se mide algún tipo de señal (una intensidad de un haz de luz, por ejemplo) por un detector. En casi todas las mediciones de este tipo, hay variaciones en la señal que no se deben al proceso de absorción/emisión. En cambio, estas variaciones son causadas por varios factores inherentes al proceso de medición en sí. Es decir, son ruido [2].

La señal es la medida deseada, y en casi todas las técnicas espectroscópicas es de naturaleza eléctrica. Una señal cero podría ser una corriente cero, pero también podría ser alguna corriente de referencia arbitraria. Las desviaciones de esta referencia representan una señal a ser medida. El ruido es la fluctuación en una señal que tiende a oscurecer la señal real. Un ejemplo espectroscópico sería un parpadeo en la fuente de un calorímetro de haz simple, causando una fluctuación en la intensidad de la luz medida por el detector. En este caso, una señal no deseada se impone sobre una medida deseada, esto es ruido [2].

El ruido tiene varias fuentes, y a algún nivel, el ruido en espectroscopía es inherente debido a la propia naturaleza de la luz y los átomos. El ruido aleatorio es, como su nombre sugiere, impredecible (por ejemplo, el parpadeo de una lámpara). El ruido no aleatorio tiene algún tipo de patrón temporal (por ejemplo, alguien que camina de un lado a otro junto a un banco óptico sensible, provocando vibraciones que interfieren con la medición de los espectros). La interferencia es un tipo de ruido causado por el entorno que impone sus características al sistema espectroscópico, por ejemplo, cualquier generador de frecuencia electrónica tiene el potencial de imponer ruido de interferencia en

un experimento espectroscópico. Algunos ruido de interferencias pueden ser no aleatorios, como lo pueden ser las señales infrarrojas de dispositivos electrónicos, como computadores e impresoras, que pueden interferir en las mediciones. El ruido aleatorio del que no se puede escapar a veces se denomina ruido blanco, cuya potencia es independiente de la frecuencia y aparece como un cierto nivel base en cualquier frecuencia. Dado que se necesita menos energía para producir ruido de baja frecuencia, hay más de este tipo de ruido a bajas frecuencias que a altas frecuencias.[2]

Finalmente, el ruido podría deberse a señales no deseadas. Este tipo de ruido es especialmente problemático por dos razones. Primero, puede ser difícil identificar la fuente de la señal no deseada. Para los espectroscopistas que construyen su propio equipo, esto puede ser un obstáculo importante. Segundo, aunque la fuente puede ser identificable, la eliminación de esta puede ser problemática [2].

2.5.3. Reducción de ruido en espectroscopía

Minimizar el ruido en las mediciones espectroscópicas es una preocupación variable para los experimentalistas. Si se utiliza un espectrómetro comercial, su diseño mismo debería tener en cuenta factores de minimización del ruido. Por otro lado, quien trabaja con un espectrómetro casero tendrá que esforzarse por ver que se minimice el ruido ya sea en el diseño del instrumento o en el rendimiento del experimento (o en ambos).

La capacidad de extraer información de cualquier técnica espectroscópica suele estar limitada por la relación señal-ruido (SNR, por sus siglas en inglés). Los espectroscopistas consideran el SNR como una indicación de la calidad de un espectro. El SNR se define como el valor promedio de la señal dividida entre su desviación estándar, como se ve en la ecuación 2.11, donde $\bar{S}$ es el valor promedio de la señal, y σ_y es su desviación estándar. La mayoría de los instrumentos espectroscópicos necesitan un SNR de al menos tres para apoyar la identificación de una transición espectral. Al igual que con cualquier determinación de la desviación estándar, la precisión del SNR mejora con el número de mediciones promediadas [2] [3].

$$SNR = \frac{\bar{S}}{\sigma_y} \quad (2.11)$$

Algunas de las cosas que se pueden hacer para minimizar el ruido en la espectroscopía son [2]:

- Aislar el sistema: El ruido puede ser causado por circuitos eléctricos, vibraciones mecánicas o la presencia de señales no deseadas de otras fuentes de luz o reflexiones no deseadas. Si se aísla el espectrómetro o la fuente de luz o el detector de estas cosas, se puede minimizar su impacto negativo en un espectro.
- Incrementar la intensidad de la fuente: Cuanto más intensa sea la fuente

de luz, menos debería verse afectado el espectro por fuentes de luz no deseadas.

- Disminuir la resolución espectral: Aunque no es la mejor solución, a veces es un compromiso necesario si no hay otra opción posible.
- Promediar múltiples escaneos: Debido a la prevalencia de las computadoras personales, es posible medir múltiples escaneos espectrales de una muestra, almacenarlos en memoria de la computadora y luego promediarlos. El promedio disminuye la cantidad de ruido aleatorio, que tiende a cancelarse a sí mismo si es realmente aleatorio. Una regla bien conocida es que el SNR aumenta proporcionalmente a la raíz cuadrada del número de escaneos. Por lo tanto, un espectro que resulta de 16 escaneos debería tener un SNR que es cuatro veces mayor que un espectro de un solo escaneo. También hay una compensación con múltiples escaneos. Se necesita más tiempo para recopilar más espectros.

Una unidad utilizada para expresar el SNR son los decibelios, haciendo más fácil comparar los valores. Obtener los valores en decibels depende de si se está trabajando con potencias o si se está trabajando con amplitudes. En el caso de las potencias se usa la siguientes ecuación:

$$dB = 10 * \log_{10} \frac{P_2}{P_1} \quad (2.12)$$

, para el caso de trabajar con amplitudes, se utiliza la siguiente ecuación [19]:

$$dB = 20 * \log_{10} \frac{A_2}{A_1} \quad (2.13)$$

Otra herramienta que permite realizar una comparación entre el antes y después de la reducción de ruido de una señal es el análisis de componentes principales (PCA, por sus siglas en ingles). El PCA es un enfoque popular para derivar un conjunto reducido de características de un gran conjunto de variables. El PCA es un algoritmo matemático que reduce la dimensionalidad de los datos (una matriz de datos X de $n * p$) mientras retiene la mayor parte de la variación en el conjunto de datos, esto lo logra identificando direcciones, llamadas componentes principales. La dirección del componente principal de los datos es aquella de la cual las observaciones varían más. Esto facilita la manera en que se grafican los datos, agrupando los datos que son más parecidos en zonas específicas de la grafica. En este caso se busca observar que tanto varían los espectros de una muestra específica comparándolos con los espectros de otras muestras antes y después de la reducción del ruido, por lo que la matriz de datos serán los espectros captados y los filtrados [20] [21].

2.6. Espectroscopía Raman

2.6.1. Dispersión Raman

La espectroscopía Raman es ampliamente útil para la identificación de estructuras moleculares, ya que con esta es posible detectar vibraciones fundamentales características de las muestras a analizar. Esto es posible gracias a la interacción de la radiación electromagnética con las vibraciones moleculares de la muestra, transfiriendo la energía de los fotones a la molécula, cambiando su estado vibratorio. La espectroscopía Raman es un evento de dispersión de la luz inelástica de dos fotones, donde el fotón incidente tiene mucha más energía que la energía cuántica vibracional, y pierde parte de su energía debido a su vibración molecular, mientras que la energía restante se dispersa como un fotón de frecuencia reducida [1].

En la espectroscopía Raman, se utiliza una fuente de luz monocromática, la cual, mediante el uso de un arreglo óptico, se hace incidir sobre una muestra, con el objetivo de que la luz incidente sea dispersada. Los fenómenos de dispersión de la luz pueden describirse clásicamente en términos de radiación electromagnética producida por dipolos oscilantes inducidos en las moléculas de la muestra por los campos electromagnéticos de la radiación incidente. Los fotones dispersados por la luz incluyen principalmente la dispersión Rayleigh, la cual es igual a la luz incidente y otra pequeña cantidad de luz dispersada, es la generación de dicha luz lo que se conoce como efecto Raman.

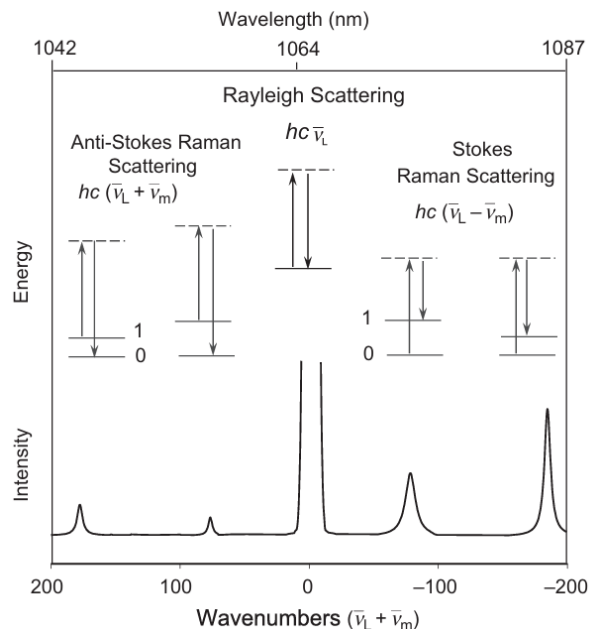


Figura 2.6: Espectro efecto Raman. Tomado de [1]

En un experimento típico Raman, la fuente de luz monocromática que irradia a la muestra es un láser, el cual está disponible en la región espectral ultravioleta, visible y cercano al infrarrojo. Por lo tanto, si se utiliza la excitación visible, la luz dispersada Raman también estará en la región visible. Los procesos de Rayleigh y Raman están representados en la figura 2.6, en la cual se puede apreciar que para la luz Rayleigh dispersada elásticamente no se pierde energía, mientras que los fotones dispersados Raman pierden algo de energía en relación con la energía de excitación. Los picos en el espectro que están en longitudes de onda mayor a la de la fuente de excitación, y por lo tanto menos energéticos, se les conoce como dispersión Stokes, mientras que los que están en una longitud de onda menor, y que por lo tanto son más energéticos, se les conoce como dispersión anti Stokes. La dispersión Raman es mucho menos probable que la dispersión Rayleigh, con una intensidad observada que es 10^{-6} veces la de la luz incidente para una dispersión Raman fuerte [1].

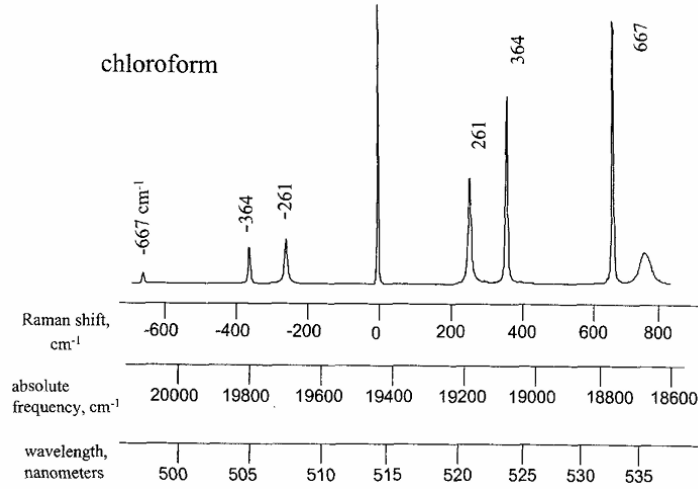


Figura 2.7: Espectro Raman Cloroformo. Tomado de [3]

En la espectroscopía Raman se suele utilizar el número de onda (cm^{-1}) en lugar de la longitud de onda (nm) para graficar los espectros. Esto se hace así ya que la espectroscopía Raman se encarga de medir los desplazamientos energéticos, los cuales son independientes de la longitud de onda de la fuente de luz utilizada, lo que significa que los desplazamientos siempre serán los mismos sin importar la longitud de onda de la fuente. Ejemplo de esto está en la figura 2.7, en la cual se representa el espectro Raman del cloroformo tomado con una fuente de luz de 514.5 nm, donde el eje x se muestra en tres escalas diferentes, pero normalmente se representa como un desplazamiento Raman en centímetros recíprocos en relación con la frecuencia del láser ($19,435 \text{ cm}^{-1}$ en este caso). El desplazamiento Raman ($\Delta\nu$) está dado por la ecuación 2.14, donde ν_o es el

numero de onda del láser, y v_1 es el número de onda de la luz dispersada. La relación del número de onda v con la longitud de onda λ está dato por la ecuación 2.15, donde para obtener la unidad de cm^{-1} , λ debe estar en cm [3].

$$\Delta v = v_0 - v_1 \quad (2.14)$$

$$v = \frac{1}{\lambda} \quad (2.15)$$

2.6.2. Ruido en el espectro Raman

Dado que la dispersión Raman es bastante débil en comparación con la fluorescencia o la absorción, las mediciones Raman han superado históricamente los límites de la tecnología existente para reducir el ruido y aumentar el SNR. Con el tiempo, las señales Raman han aumentado y las fuentes de ruido han disminuido por razones tecnológicas, produciendo un SNR mucho mayor para una muestra determinada [3].

Algunas de las fuentes de ruido en el espectro Raman son [3]:

- **Ruido de disparo de señal:** Relacionado con la naturaleza estadística de la emisión y detección de fotones.
- **Ruido térmico:** Generado por el movimiento térmico de los electrones en los componentes electrónicos del detector.
- **Ruido de lectura:** Asociado con el proceso de lectura de la señal del detector.
- **Ruido por fluorescencia:** Proviene de la emisión de luz por parte de la muestra o impurezas cuando son excitadas por el láser.

2.7. Espectrómetros

En términos simples, un espectrómetro es un dispositivo capaz de medir la distribución de la luz que ingresa en este en función de su energía o longitud de onda. El problema es que no hay una descripción única de como es que estos están compuestos, ya que los espectrómetros están diseñados para aprovechar la forma única en que los diferentes tipos de luz interactúan con la materia. Sin embargo, la mayoría de los espectrómetros tienen ciertos elementos comunes, los cuales permiten entender la manera en que la mayoría de estos dispositivos trabajan [2].

2.7.1. Espectrómetros de emisión y absorción

Desde el punto de vista energético, la emisión y la absorción son procesos opuestos, teniendo en la absorción un fotón absorbido por un átomo o una molécula, provocando un proceso; y en el caso de la emisión, se produce un

proceso y emite un fotón. En ambos casos, los factores importantes a tener en cuenta son la energía del fotón y su intensidad (cuantos fotones de cada energía están involucrados). Debido a que las mismas dos cosas son importantes en ambos casos, los espectrómetros para espectroscopía de emisión y absorción se componen en gran medida de componentes similares, pero en un orden u orientación. Para la espectroscopía de emisión o absorción son necesarios los siguientes componentes: una fuente de energía o fotones (bombilla, láser, magnetrón, sincrotrón, electricidad, una flama o una varilla de cerámica caliente), un método de diferenciación de energía (monocromador), una muestra que absorba o emita electrones (cualquier cosa) y un detector (absorbente de calor, película fotográfica o dispositivos electrónicos sensibles a la luz) [2].

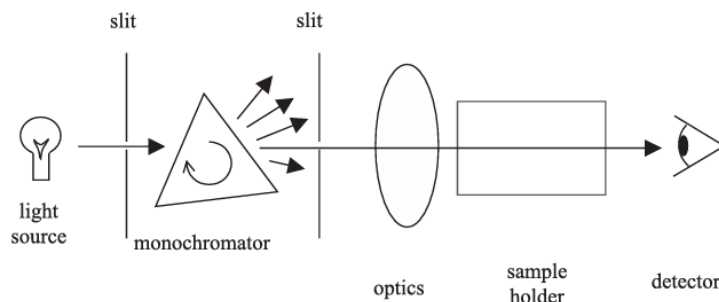


Figura 2.8: Esquema de espectrómetro de absorción. Tomado de [2]

Los espectrómetros de emisión son esencialmente los mismos que se muestran en la figura 2.8. Sin embargo, para los espectrómetros de emisión, la luz de interés no es la que incide sobre la muestra, sino la luz que proviene de la muestra. En muchos casos, los fotones interactúan con un compuesto y luego se vuelven a emitir a una longitud de onda diferente (estas interacciones incluyen fluorescencia, fosforescencia y dispersión Raman). Debido a que se debe determinar la longitud de onda de la luz emitida, generalmente se coloca un monocromador después de la muestra para diferenciar la luz emitida, esto se ilustra en la figura 2.9. La emisión generalmente ocurre en todas las direcciones, pero se detecta mejor cuando es poco probable que la luz de la fuente interfiera, esto es especialmente cierto para las espectroscopías de emisión inducidas por láser.

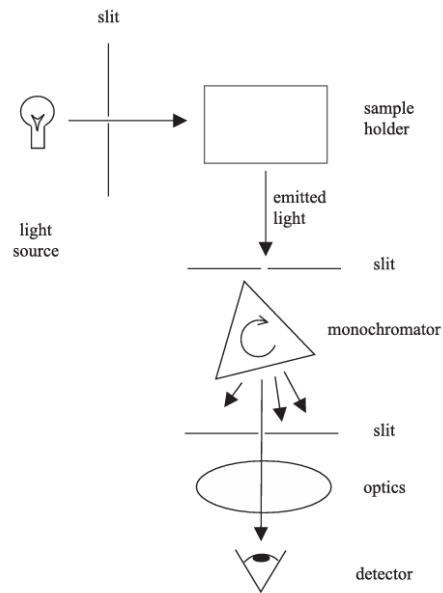


Figura 2.9: Esquema de espectrómetro de emisión
. Tomado de [2].

2.7.2. Monocromador

Un monocromador es el método de diferenciación de energía utilizado en los espectrómetros. Consiste en un rendija de entrada, seguida de un espejo para asegurar que la luz sea paralela, una rejilla de difracción (o filtros de colores, o prismas), un espejo de enfoque, que dirige la luz dispersada a la rendija de salida, y a un detector, pudiendo separar la luz por longitudes de onda, permitiendo determinar qué energías de luz son absorbidas o emitidas por una muestra particular, la figura 2.8 muestra la implementación de un monocromador en un espectrómetro básico de absorción, y la figura 2.9 muestra su implementación en un espectrómetro de emisión, en donde, dependiendo del ángulo de difracción de la rejilla o prisma, se dejan pasar longitudes de onda específicas para que pasen por la muestra y sean captadas por el detector. La figura 2.10 muestra un esquema de una rejilla de difracción con la radiación policromática incidente y la luz difractada resultante. Cuando la radiación incidente en fase se refleja en la rejilla, la radiación de una longitud de onda adecuada se enfoca hacia la rendija de salida. En la rendija de salida, la radiación enfocada estará en fase solo para una longitud de onda seleccionada y sus múltiplos enteros, los cuales interferirán constructivamente y pasarán a través de la rendija de salida. Otras longitudes de onda interferirán destructivamente y no saldrán del monocromador. Así, cada una de las ranuras actúa como una fuente individual similar a una rendija de radiación, difractándola en varias direcciones. Típicamente, se utiliza un filtro selectivo para eliminar las longitudes de onda de orden superior.

Cuando la red se rota ligeramente, una longitud de onda ligeramente diferente alcanzará el detector [2] [1].

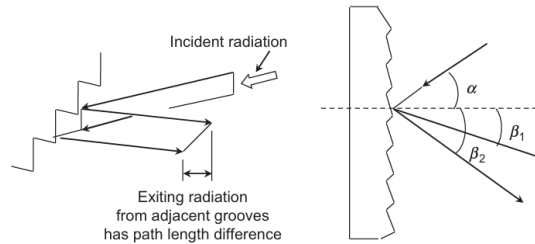


Figura 2.10: Rejilla de difracción
. Tomado de [1].

2.7.3. Sistemas ópticos

La materia trata la manipulación controlada de fuentes de onda (o rayos) mediante la interposición de cuerpos reflectores y/o refractantes, dejando de lado los efectos de difracción. Es la implementación de estos cuerpos lo que se conoce como sistema óptico. Los espectrómetros se apoyan de sistemas ópticos para la manipulación de la luz, con el objetivo de implementar diseños más eficientes y obtener mediciones más precisas. Los elementos que pueden estar involucrados en un sistema óptico tiene un comportamiento específico con la luz, el cual es aprovechado para su implementación en el diseño del sistema [17]. Los elementos ópticos son descritos a continuación:

Lente: Una lente, es un dispositivo refractor (es decir, una discontinuidad en el medio predominante) que reconfigura una distribución de energía transmitida. Eso es cierto ya sea que se trate de rayos UV, ondas de luz, infrarrojos, microondas, ondas de radio o incluso ondas de sonido. La configuración de una lente está determinada por la remodelación requerida del frente de onda para el que está diseñada. Su funcionamiento se explica suponiendo que se tiene una onda esférica divergente (como la que sale de un punto) viajando por un medio con índice de refracción n_i (como el aire), y esa onda choca con una superficie curva de un material con índice de refracción mayor n_t (como el vidrio), la velocidad de la onda disminuye, porque la luz se mueve más lento en materiales con mayor índice de refracción. La parte central de la onda, que entra primero al nuevo medio, empieza ir más lenta, mientras que las partes exteriores de la onda aún están en el medio rápido, por lo que siguen avanzando más rápido, esa diferencia de velocidad hace que las partes exteriores alcancen a la parte central, lo que va aplanando la onda. Si la superficie de la lente tiene una forma adecuada, se puede transformar esa onda esférica en una onda plana, es decir, los frentes de onda dejan de ser curvos y se vuelven rectos, esto es esencialmente enfocar la luz, ya que la misma idea puede ser explicada con rayos de luz en lugar de frentes de onda. Al entrar al material más denso, los rayos de luz se desvían hacia la normal (una línea perpendicular a la superficie), si la curvatura

de la lente es correcta, los rayos que venían divergentes salen paralelos al otro lado, esto se ilustra en la figura 2.11, donde F_1 y F_2 son los puntos focales, siendo F_1 en donde se encuentra la fuente puntual y F_2 el punto hacia donde se transmiten las ondas [17].

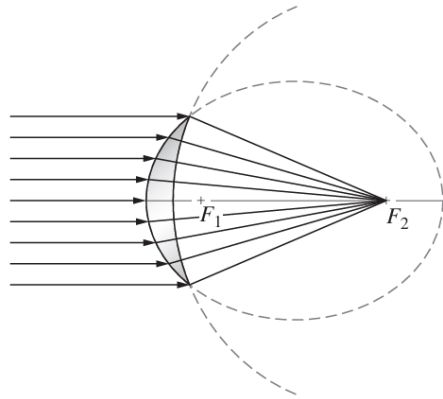


Figura 2.11: Comportamiento de una lente
. Tomado de [17].

Existen dos categorías principales para los tipos de lentes, convexas (las cuales hacen que los rayos de luz se junten en un punto llamado foco) y cóncavas (las cuales separan los rayos de luz como si vinieran de un punto detrás de la lente). A partir de estas surgen varias subcategorías, las cuales se diferencian entre sí por el tipo de forma que tienen sus caras:

- Biconvexa
- Plano convexa
- Menisco convexa
- Bicóncava
- Plano cóncava
- Menisco cóncava

Esto se muestra en la figura 2.12, siendo R_1 y R_2 los radios de curvatura para las superficies de la lente.

CONVEX	CONCAVE
 $R_1 > 0$ $R_2 < 0$ Biconvex	 $R_1 < 0$ $R_2 > 0$ Biconcave
 $R_1 = \infty$ $R_2 < 0$ Planar convex	 $R_1 = \infty$ $R_2 > 0$ Planar concave
 $R_1 > 0$ $R_2 > 0$ Meniscus convex	 $R_1 > 0$ $R_2 > 0$ Meniscus concave

Figura 2.12: Tipos de lentes
. Tomado de [17].

Algo igual de importante es la anchura de las lentes, ya que de esta dependerá la distancia a la que se ubicaran los puntos focales [17].

Bloqueadores: La naturaleza intrínseca finita de todas las lentes exige que recojan solo una fracción de la energía emitida por una fuente puntual. Por lo tanto, la limitación física que presenta la periferia de una lente simple determina cuales rayos deben entrar en el sistema para formar una imagen. En ese sentido, el diámetro claro o sin obstrucciones de la lente funciona como una apertura en la que fluye la energía. Cualquier elemento, ya sea el borde de una lente o un diafragma separado, que determina la cantidad de luz que llega a la imagen se conoce como tope de apertura (A.S. por sus siglas en ingles). El tope de apertura de un sistema óptico es la entidad física particular que limita la amplitud del haz de luz procedente de un punto de un objeto axial a medida que pasa a través del sistema. Otro concepto, útil para determinar si un rayo determinado atravesará o no todo el sistema óptico, es la pupila. Esta es simplemente una imagen del tope de apertura. La pupila de entrada de un sistema es la imagen del tope de apertura visto desde un punto axial en el objeto que mira a través de los elementos que preceden al tope. Por el contrario, la pupila de salida es la

imagen del A.S. visto desde un punto axial en la imagen mirando a través de las lentes interpuestas, si las hay. Estos elementos entran dentro de la clasificación de bloqueadores, los cuales tienen la tarea de bloquear la luz de una forma específica, y su implementación se ve ilustrada en la figura 2.13 a modo de ejemplo [17].

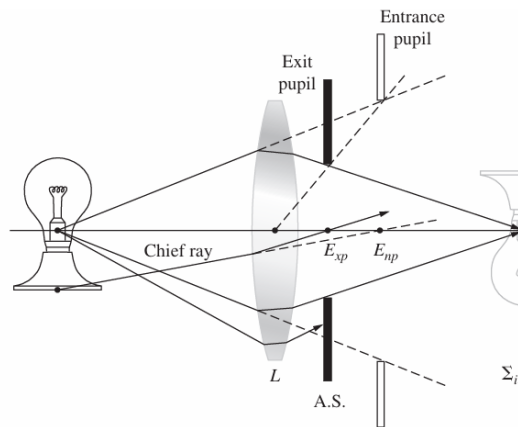


Figura 2.13: Bloqueadores
. Tomado de [17].

Espejos: Un espejo puede ser simplemente un pedazo de vidrio negro o una superficie de metal finamente pulida, que tiene la función de reflejar la luz que incide en este. También pueden ser contruidos con otros tipos de materiales, lo cual dependerá del rango de longitudes de onda que se pretende sean reflejadas y la aplicación para la cual se necesita, existiendo variaciones con recubrimientos de plata, aluminio, películas dieléctricas, etc. Existen varios tipos de espejos, según el tipo de reflexión que se desea lograr. Algunos de estos son [17]:

- Planos: Cuentan con una superficie plana, y redirigen los haces de luz sin alterar su enfoque ni su forma.
- Móviles: Son espejos que tienen la habilidad de moverse lado a lado y de arriba a abajo que les permite cambiar la dirección hacia que la luz es redirigida.
- Asféricos o parabólicos: Capaz generar una reflexión de haz colimado si se incide con una fuente puntual, y enfocar en un punto focal si se incide con un haz paralelo.
- Esféricos: Existen de dos tipos, cóncavos, los cuales focalizan la luz, y convexos, los cuales divergen la luz, parecido al efecto obtenido por las lentes que tienen estas formas, ejemplo de estos pueden apreciarse en la figura 2.14, donde el espejo de la izquierda es un convexo y el de la derecha un cóncavo.

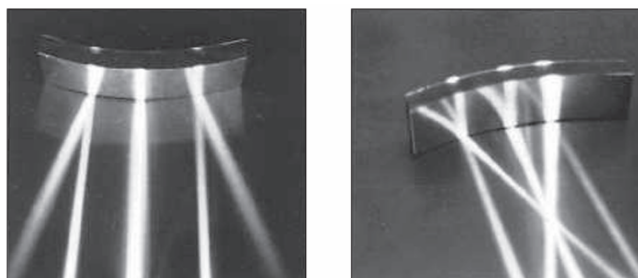


Figura 2.14: Espejos esféricos
.Tomado de [17].

Prismas: Los prismas desempeñan muchas funciones diferentes en la óptica; hay combinaciones de prismas que sirven como divisores de haz, dispositivos polarizadores e interferómetros. A pesar de esta diversidad, la gran mayoría de las aplicaciones hacen uso de solo una de las dos funciones principales del prisma. En primer lugar, un prisma puede servir como dispositivo dispersivo, como lo hace en una variedad de analizadores de espectro. Como tal, es capaz de separar, hasta cierto punto, los componentes de frecuencia constituyentes en un haz de luz policromático. Su segunda función, y la más común, es efectuar un cambio en la orientación de una imagen o en la dirección de propagación de un haz. Los prismas se incorporan a muchos instrumentos ópticos, a menudo simplemente para plegar el sistema en un espacio confinado. Hay prismas de inversión, de reversión, y prismas que desvían una viga sin inversión ni reversión, y todo esto sin dispersión [17].

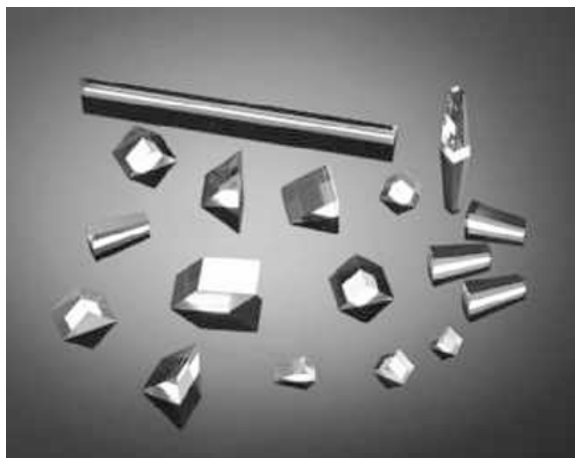


Figura 2.15: Selección varios tipos de prismas
. Tomado de [17].

Fibras ópticas: Una fibra óptica es un hilo muy delgado de material dieléctri-

co transparente, como el vidrio o el cuarzo fundido. Este guía la luz en su interior a lo largo de grandes distancias mediante un fenómeno llamado reflexión interna total. Cuando la luz entra a la fibra en un ángulo adecuado, rebota repetidamente en las paredes internas de la fibra sin escapar, gracias a la diferencia de índices de refracción entre el núcleo (centro de la fibra) y el revestimiento externo, como se puede apreciar en la figura 2.16. Este principio fue demostrado inicialmente por John Tyndall en 1870. Cuando el diámetro de la fibra es mucho mayor que la longitud de onda de la luz, el comportamiento puede entenderse con la óptica geométrica (rayos), pero si el diámetro es similar a la longitud de onda, la propagación se parece a la de las microondas en una guía de ondas, y se necesita la óptica física para describirla.

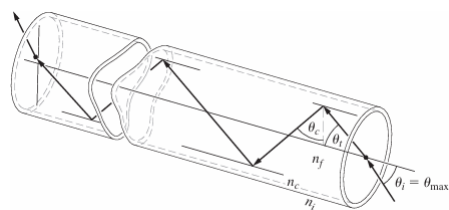


Figura 2.16: Rayos en una fibra óptica revestida
. Tomado de [17].

2.7.4. Fuentes de luz

La fuente de luz ideal generaría una intensidad constante en todas las longitudes de onda con ruido bajo y estabilidad de la salida a largo plazo. Lamentablemente, no existe ninguna fuente con esas características. Históricamente se han utilizado dos fuentes de luz en los espectrómetros UV-visible. La lámpara de arco de deuterio, que proporcionaba una intensidad continua bastante buena en la región UV, y una intensidad útil en la región visible. Y la lámpara de halógeno de wolframio, que conseguía una buena intensidad en todo el rango visible y en parte del espectro UV [22].

Más recientemente, se utiliza ampliamente una sola lámpara de xenón pulsante como única fuente, poseyendo ventajas significativas con respecto al uso de las otras dos lámparas convencionales [22].

Otra lámpara que se utiliza como fuente de luz en los espectrómetros es la lámpara halógena de tungsteno, la cual presenta una buena intensidad en parte del espectro UV y en todo el rango visible y cercano al infrarrojo (350-3000nm). En aquellos espectrómetros UV-visibles que utilizan tanto una lámpara de deuterio como una halógena de tungsteno, se usa un selector de fuente para cambiar de una lámpara a la otra según corresponda, o bien se mezcla la luz procedente de ambas fuentes para originar una sola fuente de banda ancha [22].

2.7.5. Espectrógrafo Czerny-Turner

Una de las configuraciones más comunes para implementar un espectrógrafo es siguiendo el diseño Czerny-Turner. Como se puede ver en la figura 2.17, este instrumento consiste en una ranura, que permite el paso de luz policromática hacia un espejo colimador, el cual redirigirá la luz a una rejilla de difracción, la luz dispersada es captada por otro espejo, el cual enfoca las longitudes de onda en el plano focal [3].

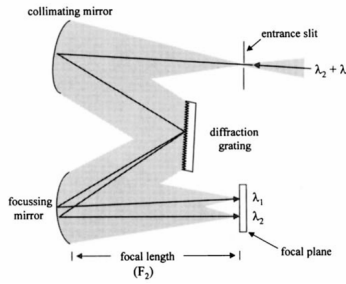


Figura 2.17: Diagrama espectrógrafo Czerny-Turner
. Tomado de [3].

A diferencia de los diagramas vistos en la sección 2.7.1, el diseño Czerny-Turner no se basa en la manipulación del ángulo de difracción de la rejilla (o prisma) para obtener una longitud de onda específica, aquí la rejilla mantiene un ángulo fijo, haciendo que la luz de entrada se descomponga en todas sus longitudes de ondas, las cuales ocupan un espacio específico en el plano focal, como se aprecia en la figura 2.18, la cual es la representación de un plano focal ideal de un espectrógrafo Czerny-Turner cuando se tiene una fuente de luz compuesta por longitudes de onda de 400nm, 500nm y 600nm. Este tipo de arreglo lo vuelve ideal para ser implementado en espectrómetros de emisión, en especial los que utilizan un láser como fuentes luz, como en la espectroscopía Raman [3].

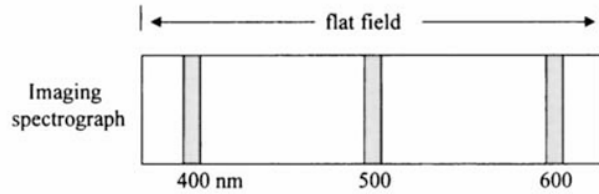


Figura 2.18: Plano focal ideal espectrógrafo Czerny-Turner
. Tomado de [3].

2.7.6. Detectores

Los detectores, son la parte final de los espectrómetros, y estos son los encargados de capturar las intensidades de los espectros generados por todos los demás componentes. Estos pueden ir desde absorbentes de calor, películas fotográficas, y en diseños más modernos, dispositivos electrónicos sensibles a la luz. Estos últimos convierten las intensidades lumínicas en señales eléctricas, teniendo una gran variedad de opciones, que difieren entre si por la tecnología implementada con la que funcionan. Algunos de los tipos de detectores electrónicos son:

PM (Photon Counting): Para los espectrómetros equipados con monocromadores y detectores individuales, la luz dispersada es recogida y enfocada en un tubo fotomultiplicador (PM), que convierte los fotones en una señal eléctrica. El tubo consta de un fotocátodo que emite electrones cuando los fotones lo golpean; una serie de dinodos, cada uno de los cuales emite un número de electrones secundarios cuando es golpeado por un electrón; y un ánodo que recoge estos electrones como señal de salida [23].

Arreglo de foto diodos (phtodiode array): Los detectores de fotones multicanal consisten en un conjunto de pequeños dispositivos fotosensibles que pueden convertir una imagen óptica en un patrón de carga que se puede leer como un espectro. Por lo general, estos detectores de tipo array no son lo suficientemente sensibles para detectar señales muy débiles, como las Raman [23].

CCD (Charge-Coupled Device): Un CCD es un semiconductor basado en silicio, dispuesto como una matriz de elementos fotosensibles, cada uno de los cuales genera foto electrones y los almacena como una pequeña carga. Cada columna de la matriz corresponde a una longitud de onda diferente. En la figura 2.19 se ve un ejemplo CCD bidimensional colocado en el plano focal de un espectrómetro dispersivo, donde las longitudes de onda dispersadas inciden en zonas específicas del sensor [3][23].

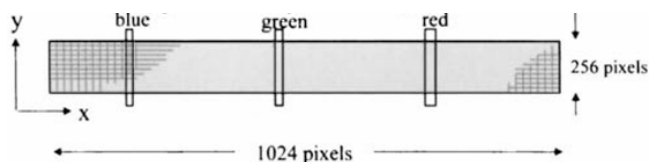


Figura 2.19: Ejemplo de la orientación de un CCD en el plano focal de un espectrógrafo, mostrando las imágenes de la rendija de entrada para la luz incidente de tres longitudes de onda

. Tomado de [3].

2.7.7. Espectrómetros Raman

La instrumentación Raman debe ser capaz de eliminar la fuerte radiación dispersada Rayleigh, mientras se analiza la radiación dispersa Raman débil. Un instrumento Raman suele constar de una fuente de excitación láser (UV, visible o cercano al infrarrojo), una óptica de recogida, un analizador espectral

(monocromador) y un detector. Esto se ejemplifica en la figura 2.20, la cual es un diseño genérico de un espectrómetro Raman, mostrando los componentes principales, láser, óptica de colección, analizador de longitud de onda, detector, computadora. Muchas variaciones de geometría y componentes son de uso común. El filtro de rechazo del láser a menudo se denomina como filtro Notch [3] [1].

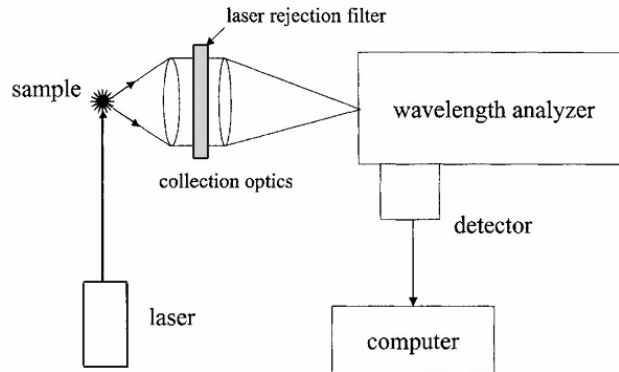


Figura 2.20: Espectrómetro Raman genérico
. Tomado de [3].

Cualquier espectrómetro Raman debe observar una dispersión débil en presencia de una dispersión Rayleigh mucho más fuerte o una reflexión difusa que se produce a la longitud de onda del láser. Por lo tanto, los espectrómetros Raman deben tener un excelente rechazo de luz parásita. El filtro de rechazo (o Notch), es pequeño y simple, y permite el uso de un pequeño espectrógrafo de una sola etapa, volviéndolos más pequeños y eficientes que un diseño de espectrómetro doble o triple para conseguir el mismo efecto [3].

Aunque la luz dispersada por Raman ocurre en todas direcciones, las configuraciones experimentales más comunes para recolectar la radiación Raman son típicamente geometrías de retrodispersión de 90° y 180° utilizando óptica refractiva y reflectante, así como la utilización de interferómetros para separar la luz en componentes individuales, esto permite tener un diseño base para un espectrómetro Raman, como el que se muestra en la figura 2.21 [1].

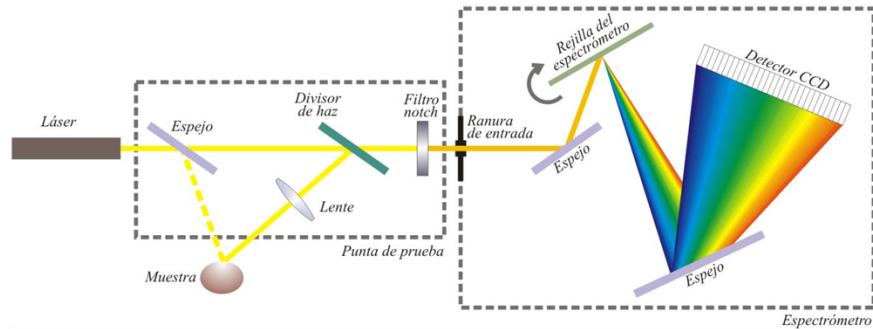


Figura 2.21: Espectrómetro Raman genérico
. Tomado de [?].

2.8. Procesamiento de señales

2.8.1. ¿Qué es el procesamiento de señales?

En general, el término procesamiento de señales se refiere a la ciencia de analizar procesos físicos que varían en el tiempo. Como tal, el procesamiento de señales se divide en dos categorías, procesamiento de señales analógicas y procesamiento de señales digitales. El término analógico se utiliza para describir una forma de onda que es continua en el tiempo y puede tomar valores de amplitud. Un ejemplo de señal analógica es algún voltaje que se puede aplicar a un osciloscopio, resultando en una visualización continua como función del tiempo. El término señal en tiempo discreto se usa para describir una señal cuya variable de tiempo independiente está actualizada de tal manera que solo conocemos el valor de la señal en instantes discretos en el tiempo. Así, una señal en tiempo discreto no se representa mediante una forma de onda continua, sino más bien como una secuencia de valores. Además de cuantizar el tiempo, una señal en tiempo discreto también puede cuantizar la amplitud de la señal. La diferencia entre una señal en tiempo continuo y en tiempo discreto se ilustra en la figura 2.22,

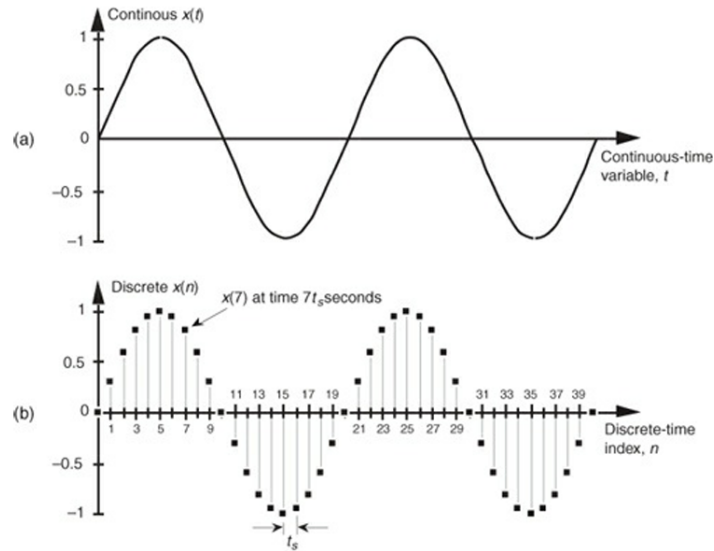


Figura 2.22: Diferencia señal en tiempo continuo y en tiempo discreto
.Tomado de [24]

en la cual se grafica una señal descrita por la ecuación $x(t) = \sin(2\pi f_o t)$, donde f_o es la frecuencia de la señal, la cual se mide en Hertz (Hz), los cuales son el número de ciclos realizados por la señal en un segundo. La variable t representa el tiempo en segundos, el término $f_o t$ tiene dimensiones de ciclos, y el término completo $2\pi f_o t$ es un ángulo medido en radianes. Al trazar la ecuación, se obtiene una curva de onda sinusoidal continua, como se ve en la gráfica (a). Si esta onda sinusoidal continua representa un voltaje físico, este puede ser muestreado una vez cada t_s segundos utilizando un convertidor analógico digital y representar la onda sinusoidal como una secuencia de valores discretos. Trazar esos valores individuales como puntos da una forma de onda discreta, como se la de la gráfica (b). Por lo tanto, la gráfica (b) es la versión en tiempo discreto de la gráfica (a), donde la variable independiente t en la ecuación y en la gráfica (a) es continua. El índice de la variable independiente n en la gráfica (b) es discreta y solo puede tener valores enteros. Es decir, el índice n se utiliza para identificar los elementos individuales de la secuencia discreta en la gráfica (b) [24].

2.8.2. Procesamiento digital de señales

El procesamiento digital de señales (DSP, por sus siglas en inglés) se distingue de otras áreas de la informática por el tipo único de datos que utiliza, que son las señales. En la mayoría de los casos, estas señales se originan como datos sensoriales del mundo real. El DSP es la matemática, los algoritmos y técnicas utilizados para manipular estas señales después de que han sido convertidas a forma digital, en términos simples, es un sistema discreto. Esto incluye una

variedad de objetivos, como la mejora de imágenes visuales, reconocimiento, comprensión de datos para almacenamiento, etc.[19].

Un sistema discreto es una colección de componentes de hardware y rutinas de software que operan en una secuencia de señales de tiempo discreto. La diferencia fundamental entre la forma en que se representa el tiempo en sistemas continuos y discretos da lugar a una diferencia muy importante en cómo se caracteriza la frecuencia en sistemas continuos y discretos. Si se tiene una señal eléctrica como la que se representa en la gráfica (a) de la figura 2.22, se podría medir su frecuencia aplicándolo a un osciloscopio, sin embargo, si simplemente se nos diera una lista de valores de la ecuación que representa el comportamiento de la señal, y se quisiera determinar la frecuencia de la forma de onda, sería algo imposible de hacer sin conocer el valor de t_s , que es el periodo de muestreo. Si se conoce el valor del periodo de muestreo, se puede determinar la frecuencia absoluta de la onda sinusoidal discreta. suponiendo que t_s es 0.05 milisegundos/muestra, y al ver la grafica (b) sabemos que la onda sinusoidal se repite cada 20 muestras, el periodo de la onda sinusoidal es:

$$PeriodoOnda = \frac{20muestras}{periodo} * \frac{0,05milisegundos}{muestras} = 1milisegundo \quad (2.16)$$

debido a que la frecuencia de una onda sinusoidal es el recíproco de su periodo, ahora sabemos que la frecuencia absoluta de la onda (f_s) es $1/t_s$, o $1/(1ms)$ que da como resultado $1kHz$. Al tratar con sistemas discretos, la determinación de la frecuencia absoluta en Hz depende de la frecuencia de muestreo [18].

2.8.3. Convertidor Anológico Digital

La mayoría de las señales que se encuentran directamente en la ciencia y la ingeniería son continuas, las cuales son captadas mediante el uso de sensores para convierte esas señales física en señales eléctricas. La conversión de analógico a digital (ADC, por sus siglas en ingles) y la conversión de digital a analógico (DAC, por sus siglas en ingles) son los procesos que permiten a las computadoras digitales interactuar con estas con estas señales cotidianas. La información digital es diferente de su contraparte continua en dos aspectos importantes, se muestrea y se cuantiza, y ambos limitan cuánta información puede contener una señal digital [19].

El proceso de digitalización se divide en dos etapas principales, la etapa de muestreo y retención (Sample and Hold, S/H), y el convertidor analógico digital (ADC), como se ve en la figura 2.23. El S/H es necesario para mantener constante el voltaje que ingresa al ADC mientras se lleva a cabo la conversión. Sin embargo, la salida del S/H solo puede cambiar en intervalos periódicos, momento en el cual se hace idéntico al valor instantáneo de la señal de entrada. Los cambios en la señal de entrada que ocurren entre estos tiempos de muestreo se ignoran por completo. Es decir, el muestreo convierte la variable independiente de continua a discreta. Así, los valores que salen del S/H, y que entran al ADC generan un valor digital en la salida del ADC. Dependiendo de la resolución del

ADC, este tendrá un rango de números enteros asignados a ciertos niveles de voltaje de entrada, a esto se le conoce como cuantización. Si se tiene un ADC que tiene una resolución de 12 bits, puede producir valores entre 0 y 4095. Pero se debe tener en cuenta que la cuantización introduce un error en la señal digitalizada, por ejemplo, tanto 2.56000 volts como 2.56001 volts se convertirán en el número digital 2560, debido a que solo se pueden asignar valores entre 0 y 4095. Cualquier muestra en la señal digitalizada puede tener un error máximo de $\pm 1/2$ LSB (Bit menos significativo, jerga para la distancia entre niveles de cuantización adyacentes), esto se puede ver en la gráfica de la gráfica d) de la figura 2.23, la cual se obtiene de restar la gráfica b) de la gráfica c). En otras palabras, la salida digital c) es equivalente a la entrada continua b), más un error de cuantización d) [19].

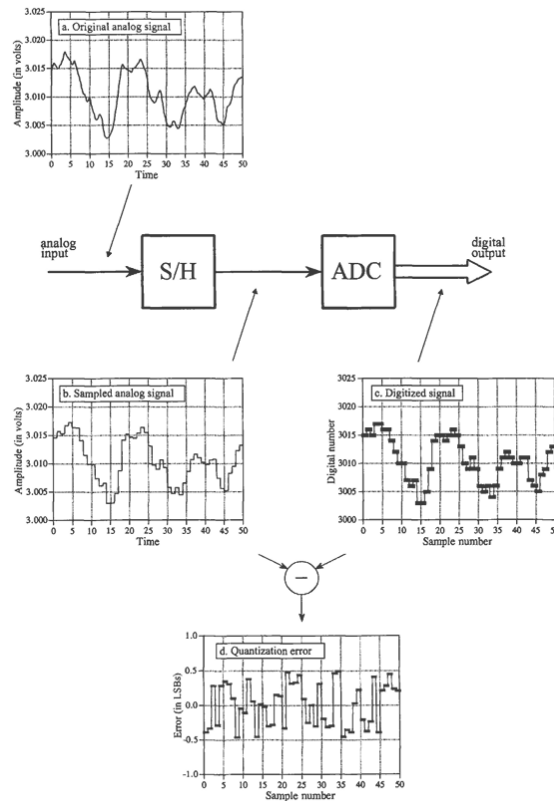


Figura 2.23: Proceso de digitalización de una señal . Tomado de [19].

2.8.4. Teorema de muestreo

Se considera que el muestreo de una señal analógica se ha realizado correctamente si se puede reconstruir la señal analógica a partir de las muestras. Incluso si los datos muestreados parecen confusos o incompletos, se sabe que la información clave se ha capturado si se puede revertir el proceso. El teorema de muestreo, frecuentemente se le llama teorema de muestreo de Shannon, o teorema de muestreo de Nyquist, en honor a los autores de los documentos sobre el tema en la década de 1940. El teorema de muestreo indica que una señal continua puede ser muestreada correctamente, solo si no contiene componentes de frecuencia por encima de la mitad de la tasa de muestreo. Por ejemplo, una tasa de muestreo de 2000 muestras por segundo requiere que la señal analógica esté compuesta por frecuencias por debajo de 1000 ciclos/segundo. Si hay frecuencias por encima de este límite presentes en la señal, serán aisladas a frecuencias entre 0 y 1000 ciclos por segundo, combinándose con cualquier información que legítimamente estuviera ahí [19].

Dos términos se utilizan ampliamente al hablar del teorema de muestreo, la frecuencia de Nyquist y la tasa de Nyquist. Desafortunadamente, su significado no está estandarizado, pero sus usos más comunes son para indicar que son la mitad de la frecuencia de muestreo. Por ejemplo, si se considera una señal compuesta por frecuencias entre corriente continua (DC) y 3kHz, para digitalizar correctamente esta señal debe hacer a 6kHz o más. Suponiendo que se elige muestrear a 8kHz, esto nos permite representar correctamente frecuencias en DC y 4kHz. En esta situación hay cuatro frecuencias importantes, la frecuencia más alta en la señal (3kHz), el doble de esta frecuencia (6kHz), la frecuencia de muestreo (8kHz) y la mitad de la frecuencia de muestreo. Según la definición más común, la frecuencia y tasa de Nyquist es la mitad de la frecuencia de muestreo, que es 4kHz, o 4000 muestras por segundo [19].

2.8.5. Filtros digitales

Los filtros digitales son una parte muy importante para los DSP, volviéndolos populares debido a su rendimiento extraordinario. Los filtros tienen dos usos principales, separación de señales y restauración de señales. La separación de señales es necesaria cuando una señal ha sido contaminada por interferencias, ruidos u otras señales. Se podría utilizar un filtro para separar estas señales para que puedan ser analizadas individualmente. La restauración de señales se utiliza cuando una señal ha sido distorsionada de alguna manera. Estos problemas se pueden abordar con filtros analógicos o digitales. Los filtros analógicos son económicos, rápidos o digitales. Los filtros digitales, en comparación, son muy superiores en el nivel de rendimiento que se puede alcanzar. Los filtros digitales pueden lograr rendimientos mil veces mejor que los filtros analógicos. Esto hace una diferencia dramática en cómo se abordan los problemas de filtrado. Con los filtros analógicos, la énfasis está en manejar las limitaciones de la electrónica, como la precisión y estabilidad de los componentes. En comparación, los filtros digitales son tan buenos que el rendimiento del filtro se ignora con frecuencia.

La énfasis se desplaza hacia limitaciones de las señales y los problemas teóricos relacionados con su procesamiento [19].

Actualmente existen una gran cantidad de técnicas de filtrado digital, cuyo uso dependerá de las necesidades particulares de cada caso, lo que significa que un mismo tipo de filtrado digital será útil para todos los casos. Encontrar la técnica adecuada requiere de largos procesos de experimentación, lo que implica tener que probar varias técnicas de filtrado hasta encontrar alguna que cumpla con los objetivos.

La parte más importante de cualquier tarea de un DSP es entender cómo está contenida la información de las señales con las que está trabajando. Las dos formas comunes en que la información se representa en señales que ocurren de manera natural son la representación en el dominio del tiempo y representación en el dominio de la frecuencia [19].

El dominio del tiempo describe cuándo algo ocurre y cuál es su amplitud. Cada muestra en la señal indica lo que está sucediendo en ese instante y el nivel del evento. Cada muestra contiene información que se puede interpretar sin referencia a ninguna otra muestra. Incluso si solo se tiene una muestra de esta señal, aún se sabe algo sobre lo que se está midiendo. Esta es la forma más sencilla de que la información esté contenida en una señal. En contraste, en el dominio de la frecuencia es más indirecto. Muchas cosas en nuestro universo muestran movimiento periódico, y al medir la frecuencia, fase y amplitud de este movimiento periódico, a menudo se puede obtener información sobre el sistema que produce el movimiento. Una sola muestra, por sí misma, no contiene información sobre el movimiento periódico y, por lo tanto, no contiene información sobre el sistema que lo produce. La información está contenida en la relación entre muchos puntos en la señal. Esto nos lleva a la importancia de la respuesta al escalón y la respuesta de frecuencia. La respuesta al escalón describe cómo la información representada en el dominio del tiempo está siendo modificada por el sistema. En contraste, la respuesta de frecuencia muestra cómo la información representada en el dominio de la frecuencia está siendo cambiada. Esta distinción es absolutamente crítica en el diseño de filtros porque no es posible optimizar un filtro para ambas aplicaciones. Un buen rendimiento en el dominio del tiempo resulta en un rendimiento pobre en el dominio de la frecuencia, y viceversa. Pasar de un dominio a otro es bastante sencillo, utilizando el algoritmo de la transformada rápida de Fourier (FFT, por sus siglas en inglés), la cual permite descomponer una señal como una combinación de ondas senoidales de diferentes frecuencias (los cuales se conocen como armónicos), lo que permite saber que frecuencias forman una señal y en qué grado están presentes conociendo su amplitud, esto se puede ver ilustrado en la figura, la cual grafica una respuesta al impulso, una respuesta al escalón y una respuesta en frecuencia de un filtro. La respuesta de frecuencia puede llegar a ser expresada en decibelios como se ve en la gráfica d) [19].

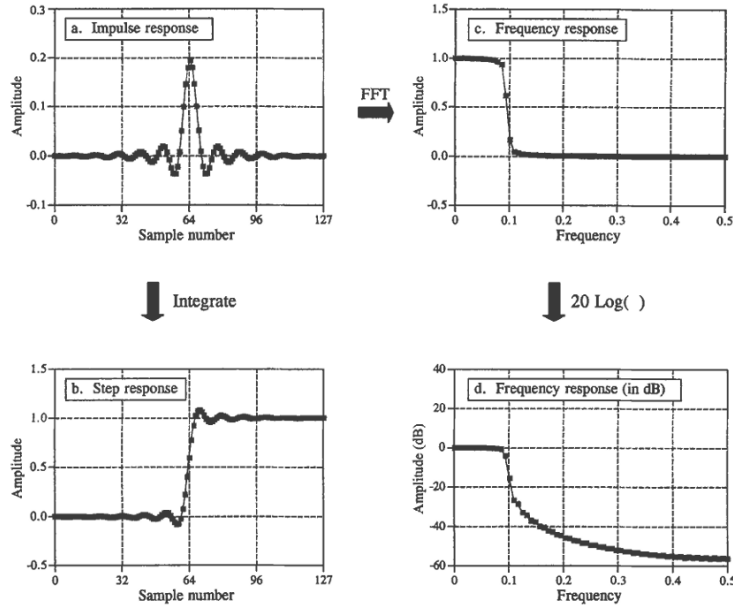


Figura 2.24: Respuesta dominio del tiempo y frecuencia
.Tomado de [19]

Los filtros en el dominio de la frecuencia se utilizan generalmente para permitir ciertas frecuencias (la banda de paso), mientras bloquean otras (la banda de rechazo). Cuatro respuestas son las más comunes, pasa bajas, pasa altas, pasa banda y rechaza banda. Los filtros pasa bajas y pasa altas trabajan considerando una frecuencia específica (frecuencia de corte), y es a partir de esta que se permitirá el paso de las frecuencias que estén por debajo o por arriba de esta, según sea el caso. Los filtros pasa banda y rechaza banda tienen dos frecuencias de corte, una inferior y otra superior, lo cual se utilizan para delimitar a un conjunto de frecuencias específicas (ancho de banda), y dependiendo del filtro, se dejará pasar o se rechazarán todas esas frecuencias. En la figura 2.25 se puede apreciar la respuesta en frecuencia de mencionadas de los filtros [19].

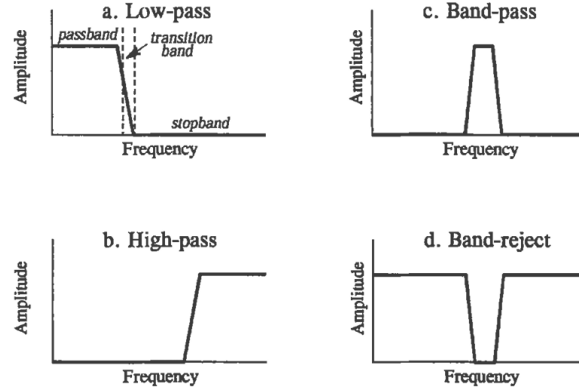


Figura 2.25: Respuesta en frecuencia filtros . Tomado de [19].

2.8.6. Savitzky-Golay

Un uso obvio para los filtros pasa bajas es suavizar datos. La premisa del suavizado de datos es que se está midiendo una variable que varía lentamente y que también está corrupta por ruido aleatorio. Entonces, a veces puede ser útil reemplazar cada punto de datos por algún tipo de promedio local de los puntos de datos circundantes. Dado que los puntos cercanos tienen un valor muy parecido al mismo valor subyacente, promediar puede reducir el nivel de ruido sin sesgar (demasiado) el valor obtenido. El suavizado de datos está probablemente más justificado cuando se usa simplemente como una técnica gráfica, para guiar la vista a través de un bosque de puntos de datos (todos con grandes barras de error), o como un medio para hacer estimaciones iniciales aproximadas de parámetros simples a partir de un gráfico [25].

Un tipo particular de filtro pasa bajas, bien adaptado para el suavizado de datos son los filtros Savitzky-Golay, también conocidos como mínimos cuadrados, o polinomio de suavizado digital (DISPO, por sus siglas en ingles). En lugar de tener sus propiedades definidas en el dominio de la frecuencia y luego traducirlas al dominio del tiempo, los filtros Savitzky-Golay derivan directamente de una formulación particular del problema de suavizado de datos en el dominio del tiempo. Los filtros Savitzky-Golay fueron inicialmente (y todavía se utilizan a menudo) para hacer visibles los anchos y alturas relativos de las líneas espectrales en datos espectrométricos ruidosos [25].

Un filtro digital se aplica a una serie de valores de datos espaciados equitativamente $f_i \equiv f(t_i)$, donde $t_i \equiv t_0 + i\Delta$ para un espacio de muestra constante Δ y $i = \dots -2, -1, 0, 1, 2, \dots$. El tipo más simple de filtro digital (el filtro de respuesta impulsiva finita o no recursivo) sustituye cada valor de datos f_i por una combinación lineal g_i de sí mismos y algunos de los números cercanos vecinos, como se ve en la siguiente ecuación:

$$g_i = \sum_{n=-n_L}^{n_R} C_n f_i + n \quad (2.17)$$

Aquí n_L es el número de puntos utilizados a la izquierda de un punto de datos i , es decir, ante que él, mientras que n_R es el número utilizado a la derecha, es decir después. Un llamado filtro casual tendría $n_R = 0$ [25].

Como punto de partida para comprender los filtros Savitzky-Golay se considera el procedimiento de promediado más simple posible. Para un valor fijo de $n_L = n_R$ se calcula cada g_i como el promedio de los puntos de datos desde f_{i-n_L} hasta f_{i+n_R} . Esto a veces se llama promediado en ventana móvil y corresponde a la ecuación 2.17 con una constante $C_n = 1/(n_L + n_R + 1)$. Si la función subyacente es constante, o cambia linealmente con el tiempo (ya sea en aumento o disminución), entonces no se introduce sesgo en el resultado. Los valores más altos en un extremo del intervalo de promediado son, en promedio, equilibrados por valores más bajos en el otro extremo. Sin embargo, se introduce un sesgo si la función subyacente tiene una segunda derivada distinta a cero. En aplicaciones espectrométricas, una línea espectral estrecha ve reducida su altura y aumentada su anchura. Dado que estos parámetros son de interés físico, el sesgo introducido resulta claramente indeseable.[25]

La idea del filtrado Savitzky-Golay es encontrar coeficientes de filtro C_n que preserven momentos mayores. Equivalentemente, la idea es aproximar la función subyacente dentro de la ventana móvil no con una constante (cuyo estimado es el promedio), sino con un polinomio de mayor orden, típicamente cuadrático o cuártico. Para cada punto f_i , se ajusta por mínimos cuadrados un polinomio a los $n_L + n_R + 1$ puntos dentro de la ventana móvil, y luego se asigna a g_i el valor de ese polinomio en la posición i . No se utiliza el valor del polinomio en ningún otro punto. Cuando se pasa al siguiente punto f_{i+1} , se realiza un nuevo ajuste por mínimos cuadrados usando una ventana desplazada.[25]

2.8.7. Wavelets

Otra buena herramienta utilizada para el procesamiento digital de señales son las wavelets, debido a que la manera en la que pueden eliminar el ruido de una señal se caracteriza por estar en el dominio de la transformación (como también lo es la FFT) en lugar de estar en el dominio de la señal original(tiempo o espacio) [26].

Una wavelet es una oscilación similar a una onda con una amplitud que comienza en cero, aumenta y luego disminuye nuevamente a cero. A diferencia de los senos utilizados en la transformada de Fourier para descomposición de una señal, las wavelets son generalmente muchos más concentradas en el tiempo. Por lo general, proporcionan un análisis de la señal que está localizado tanto en el tiempo como en la frecuencia, mientras que la transformada de Fourier está localizada solo en la frecuencia. Un ejemplo de wavelet está en la figura 2.26, la cual es la wavelet Coiflet [26].

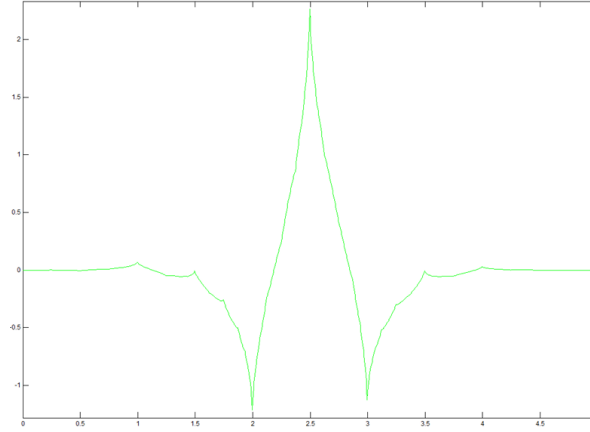


Figura 2.26: Wavelet Coiflet
. Tomado de [26].

Dada una wavelet madre $\psi(t)$ (que puede considerarse simplemente como una función base de L^2), la transformada continua de wavelet (CWT, por sus siglas en ingles) de una función $x(t)$ (suponiendo que x pertenece a L^2) se define como:

$$X(a, b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} \psi\left(\frac{t-b}{a}\right) x(t) dt \quad (2.18)$$

El parámetro de escala o dilatación a corresponde a la información de frecuencia (a frecuencia⁻¹), y el parámetro de traslación b se relaciona con la ubicación de la función wavelet a medida que se desplaza a través de la señal, por lo que corresponde a la información de tiempo en la transformación. La integral en 2.18 puede verse como una operación de convolución de la señal y una función base $\psi(t)$ (hasta dilataciones y traducciones). Debe enfatizarse que, a diferencia de la transformada de Fourier en la que la función base es $e^{i\omega t}$, hay muchas (de hecho, infinitas) elecciones posibles de $\psi(t)$. [26]

En la práctica, la transformada que se utiliza es la transformada wavelet discreta (DWT, por sus siglas en ingles) que transforma señales discretas en coeficientes discretos en el dominio de wavelets. Esta transformada es esencialmente una versión muestreada de CWT. En lugar de trabajar con $a, b \in \mathbf{R}$, los valores de $X(a, b)$ son calculados sobre una malla discreta:

$$a = 2^{-j}, b = k * 2^{-j}, j, k \in \mathbf{R} \quad (2.19)$$

donde este tipo de discretización se llama dilatación dyática y posición dyática, respectivamente. [26]

En el caso de la DWT, asumiendo que la longitud de la señal satisface $N = 2^j$ para algún j positivo, la transformada se puede calcular de manera eficiente algoritmo de Mallat, el cual tiene complejidad de $O(N)$. Esencialmente, el algoritmo

es un esquema jerárquico rápido para derivar los productos internos requeridos (de la ecuación 2.18, como una función de a y b) utilizando un conjunto de filtros pasa bajas y pasa altas consecutivos, seguido de un diezmado. Esto resulta en una descomposición de la señal en diferentes escalas que pueden considerarse como diferentes bandas de frecuencia. Los filtros pasa bajas y pasa altas utilizados en este algoritmo se determinan según la wavelet madre que se este utilizando. Las salidas de los filtros pasa bajas se conocen como coeficientes de aproximación y las salidas de los filtros pasa altas se conocen como coeficientes de detalle [26].

En cada nivel de descomposición, los filtros producen señales que abarcan solo la mitad de la banda de frecuencias. Esto duplica la resolución en frecuencia, ya que la incertidumbre en frecuencia se reduce a la mitad. Por lo tanto, la salida de cada filtro puede ser muestreada a menor frecuencia por un factor de 2, de acuerdo con el teorema de Nyquist. Usando este enfoque, la resolución temporal se vuelve arbitrariamente buena en altas frecuencias, mientras que la resolución en frecuencia se vuelve arbitrariamente buena en bajas frecuencia. El número de niveles de descomposición debe ser determinado por el usuario (según las características de la señal a analizar). Un número típico es de 3 a 5. La DWT de la señal original se obtiene concatenando todos los coeficiente comenzando desde el último nivel de descomposición. El algoritmo Mallat se ve ilustrado en la figura 2.27 [26].

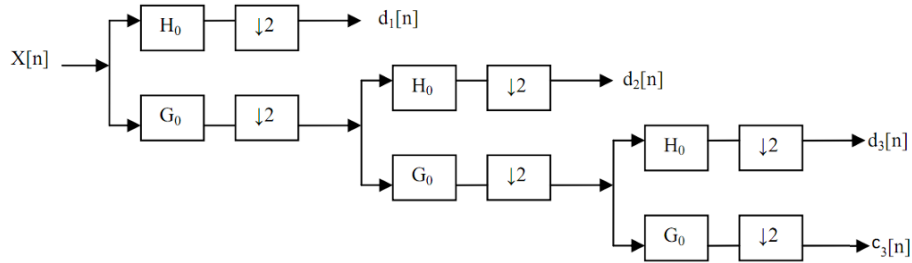


Figura 2.27: Algoritmo Mallat
. Tomado de [26].

2.9. Interpolación lineal

Una interpolación es el procedimiento mediante el cual, a partir de un conjunto de datos conocidos (puntos experimentales), se estima el valor de una variable dependiente para un valor intermedio de la variable independiente, sin necesidad de conocer la ecuación exacta que describe todo el fenómeno. Uno de los principales errores que se puede cometer es ajustarse a una ecuación y utilizar esta ecuación para interpolar otros puntos. Estos ajustes a líneas de tendencia en ocasiones proporcionarán aproximaciones adecuadas, pero en otros muchos casos, aún dando una buena estimación de la función en los puntos experimen-

tales usados para su cálculo, proporcionarán valores erróneos y en otros casos imposibles de acuerdo con el proceso estudiado [27].

La interpolación lineal es la forma más sencilla de interpolación que existe. Usa dos puntos con el fin de desarrollar una aproximación lineal de la función. Estos puntos usados en el proceso serán los más próximos al punto de interés, teniendo que ser uno menor y otro mayor que este. Por tanto, entre cada pareja de valores de (X_k, Y_k) y (X_{k+1}, Y_{k+1}) , se calcula la recta existente entre ambos para calcular Y al valor de X dado. El valor estimado esta dado por [27]:

$$Y = Y_k + \frac{Y_{k+1} - Y_k}{X_{k+1} - X_k}(X - X_k) \quad (2.20)$$

2.10. Python

El software libre se ha convertido en uno de los movimientos tecnológicos de mayor auge en el siglo XXI. Para su desarrollo ha sido necesario contar con un grupo de herramientas que hagan óptima su utilización y sean fáciles de aprender. Python es un lenguaje de programación que cumple con lo planteado que fue creado por Guido van Rossum, un programador holandés a finales de los 80's y principios de los 90's, y se ha perfilado como una opción recomendada para el desarrollo de software libre. El software libre es un movimiento que proclama el acceso al código fuente de un programa, lo que admite ser libre de uso, ejecución distribución y modificación. Es decir, el nuevo software creado bajo este concepto podría emplearse para cualquier fin, ejecutarse en cualquier ambiente y modificarse de ser necesario [28].

Debido a su estructura sencilla y a la gran disponibilidad de de herramientas y paquetes para el procesamiento y visualización de datos y señales, Python se a convertido en una opción viable para el desarrollo científico, emulando en muchas ocasiones las funciones disponibles en otros lenguajes de corte científico ya existes como lo es Matlab [28].

2.11. Conclusiones

En este capítulo se explicó todos los conceptos fundamentales utilizado en el desarrollo de esta tesis, poniendo en contexto al lector con el objetivo de que este sea capaz de entender todo el trabajo desarrollado, y que será presentado en los capítulos siguientes.

Capítulo 3

Desarrollo del prototipo

3.1. Introducción

En este capítulo se describen los pasos que se llevaron a cabo para desarrollar la parte del hardware y software para la construcción del espectrómetro que trabaje en el rango visible.

3.2. Hardware

3.2.1. Optical bench

De la sección 2.7.1 se puede resaltar que un espectrómetro de absorción consta principalmente de cinco componentes, la fuente de luz, el monocromador, los componentes ópticos, la muestra y el detector. Desarrollar un espectrómetro con esa configuración generaría varias limitaciones, principalmente por la implementación del monocromador, ya que, como se explico en la sección 2.7.1, el objetivo de este componente es permitir el paso de una longitud de onda específica hacia el espectrómetro, lo que implica que el tiempo que tomaría en captarse un espectro de una banda más ancha, por ejemplo todo el rango visible, sería demasiado, sin mencionar que esa configuración no es completamente compatible con otras espectroscopías, y que el equipo se volvería más grande y de difícil transporte.

En la sección 2.7.5 se explora una alternativa a la configuración convencional de los espectrómetros, la configuración Czerny-Turner. Esta tiene la ventaja de no necesitar de un monocromador, ya que realiza la dispersión de todas las longitudes de onda la fuente de luz mediante el uso de una rejilla de difracción, para después ser redirigida mediante el uso de espejos, hacia un sensor lineal. Esto representa una ventaja, ya que permite realizar mediciones con todo el espectro de la fuente de luz en menor tiempo, y sin tener que seleccionar bandas específicas para excitar a la muestra como se haría en un diseño que implemente un monocromador. Otra ventaja que presenta esta configuración es que

pueden encontrarse ya implementados como sistemas ópticos de tamaños bastante pequeños, lo que facilita su portabilidad y manejo. Estos sistemas ópticos son conocidos como optical bench (banco óptico en español), los cuales constan de componentes ópticos como lentes, espejos, prismas, etc., y pueden estar implementados en bases rígidas con soportes móviles que permiten realizar ajustes de los componentes.

En el mercado existen muchas opciones de espectrómetros que implementan configuraciones Czerny-Turner, sin embargo, encontrar solamente el optical bench es algo complicado. Las tiendas en línea tienen disponibles varias opciones de optical bench que fueron extraídos de espectrómetros, los cuales varían en fabricante, diseño, precio y como es que se comportan los componentes ópticos según la longitud de onda de la luz incidente.

Se optó por utilizar un optical bench de la marca *B&W Tek*, el cual cuenta con una rejilla de difracción de 1800 líneas por milímetro, una longitud de onda de resplandor de 500nm y 2 espejos de enfoque [29].

La siguiente imagen muestra al optical bench por dentro:

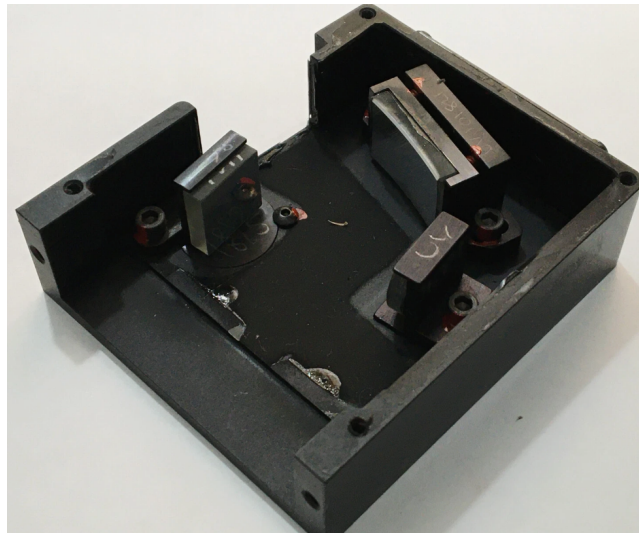


Figura 3.1: Optical bench Czerny-Turner
. Tomado de [29].

Como se puede apreciar, todo los componentes ópticos están montados sobre una estructura metálica de no más de 6cm de ancho, 6.6cm de largo y 2.2cm de altura, la cual cuenta con un conector de fibra óptica y una cubierta superior desmontable, y una apertura para la colocación de un sensor de imagen lineal, además de que cada componente óptico cuenta con tornillos que permiten manipular su posición, lo cual es necesario para poder calibrar el espectrómetro y alinear la luz correctamente con el sensor.

La figura muestra la curva de eficiencia de la rejilla de difracción implemen-

tada en el optical bench, la cual tiene una respuesta promedio superior al 50 % para el rango visible.

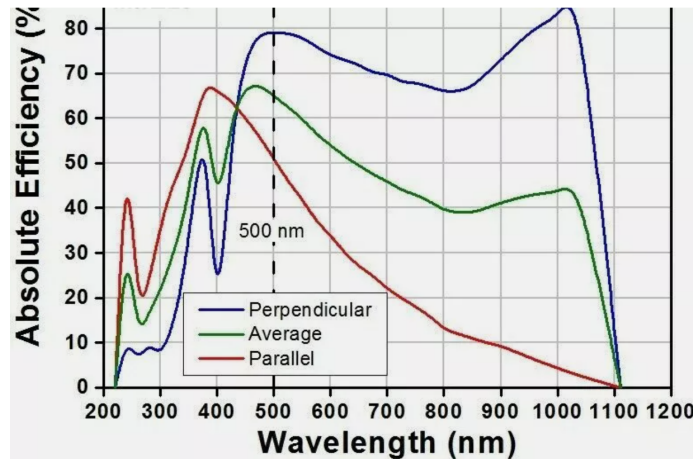


Figura 3.2: Curva de eficiencia optical bench Czerny-Turner . Tomado de [29].

3.2.2. Detector

La elección del detector es uno de los pasos más importantes para el desarrollo de un espectrómetro, ya que existen varios tipos de sensores, como los expuestos en la sección 2.7.6, cuya diferencia principal está en la manera en que estos trabajan, además de contar con diferentes niveles de sensibilidad a longitudes de onda específicas, tamaños y formas.

Para nuestro caso particular, se tuvo que buscar una opción que fuera compatible con el optical bench que se eligió. La primera limitación que eso implica es que las dimensiones del sensor no deben exceder las de 5x2cm, que son las medidas de la apertura del optical bench, además de que el sensor no debe contar con alguna forma de identificar colores, como la identificación del RGB (siglas de rojo, verde y azul en ingles), ya que el propósito del sensor es identificar la intensidad de la luz difractada, para después identificar la longitud de onda revisando en que zona del sensor incidió la luz, como se explica en la sección 2.7.5.

Se optó por buscar un sensor lineal CCD que se adaptara a las necesidades del espectrómetro, ya que en este tipo de tecnología es bastante común encontrar sensores lineales. El uso de los sensores CCD estaba principalmente orientado a su implementación en cámaras digitales y escáneres de imágenes, pero con la aparición de los sensores CMOS (los cuales normalmente se encuentran como una matriz de píxeles en lugar de un arreglo lineal) su uso se ha visto reducido, y en consecuencia varias empresas han descontinuado la fabricación de sensores CCD. Ejemplo de esto es el sensor que se consideró como primera opción para

el espectrómetro, el Sony ILX511, a continuación se muestra una ilustración del mismo:

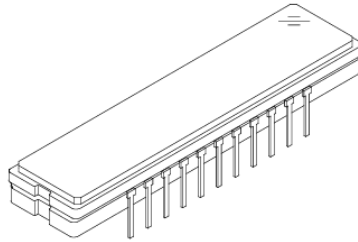


Figura 3.3: Sensor Sony ILX511
. Tomado de [30].

Este es un sensor CCD lineal de imagen de 2048 píxeles. Este sensor contaba con dimensiones de 4.16x1cm, además de tener una buena sensibilidad para el rango visible, como se ve en la figura 3.4 [30].

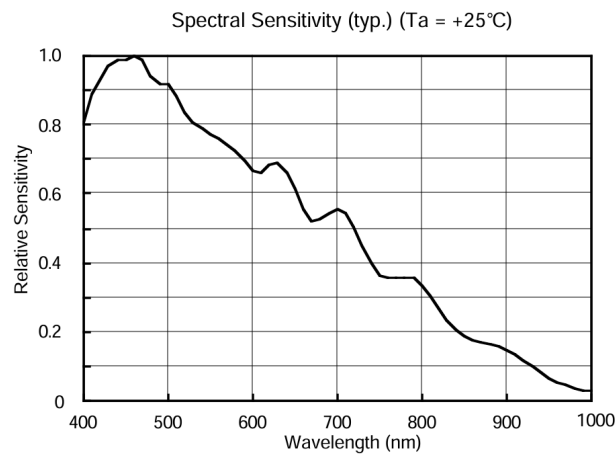


Figura 3.4: Gráfica de sensibilidad sensor Sony ILX511
. Tomado de [30].

La empresa Hamamatsu cuenta con varias opciones de sensores lineales, ya que cuenta con toda una línea especializada en espectrofotometría, donde se implementa las tecnologías CCD y CMOS para la fabricación de sensores de imagen lineales, con diferentes propiedades especiales para cada modelo, como el tamaño, forma y el tipo de espectroscopía para el que están diseñados. Un ejemplo de eso es el sensor S10122-512Q-01, el cual se muestra en la figura 3.5.

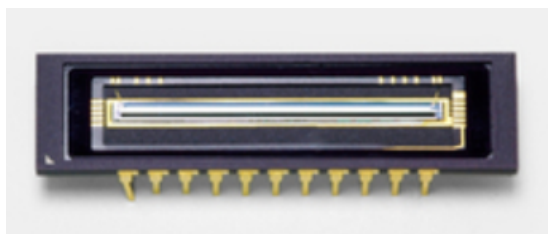


Figura 3.5: Sensor Hamamatsu S10122-512Q-01
. Tomado de [31].

Este cuenta con 512 píxeles, un rango de respuesta espectral de 200 a 1000nm (como se ve en la figura 3.6), y un tamaño de imagen de 25.6x0.5mm, lo que lo vuelve una opción viable para ser implementada en espectrómetro. Desafortunadamente, el costo por uno de estos sensores es bastante elevado al ser componentes profesionales para espectroscopía, sin mencionar los costos de importación [31].

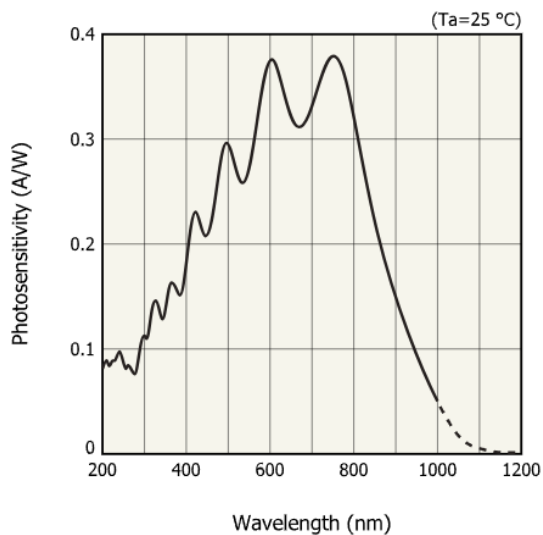


Figura 3.6: Respuesta espectral S10122-512Q-01
. Tomado de [31].

La opción que se seleccionó fue el TCD1304DG de Toshiba, el cual es un sensor de imagen lineal CCD de 3648 píxeles, el cual se puede apreciar en la figura 3.7.



Figura 3.7: Sensor Toshiba TCD1304DG
. Tomado de [32].

A pesar de ser un sensor de uso más generalizado, y no específico para la espectroscopía, cumple con los requisitos para ser implementado en el espectrómetro. Sus dimensiones son de 4.16x0.965cm, solo requiere de tres señales digitales para su manejo, lo que lo vuelve de fácil uso, puede trabajar con un voltaje de alimentación mínimo de 3.0v, y tiene buena respuesta espectral en el rango de la luz visible como se puede apreciar en la figura 3.8, además de tener un precio por unidad de \$56.10 USD, convirtiéndose en la mejor opción para utilizar en el espectrómetro.

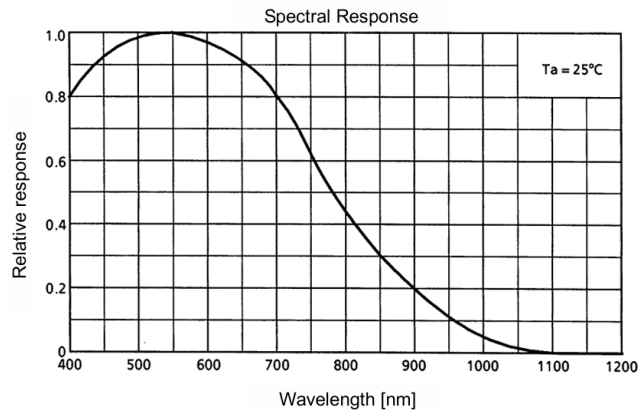


Figura 3.8: Respuesta espectral TCD1304DG
. Tomado de [32].

3.2.3. Microcontrolador

Al igual que el sensor, la elección del microcontrolador es bastante importante, ya que es a través de este que se controlará el sensor, se capturarán los datos y se enviarán a la etapa de procesamiento de los mismos.

El sensor TCD1304DG cuenta con tres pines de control, master clock (θM), shift gate (SH) e integration clear gate (ICG), a los cuales se les debe conectar pulsos de voltaje con periodos de tiempo específicos para obtener en el pin output signal (OS) una señal analógica correspondiente a los niveles de intensidad captado por los pixeles. Tomando esto en cuenta, el microcontrolador debe ser capaz de generar pulsos de voltaje a diferentes frecuencias, esta característica se

conoce como modulación por ancho de pulso (PWM, por sus siglas en ingles). También, el microcontrolador debe se capaz de captar la señal generada por el sensor, por lo que debe contar con un ADC, el cual se encarga de digitalizar las señales captadas, como se explicó en la sección 2.8.3. Como último requisito, el microcontrolador debe contar con lo que se conoce como protocolos de comunicación, los cuales permiten enviar datos digitales entre diferentes dispositivos mediante conexiones físicas, lo cual es indispensable para el desarrollo del espectrómetro ya que se pretende enviar los datos hacia otro dispositivo, como una computadora. Algunos ejemplos de protocolos de comunicación son el UART, I2C, SPI, RS232, etc.

En el mercado existen muchas opciones de microcontroladores, con diferentes capacidades, arquitecturas y características. La opción que se seleccionó fue el microcontrolador STM32F401ccu6, fabricado por STMicroelectronics. Este se basa en núcleo RISC de 32 bits ARM Cortex-M4 de alto rendimiento que funciona a una frecuencia de hasta 84 MHz. Este cuenta con unidad de punto flotante (FPU, por sus siglas en ingles) de precisión simple que admite todas las instrucciones de procesamiento de datos y tipos de datos de precisión simple ARM. De todos los periféricos con los que cuenta, se puede destacar un ADC de 12 bits, conectividad I2C, SPI, USB 2.0 y USART, acceso directo a memoria (DMA, por sus siglas en ingles) de propósito general y 11 temporizadores, útiles para la generación de PWM e interrupciones, entre muchas cosas más que se pueden apreciar en la figura 3.9 [33].

System	ART Accelerator™	Connectivity
Power supply 1.2V internal regulator POR/PDR/PVD/BOR	Arm® Cortex®-M4 CPU 84 MHz	3x I²C
Xtal oscillators 32 kHz + 4 ~26 MHz	Floating Point Unit (FPU)	4x SPI (2x with I²S)
Internal RC oscillators 32 kHz + 16 MHz	Nested vector interrupt controller (NVIC)	SDIO
PLL	JTAG/SW debug	USB 2.0 OTG FS
Clock control	Embedded trace macrocell (ETM)	3x USART LIN, smartcard, IrDA, Modem control
RTC/AWU	Memory protection unit (MPU)	
2x watchdogs (independent + window)	AHB-Lite bus matrix	Analog
36/50/81 I/Os	APB bus	1x 12-bit ADC 2.4 MSPS 16 channels / 0.41 µs
Cyclic redundancy check (CRC)	16-channel DMA	Temperature sensor
96-bit unique ID	256-Kbyte Flash memory	
Voltage scaling	64-Kbyte SRAM	Control
	80-byte backup data	5x 16-bit timer
		1x 16-bit motor control PWM synchronized AC timer
		2x 32-bit timer

Figura 3.9: Características STM32F401CC6
. Tomado de [33].

La mayoría de este tipo de microcontroladores son fabricados en encapsulados de montaje superficial, como los que se pueden ver en la figura 3.10. Esto significa que, para poder utilizar estos microcontroladores se deben soldar en placas de circuito impreso (PCB, por sus siglas en ingles) juntos con todos los componentes mínimos necesarios para poder utilizarlos.

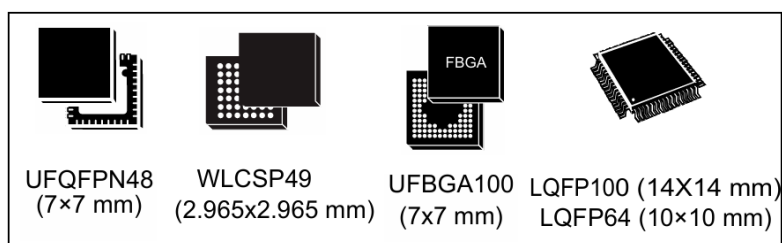


Figura 3.10: Encapsulados microcontroladores STM32F401
. Tomado de [33].

En el mercado existen muchas opciones de microcontroladores de este tipo ya implementados en lo que se conoce como tarjetas de desarrollo. Estas

tarjetas son PCB's que ya cuentan con todo el circuito necesario para que el microcontrolador funcione, además de colocar pines conectados a los puertos del microcontrolador para un fácil acceso a los periféricos y para la programación del mismo. Muchos diseños de estas tarjetas de desarrollo están pensados para su colocación en una herramienta de laboratorio conocida como protoboard, la cual permite armar circuitos electrónicos de manera rápida sin necesidad de soldar, pudiendo realizar experimentación con diferentes elementos del circuito antes de definir el diseño final. En el caso particular del STM32F401CC6, el cual se presenta en encapsulado UFQFPN48, se encuentra mayormente en el mercado implementado en una tarjeta de desarrollo conocida como BlackPill, como se puede ver en la figura 3.11.

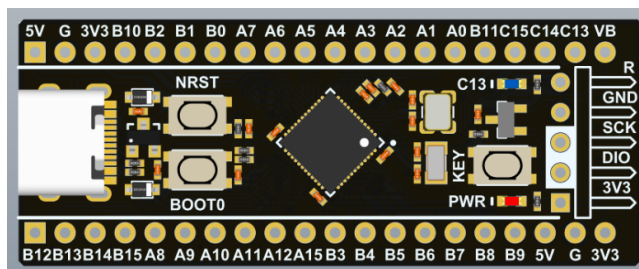


Figura 3.11: BlackPill
. Tomado de [34].

Esta tarjeta no es un producto oficial de STMicroelectronics, ya que es elaborada por WeAct Studio. Esta cuenta con todo lo necesario para utilizar el microcontrolador, además de incluir algunas cosas extra, como un conector USB tipo C, botones de reset, programación y para control del usuario, leds de encendido y de control del usuario, conexiones directas a los pines de programación serial y salidas de 5 y 3.3 volts. Las dimensiones de esta tarjeta son de 5.3x2.1 cm, lo que la vuelve lo bastante pequeña para ser utilizada sobre una protoboard, y para mantener el diseño del espectrómetro lo más pequeño y transportable posible.

3.2.4. Circuito de Control

El diseño del circuito de control tiene el objetivo de cumplir los requisitos de hardware necesarios para poder utilizar el sensor TCD1304DG. En su hoja de datos, además de mostrar las características del sensor y las gráficas de las señales que deben ser conectadas al sensor para su utilización, se muestra un esquema de un circuito típico, como se ve en la figura 3.12, el cual el fabricante recomienda implementar para el correcto funcionamiento del sensor.

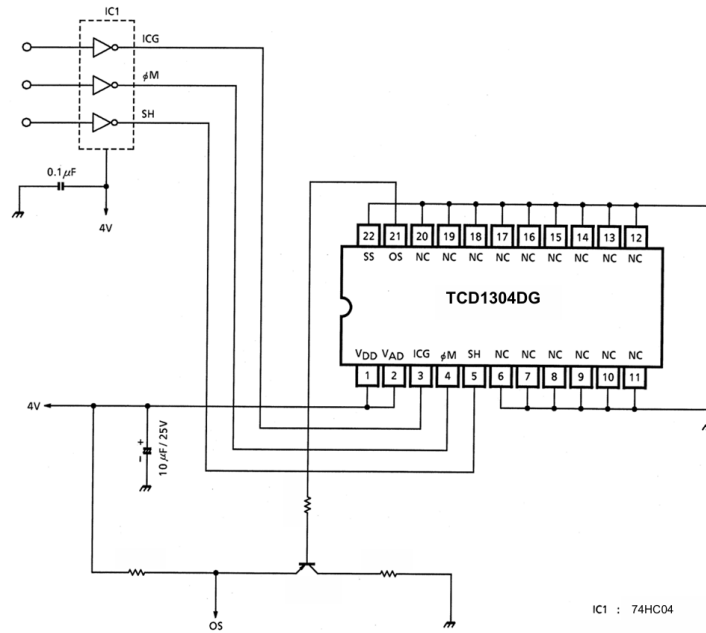


Figura 3.12: Circuito de control típico TCD1304DG
. Tomado de [32].

En el diagrama, lo primero que se puede notar es el circuito integrado TCD1304DG, que es el sensor CCD. Las terminales de la 6 a la 20, y la número 22 también, son conectadas a tierra, mientras que las terminales 1 y 2 son conectadas a 4 volts. En este circuito se colocan dos capacitores conectados entre 4 volts y tierra, uno tiene un valor de $0.1\mu F$, y el otro, que es un capacitor polarizado, tiene un valor de $10\mu F$ y tiene una tolerancia a 25 volts. La función de los capacitores es reducir el ruido proveniente de la fuente de voltaje, el cual puede llegar a afectar el funcionamiento del sensor. En el esquema hay un segundo circuito integrado, que corresponde al 74HC04, que es un conjunto de compuertas lógicas NOT, que se encargan de invertir señales digitales. Estas compuertas son útiles en caso de que el controlador que genera las señales para el sensor trabaje con un lógica opuesta a lo indicado por las gráficas de la hoja de datos. Por último, la terminal 22 del sensor, que corresponde a la salida de la señal, está conectada a un transistor PNP y a unas resistencias. El propósito de este circuito amplificador con transistor es aumentar la señal de salida para que trabaje en un rango de voltaje mayor, lo que permite aprovechar de mejor manera el rango de voltaje del ADC con el que se pretende leer la señal. Sin embargo se debe ser cuidadoso al implementar este amplificador, ya que el modelo del transistor y los valores de las resistencias no están presentes en el esquema, lo que significa que estos deben ser propuestos por el usuario para que la ganancia se adapte al ADC utilizado sin hacer que este se sature o se dañe.

A pesar de que el circuito de la figura 3.12 es funcional, hay varias co-

sas que pueden modificarse para volverlo más eficiente para el desarrollo del espectrómetro. La primera de estas modificaciones es reemplazar el circuito amplificador por un amplificador operacional (OPAM, por sus siglas en ingles) en configuración de amplificador no inversor. Un OPAM, es un circuito integrado ampliamente utilizado en el procesamiento de señales, contando con diferentes tipos de configuración que ayudan a adaptar las señales a según las necesidades del usuario. El modelo empleado para el espectrómetro es el TL082, el cual es un modelo de circuito integrado que cuenta con dos OPAMS independientes en su encapsulado, siendo bastante común y fácil de conseguir. La ganancia de un amplificador no inversor esta dada por la siguiente fórmula:

$$G = 1 + \frac{R1}{R2} \quad (3.1)$$

La señal de entrada es multiplicada por la ganancia calculada, lo que genera una señal amplificada en la salida. Para realizar el cálculo de la ganancia, idealmente se deben conocer los rangos de voltaje de la señal de entrada, y a que rangos de voltaje se quiere que llegue en la salida. En este caso particular, se midió la salida de la señal del sensor con un osciloscopio, obteniendo una señal que llegaba a un valor máximo de 2.6v aproximadamente. El ADC del microcontrolador puede leer valores máximos de 3.3v, por lo que para aprovechar mejor ese rango de voltajes, se decidió que R1 tuviera un valor de 22 kilo ohm y R2 un valor de 100 kilo ohm como se ve en la figura 3.13. Estos valores de resistencias darían una ganancia de 1.22, por lo que la señal de salida ahora llegaría a un valor máximo de 3.17v aproximadamente.

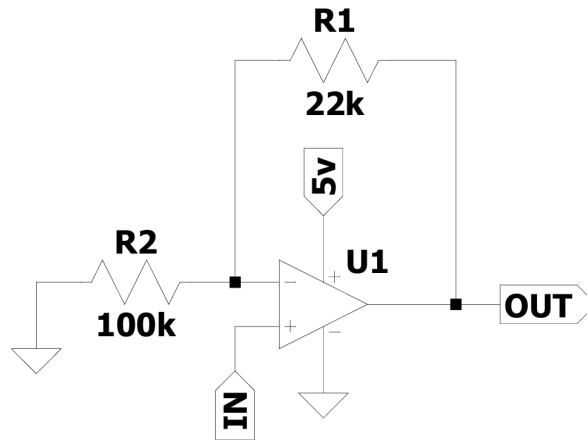


Figura 3.13: Amplificador no inversor para la salida del sensor

La siguiente modificación a realizar en el esquemático es la eliminación de las compuertas NOT. Esto es posible gracias a que el software STM32 Cube IDE, que es el software con el que se programa y configuran los periféricos

del microcontrolador elegido, permite cambiar la polaridad de las señales PWM generadas, lo que significa que puede invertir las señales de control del sensor sin necesidad de las compuertas NOT, por lo que se puede prescindir de estas, volviendo al circuito más simple y compacto. Debido a su tamaño, este circuito puede ser implementado fácilmente en una protoboard para realizar pruebas de funcionamiento como se ilustra en la figura 3.14.

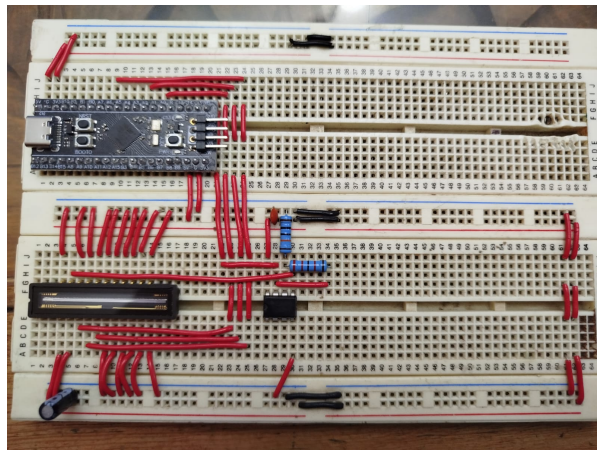


Figura 3.14: Circuito de control implementado en protoboard

Para poder implementar el circuito en conjunto con el optical bench es necesario llevarlo a una PCB. Se optó por utilizar el software de uso libre KiCad, el cual permite realizar múltiples pruebas al diseño y al esquemático realizado, además de contar con gran cantidad de bibliotecas de componentes y sus huellas de soldadura.

Se decidió separar el circuito en dos diseños de PCB separados, uno que irá montado en el optical bench y que solamente implementará el sensor CCD con las terminales necesarias para conectar la alimentación, las señales de control y la salida, mientras que la segunda PCB tendrá la tarjeta BlackPill, el circuito con el OPAM y las resistencias, los capacitores y las terminales necesarias para conectarse con la otra PCB. Originalmente se tenía pensado realizar una sola PCB que estuviera montada en el optical bench, pero debido a la falta de espacio para algunos componentes se decidió dividir en dos.

El esquemático de la primera PCB se muestra en la figura 3.15. En este se pueden ver dos grupos de 20 terminales, J1 y J2, los cuales corresponden a los pines de la tarjeta BlackPill. Los pines 9, 13 y 15 son utilizados para generar las señales digitales que controlan al sensor, y están conectadas mediante el uso de etiquetas (oM, SH e ICG, respectivamente) a los pines 2, 4 y 5 un grupo de 6 terminales, J3, el cual es el encargado de comunicar la primera PCB con la segunda que va montada en el optical bench. Los pines 1 y 2 de J2, y los pines 18 y 19 corresponden a los pines de alimentación, +5v y GND, y estos son

conectados a los capacitores C1 y C2, a los pines 4 y 8 del TL082 y a los pines 1 y 6 de J3. Finalmente, la señal de salida del sensor ingresará por el pin 3 de J3, el cual se conecta a la entrada no inversora del primer OPAM del TL082, la cual corresponde al pin 5 del mismo (usando la etiqueta OS), mientras que la salida del OPAM se conecta a la entrada del ADC del microcontrolador (usando la etiqueta ADC_IN), correspondiente al pin 12 de J2.

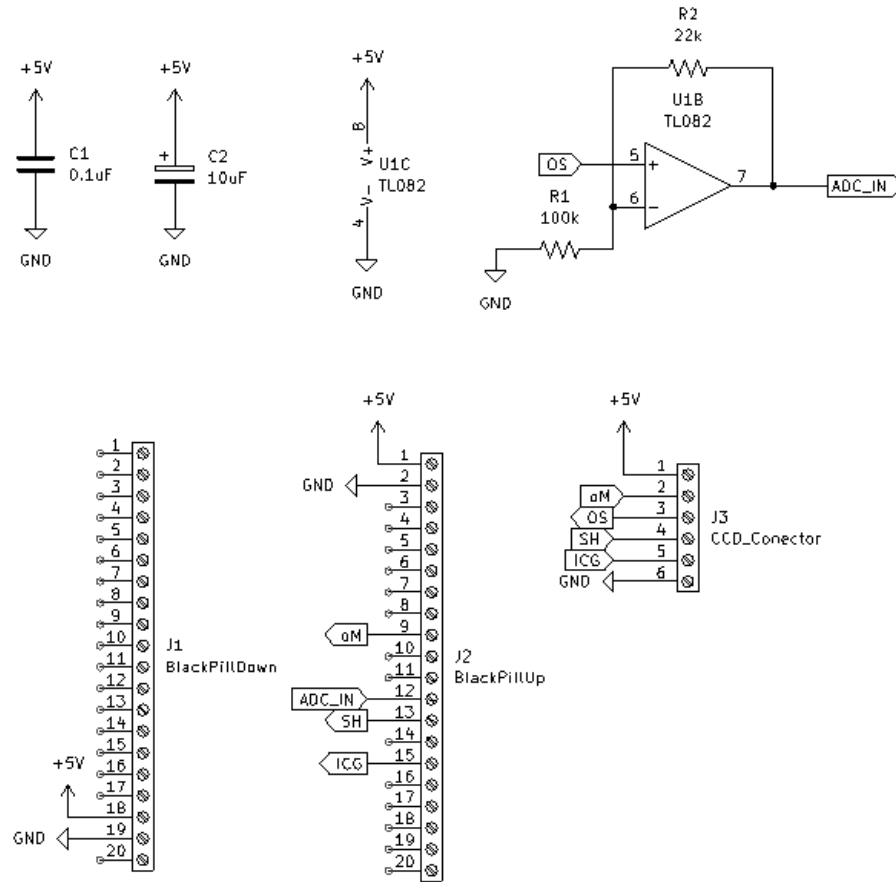


Figura 3.15: Esquemático primera PCB circuito de control

A pesar de que esta PCB no va montada en el optical bench, se buscó hacerla lo más pequeña posible, teniendo como dimensiones finales 4.2x6.5 cm. Todos los componentes soldados fueron del tipo agujero pasante (o through-hole en ingles), ya que estos son más fáciles de conseguir, y no requieren de equipo especial para soldarlos, a diferencia de los componentes de montaje superficial (SMD, por sus siglas en ingles), los con los cuales muchas veces se llega a reque-

rir estaciones de soldar especializadas, lupas, hornos, etc. Las resistencias y el TL082 se encuentran debajo de la tarjeta BlackPill, mientras que los capacitores y el conector J3 se encuentran a un costado. Esto se puede apreciar en la figura 3.16, donde se muestra la PCB terminada.

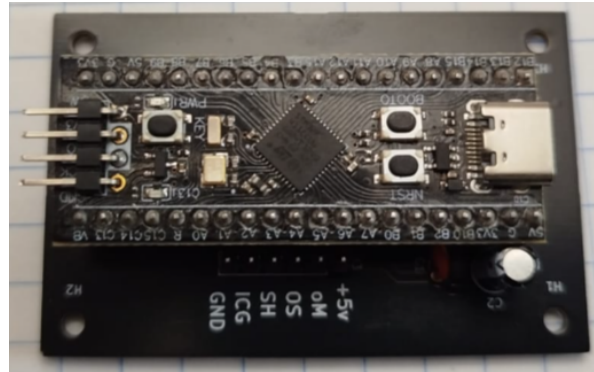


Figura 3.16: Primera PCB circuito de control terminada

Para la segunda PCB, lo que se debió considerar que esta pudiera ser montada correctamente en la abertura del optical bench, que el sensor estuviera centrado para captar de buena manera la luz difractada, incluir un conector para comunicarse con la otra PCB, y mediante el cual se se obtendría la alimentación para el sensor, las señales digitales para su control y enviar la señal de salida hacia el microcontrolador. El diagrama esquemático para esta segunda PCB se puede ver en la figura 3.17, en esta se puede apreciar solamente al sensor CCD conectado mediante el uso de etiquetas al grupo de 6 seis terminales.

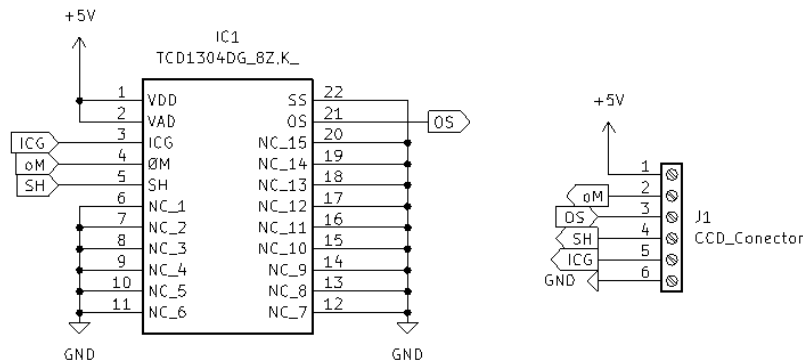


Figura 3.17: Esquemático segunda PCB circuito de control

El diseño final de la segunda PCB tuvo como dimensiones 2.15x6cm, lo que se acopló fácilmente al optical bench como se puede ver en la figura 3.18, mientras

que sensor fue montado sobre dos tiras de pines hembra, las cuales estaban soldadas directamente a la PCB, esto para dar la posibilidad de reemplazar al sensor en caso de que este se dañe.

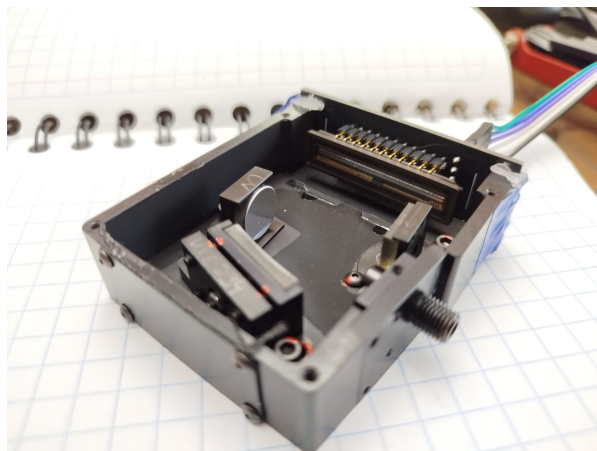


Figura 3.18: Segunda PCB circuito de control terminada montada sobre el optical bench

Al conectar las dos PCB's, lo único que requiere el espectrómetro para ser utilizado es conectar el receptor, lo cual se hace conectando un cable USB-C al microcontrolador, e ingresar alguna luz a través de la apertura del optical bench para que sea captada por el sensor, para lo cual se debe contar con un arreglo óptico específico, lo cual es desarrollado en la sección 3.2.5. La imagen muestra como se ve el el optical bench con el circuito de control implementado.

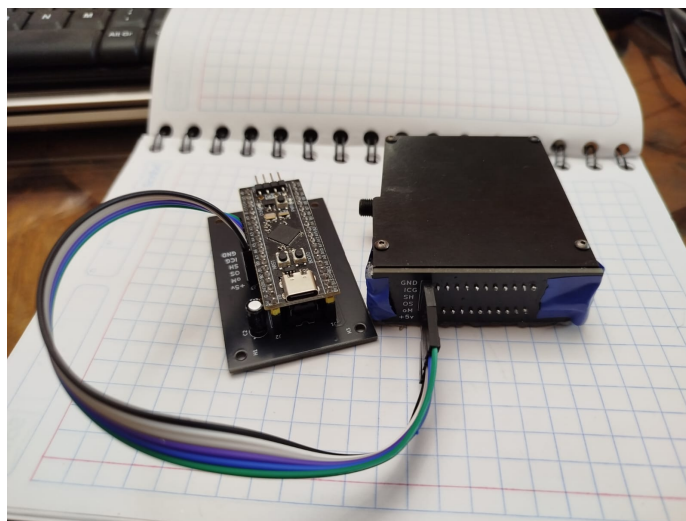


Figura 3.19: Óptical bench con circuito de control implementado

3.2.5. Arreglo óptico

Una vez que se tiene implementado el circuito de control en el optical bench, solo se necesita de dos cosas para que el espectrómetro este completo, la fuente de luz (de la cual se hablará en la sección 3.2.6) y el arreglo óptico, que es de lo que se hablará en esta sección.

Como se menciona en la sección 2.7.3, los sistemas ópticos se encargan de la manipulación controlada de la luz mediante la interposición de cuerpos reflectores y reflectantes. En el caso del espectrómetro de absorción, el arreglo óptico tiene como objetivo enfocar la luz de la fuente hacia la muestra que se desea medir, y de esta hacia el sistema electrónico que se encarga de medir la intensidad de la luz. La implementación del monocromador antes o después de la muestra depende del diseño específico de espectrómetro. Al contar con el optical bench Cerny-Turner, ya no se requiere considerar un monocromador externo como parte del sistema óptico. Lo que se requiere es concentrar la luz proveniente de la fuente en un solo punto sobre la muestra, la cual estará en una cubeta de cuarzo diseñada para trabajar en un rango de 190-2500 nm, como la que se ve en la figura 3.20.

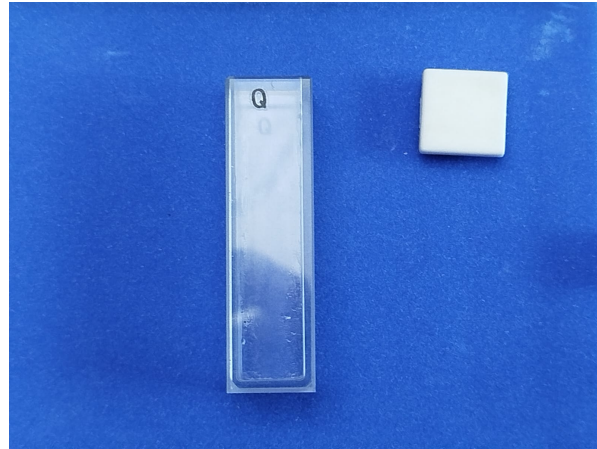


Figura 3.20: Cubeta para muestras

De igual forma, también se requiere concentrar la luz que atraviesa la muestra en la entrada del optical bench, que será a través de una fibra óptica. En la sección 2.7.3 se mencionan muchos elementos con funciones específicas que pueden formar parte de un sistema óptico. En esta misma sección se menciona que las lentes son dispositivos refractores, las cuales pueden ser utilizadas para converger la luz en un solo punto si la lente es convexa, o separar los rayos de luz si es cóncava. Para el arreglo óptico se utilizaron dos lentes plano convexas, como se puede ver en la figura 3.21, las cuales están montadas sobre marcos para facilitar su manipulación.

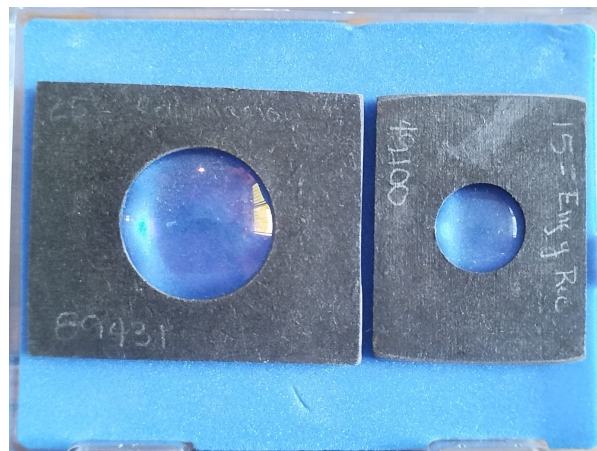


Figura 3.21: Lentes plano convexas para arreglo óptico

Una de la lentes tiene un diámetro de 2.5 cm, la cual irá posicionada entre la fuente de luz y la cubeta para muestras. La otra lente tiene un diámetro de 1.5 cm, y estará ubicada entre la otra cara de la cubeta y la fibra óptica, la cual también está montada sobre un marco. Para poder montar todo el arreglo óptico, se utilizaron pequeñas bases, las cuales permiten alterar la distancia entre cada elemento y la altura a la que estos se encuentran, además de brindar versatilidad para desmontar y transportar el equipo o reemplazar alguno de los elementos si así se desea. Se puede ver en la figura 3.22 el espectrómetro construido, con el arreglo óptico en conjunto con el óptico bench y el circuito de control implementado.

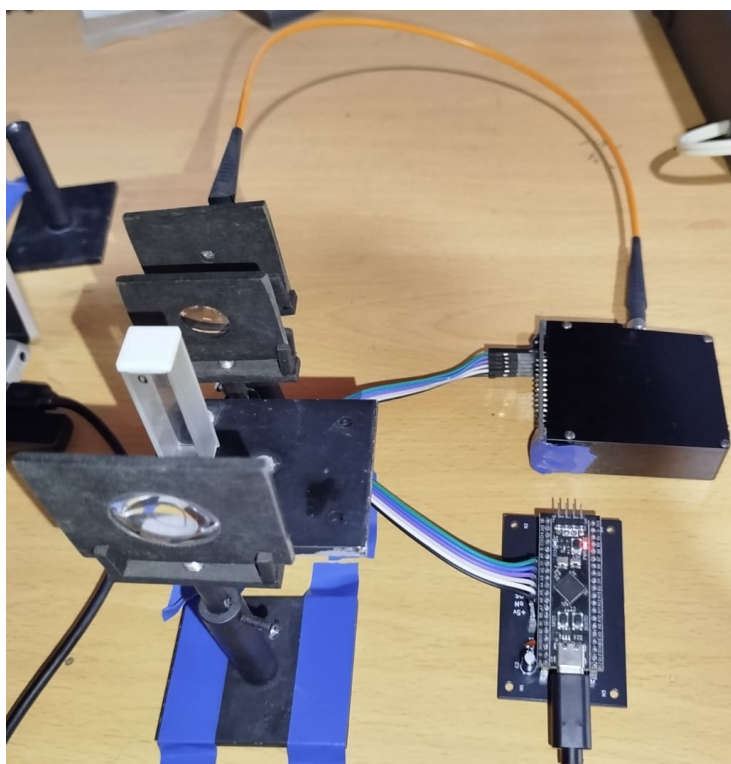


Figura 3.22: Espectrómetro construido

3.2.6. Fuente de luz

Como se explicó en la sección 2.7.4, las fuentes de luz más utilizadas para los espectrómetro UV-Vis son la lámpara de arco de deuterio, la cual se usa para principalmente para cubrir la región UV, y la lámpara halógena de wolframio, para cubrir el espectro visible. Ya que se pretende que el espectrómetro desarrollado trabaje solo en el rango de luz visible, se puede prescindir de una fuente

de luz que cubra el rango UV, por lo que la lámpara halógena de wolframio sería suficiente. Desafortunadamente, actualmente es complicado conseguir este tipo de lámpara, por lo que se tuvo que optar por una alternativa. En la sección 2.7.4 se menciona a la lámpara de xenón pulsante y a la lámpara halógena de filamento de tungsteno como alternativas, siendo esta última bastante fácil de conseguir, ya que sigue siendo implementada en iluminación automotriz y en lámparas de mano.

La fuente de luz utilizada fue una lámpara halógena de filamento de tungsteno, la cual se puede observar en la figura 3.23.



Figura 3.23: Lámpara halógena de tungsteno

Esta funciona con 6 Vdc cuenta con dos bombillas, una que emite luz de forma dispersa y otra que está montada sobre un reflector esférico cóncavo que ayuda a concentrar la luz en un punto específico, es por esto mismo que esta lámpara fue elegida para ser utilizada en el espectrómetro. La respuesta espectral de esta lámpara, captada por el espectrómetro, se muestra en la figura 3.24, donde el eje X representa las longitudes de onda, las cuales son 455-652 nm (en la sección 3.3.3 se hablará más a fondo de como es que se calibra el equipo para reconocer las longitudes de onda en las gráficas obtenidas), y el eje Y representa la intensidad de la luz captada (la cual va desde 0 hasta 4095, rango que está dado por la resolución del ADC del microcontrolador, y de lo que se hablará más al respecto en la sección 3.2.7).

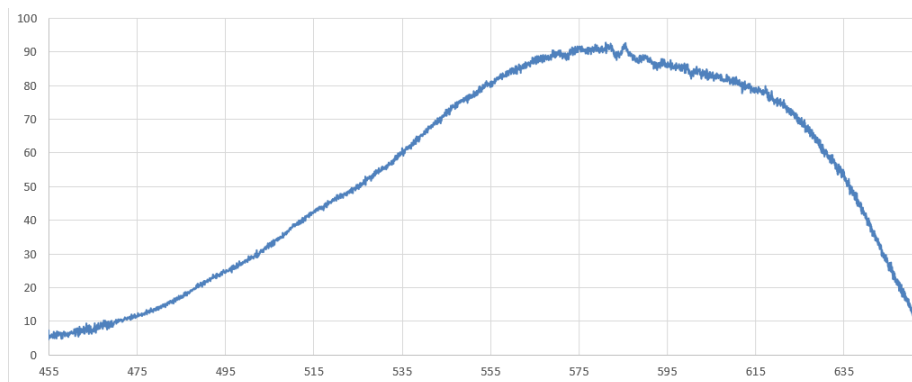


Figura 3.24: Respuesta espectral lámpara halógena de tungsteno

3.2.7. Captura de datos

La captura de los datos es un trabajo que se realiza enteramente en el microcontrolador, y para poder entender la manera en que esta se realiza deben abordarse las diferentes etapas para la programación del dispositivo.

El software utilizado para programar el microcontrolador seleccionado es el STM32Cube IDE, el cual permite realizar la configuración de periféricos mediante herramientas gráficas que facilitan la tarea, además de tener la posibilidad de escribir el código en lenguaje C, C++ o en lenguaje ensamblador, o inclusive combinar varios lenguajes en un solo proyecto.



Figura 3.25: Software STM32Cube IDE para la programación del microcontrolador

Como primer paso para crear el proyecto, se debe ir a la pestaña "File", la cual despliega varias opciones, de entre las cuales se selecciona "New" y después "STM32project", lo abre automáticamente un buscador, en el que debe ser introducido el modelo de microcontrolador a utilizar y seleccionar el tipo de encapsulado, en este caso sería el STM32f401CC6 en encapsulado UFQFON48, que es el que implementa la tarjeta Black Pill. Una vez nombrado el proyecto, y seleccionada la dirección en donde se guardará, el software despliega la interface gráfica para la configuración de los periféricos, como se ve en la figura 3.26.

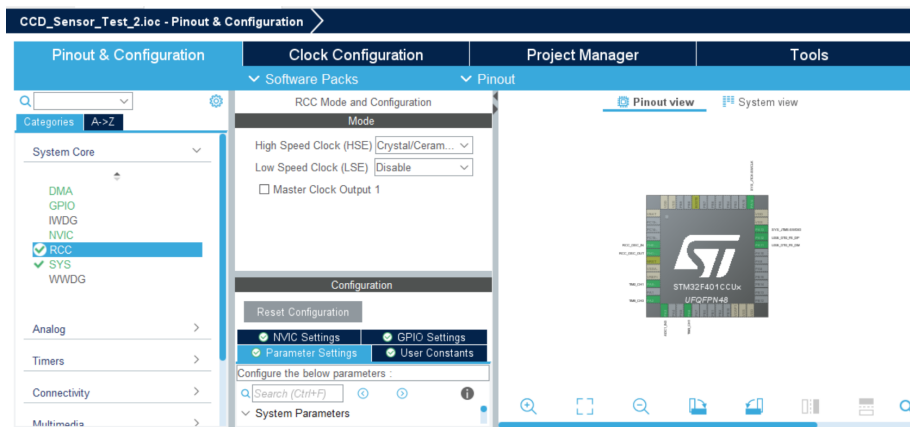
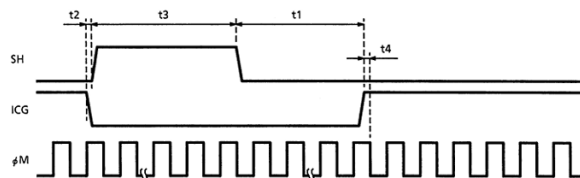


Figura 3.26: Interface gráfica para la configuración del microcontrolador

Lo primero que se debe configurar es el "HighSpeedClock" con la opción de "Crystal/CeramicResonator", la cual permite utilizar un cristal de cuarzo externo, que en este caso es de 25 MHz, lo que permite trabajar con una velocidad máxima de 84 MHz gracias a un circuito PLL interno en el microcontrolador.

El sensor CCD requiere de tres señales PWM para ser controlado, como se mencionó en la sección 3.2.4, las cuales deben tener frecuencias, ciclos de trabajo y retrasos entre señales específicos, lo cual puede verse ilustrado en la figura 3.27. Por lo tanto, lo siguiente a configurar en el microcontrolador son las señales PWM, lo cual se hace mediante la manipulación de los Timer's.

TIMING REQUIREMENTS



CHARACTERISTIC	SYMBOL	MIN	TYP.	MAX	UNIT
ICG Pulse DELAY	t1	1000	5000	—	ns
Pulse Timing of ICG and S H	t2	100	500	1000	ns
SH Pulse Width	t3	1000	—	—	ns
Pulse Timing of ICG and ϕ M	t4	0	20	*	ns

*: You keep ϕ M "High" Level.

Note: If you use electronic shutter function. t_{INT} (MIN.) = 10 μ s

Figura 3.27: Requerimientos de señales de control de sensor CCD
. Tomado de [32]

En la sección 3.2.4 se establece que las señales ϕ M, SH e ICG son generadas por los pines 9, 13 y 15 del grupo de terminales J2, los cuales corresponden a los pines PA6, PA2 y PA0 del microcontrolador. Estos pines corresponden a las salidas PWM generadas por los Timer's TIM3, TIM5 y TIM2 respectivamente, por lo que serán estos los que deberán configurarse para obtener las señales de control adecuadas. Debido a la configuración de la frecuencia del microcontrolador hecha anteriormente, los Timer's trabajan con una frecuencia base de 84MHz, la cual deberá ser dividida entre el preescalador y el periodo del contador para reducir esa frecuencia. La hoja de datos del sensor establece que el valor típico de ϕ M es de 2MHz, por lo que los valores del preescalador usados son 0, lo que hará que la frecuencia no se divida entre este valor, y el periodo de contador será 42 lo que genera una frecuencia de 2MHz, además se estableció un ciclo de trabajo del 50 % estableciendo el valor en el parámetro "pulse". La figura 3.28 muestra la señal ϕ M generada por el microcontrolador captada por un osciloscopio.

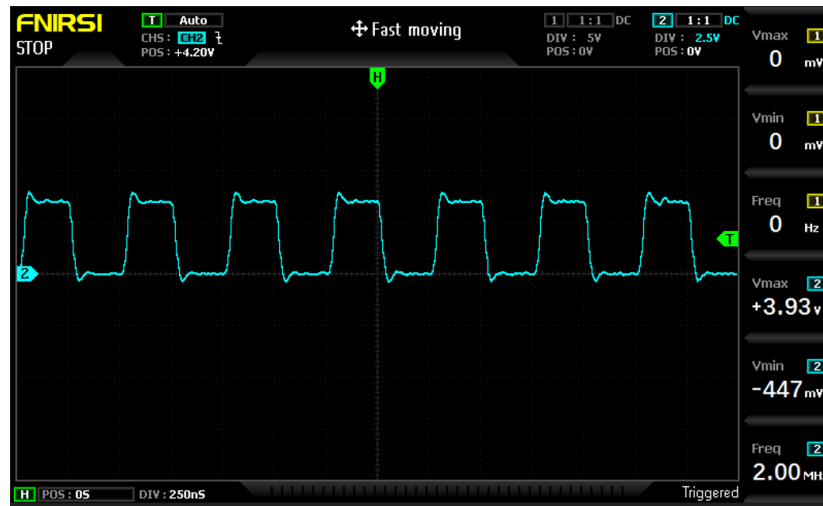


Figura 3.28: Señal oM captada por osciloscopio

Para la señal ICG se estableció una frecuencia de 133.33Hz, y un ciclo de trabajo de 0.1333 %, además de tener que cambiar la polaridad de esta señal manipulando el parámetro "*CHPolarity*" para que coincida con lo mostrado en la hoja de datos, y así poder eliminar la compuerta lógica NOT del circuito de control como se explicó en la sección 3.2.4. Para la señal SH, se estableció que la fuente de disparo fuera la señal ICG, lo que hará que estén sincronizadas y se activen al mismo tiempo. Se estableció la frecuencia de SH en 50KHz, y su ciclo de trabajo en 20 %. Lo último que se debe hacer para que estas señales esten listas para conectarlas al sensor es generar el retraso t2 presente en la figura 3.27. Para lograrlo se debe ir al código de programación, como se ve en la figura 3.29, en donde se utiliza la función especial, la cual genera un retraso de 600 ns a la señal ICG.

```

113 HAL_TIM_PWM_Start(&htim2, TIM_CHANNEL_1); //ICG
114 HAL_TIM_SET_COUNTER(&htim2, 66); //600NS DELAY
115 HAL_TIM_PWM_Start(&htim5, TIM_CHANNEL_3); //SH

```

Figura 3.29: Generación de retraso señal ICG en código de programación

Las señales resultes captadas por el osciloscopio se pueden ver en la figura 3.30, las cuales son bastante parecidas a lo visto en la figura 3.27 proveniente de la hoja de datos del sensor.

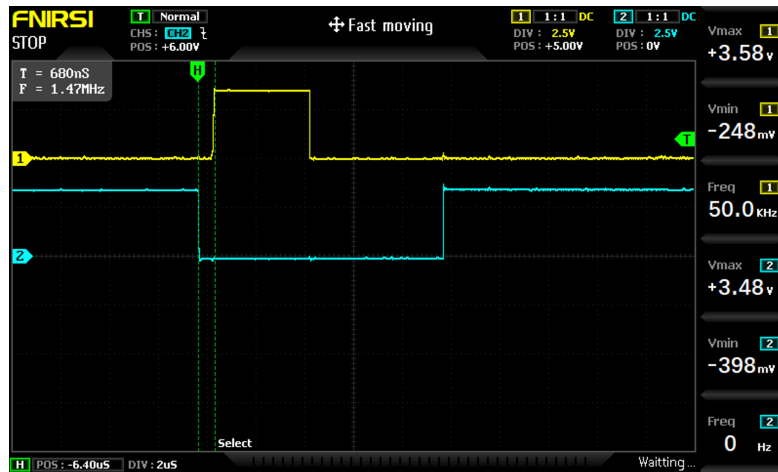


Figura 3.30: Señal ICG y SH captadas por osciloscopio

Conectando las tres señales de control, el sensor comenzará a generar una señal por la terminal OS. Según la hoja de datos del sensor, al principio y al final del periodo de la señal generada por OS hay un escalón, el cual corresponde a un conjunto de pixeles que sirven para indicar el principio y final de la lectura, y de esta manera ubicar físicamente en que sección del sensor se está recibiendo luz, esto puede apreciarse en la figura 3.31. La hoja de datos llama a estos pixeles como "Dummy pixels" (pixeles ficticios en español), ya que a pesar de estar presentes en la señal de salida del sensor, estos no reaccionan a la luz captada, manteniendo siempre un mismo valor.

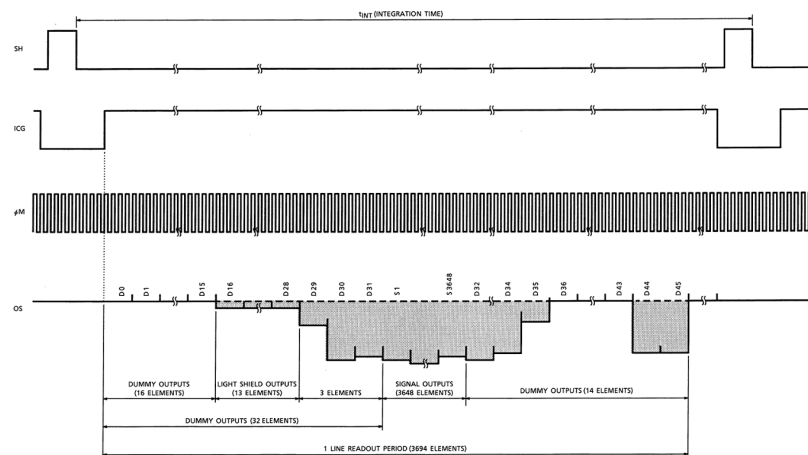


Figura 3.31: Representación ideal de la señales de control y de salida del sensor CCD

Tomado de [32].

Identificar a los dummy pixels es muy importante para capturar la señal del sensor, ya que estos nos indican cuando termina y comienza un ciclo de la señal de salida, garantizando una buena sincronización de la señal al momento de graficarla, y así evitar que esta se desplace cuando se capturen varios espectros. Desafortunadamente, captar a los dummy pixels resulta ser algo complicado debido a que cuando el sensor llega a estar en completa oscuridad la señal útil llega al mismo nivel de voltaje de los escalones generados, por lo que estos termina desapareciendo al tener una señal contante en un solo nivel de voltaje. Afortunadamente, existe otra manera de identificar el punto donde inicia y termina la señal de salida del sensor, la cual involucra a la señal ICG. En la figura 3.32 se puede apreciar como los cambios de estado de la señal ICG (señal azul) están en sincronía con los escalones generados por los dummy pixels en la señal OS (señal amarilla), por lo que solo se tiene que detectar los cambios de estado de ICG para saber cuando empezar a capturar la señal OS y tener una correcta sincronización.

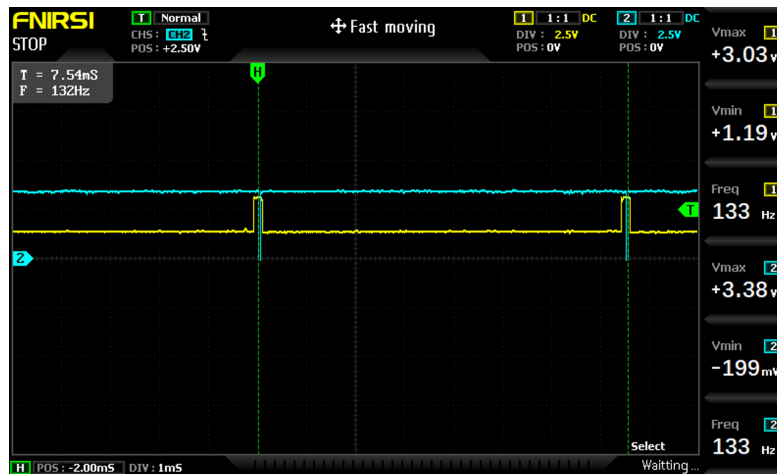


Figura 3.32: Señales ICG y OS captados por osciloscopio

Para detectar los cambios de estado de la señal ICG debe activarse la opción de interrupción global en la configuración del Timer en la sección "*NVICSettings*", esto interrumpirá al código principal y ejecutará lo que se programe en una función especial, como se puede ver en la figura 3.33. Algo igual de importante que debe ser configurado para la captura de la señal OS es el ADC y un Timer que será usado para establecer la frecuencia de muestreo de la señal. El Timer utilizado es el Timer 4, el cual se configura como "*PWMGenerationNoOutput*", lo cual genera un evento a una frecuencia determinada (que en este caso es de 500KHz) para tomar una muestra con el ADC, y al no ser necesario tener una salida de este PWM el valor del ciclo de trabajo no es importante. Para configurar el ADC, solamente se activa la entrada deseada (Pin PA3 del microcontrolador en este caso), se activa la resolución de 12 bits, se activa la fuente

de conversión de disparador externo como el evento generado por el Timer 4 y se activa la opción de acceso directo a memoria (DMA, pos sus siglas en ingles), la transferirá directamente los datos adquiridos por el ADC a la memoria RAM sin intervención del CPU, lo que mejora la velocidad, libera la carga de trabajo del CPU y garantiza una transferencia automática y continua. En la figura 3.33 se puede ver como el evento de ICG inicia la captura de datos del ADC a la frecuencia establecida por el Timer 4, y el CCDPixelBuffer será el encargado de almacenar la información, y esto seguirá así hasta que el buffer se llene. El buffer tiene un tamaño de 3770, ya que con el periodo de muestreo, que son $2\mu s$, se logra cubrir todo el periodo de OS, que son 7.54ms, con 3770 muestras.

```
void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim){
    if (htim->Instance == TIM2){
        HAL_ADC_Start_DMA(&hadc1, (uint32_t *)CCDPixelBuffer, CCDBuffer);
    }
}
```

Figura 3.33: Función para captura de la señal OS

3.3. Software

3.3.1. Envío y recepción de datos

Una vez que se han configurado correctamente las señales de control del sensor CCD, y capturar la señal OS con el ADC, lo único que resta es enviar los datos al dispositivo receptor para que sean procesados. En este caso, el dispositivo receptor es una computadora, la cual puede establecer comunicación con el microcontrolador mediante alguno de los protocolos de comunicación con los que este cuenta. Para este caso se decidió utilizar la comunicación USB, ya que la tarjeta BlackPill ya cuenta con un puerto USB-C, por lo que no se requiere más que un cable USB para conectarlo a la computadora. Configurar al microcontrolador como dispositivo USB es bastante sencillo, solamente debe activarse la comunicación como puerto virtual COM en la configuración *USB_Device* dentro de la interface gráfica de configuración, y el modo "*Device_Only*" en la configuración *USB_OTG_FS*. Para poder transmitir la información se cuenta con una función especial que se utiliza dentro del código de programación. En la sección anterior se explicó que el ADC se mantiene tomando muestras hasta que el buffer que se le asignó se llene por completo, por lo que el momento en que los datos deben ser enviado es cuando el ADC deje de tomar muestras, para lo cual existe una función a la que el programa es redirigido cuando esto pasa, y es dentro de esta en donde se ejecuta la transmisión de datos por USB como se ve en la figura 3.34.

```

void HAL_ADC_ConvCpltCallback(ADC_HandleTypeDef* hadc){
    CDC_Transmit_FS((uint8_t *)CCDPixelBuffer,CCDBuffer*2);
}

```

Figura 3.34: Función para envío de datos por USB

Para este punto, el resto del trabajo se realiza en el dispositivo receptor, por lo que la programación del microcontrolador ha terminado (Para ver todo el código de programación del microcontrolador consulte el apéndice A.1). Para recibir los datos se decidió crear un programa en python, ya que es un lenguaje que es un lenguaje con el que es bastante fácil trabajar y visualizar datos debido a la gran cantidad de librerías especializadas con las que cuenta, como se explicó en la sección 2.10.

Lo primero que debe hacerse es configurar el puerto USB, lo cual se hace utilizando la librería "serial", donde se establece el puerto COM a utilizar, que en este caso es el COM5, también se utiliza la función *.ion()*, la cual pertenece a la librería "matplotlib.pyplot", y que activa el modo interactivo para graficar los datos recibidos. Después se crea un arreglo de 3770 espacios inicializados en cero, ya que es aquí donde se guardarán los datos recibidos. Se establecen los valores de dos variables muy importantes, *NUM_ESPEC*, el cual establece cuantos espectros serán los que se promediarán, y *INTEG_TIME* que establece el tiempo de integración entre cada espectro tomado. Se considera como un espectro a un cúmulo de 3770 datos recibidos, y estos se promedian para obtener una lectura más clara, precisa y eliminar algo del ruido que pudiera presentarse. El tiempo de integración entre cada espectro tomado puede afectar de manera positiva o negativa dependiendo de que tan corto o largo sea. Para todas las lecturas hechas con el espectrómetro siempre se utilizará un tiempo de integración de 1ms y 100 espectros promediados. Todo lo explicado hasta este punto puede verse en la figura 3.35.

```

# Configura el puerto USB CDC
ser = serial.Serial( port='COM5', timeout=1) # Cambia 'COM5' al puerto correcto

plt.ion() # Modo interactivo para la gráfica

# Inicializa el arreglo con ceros para mantener siempre 3770 datos
NUM_SAMPLES = 3770
data_array = np.zeros(NUM_SAMPLES, dtype=np.uint32)

# Variables de control de captura de espectros
NUM_ESPEC = 100
INTEG_TIME = 0.001

```

Figura 3.35: Configuración en Python de USB y variables para captura de espectros en el receptor

Dentro de la función *read_and_plot* primero se asegura que hay suficientes datos que leer, si hay menos que 3770 significa que los datos no se enviaron completos y deben ser descatados. Ya que los datos que se envían está en forma de bytes puros deben ser convertidos a formato *uint16_t*, que es el formato con el que fueron almacenados los datos en el buffer en el microcontrolador, para lo cual se utiliza la función *frombuffer* de la librería *numpy*. Algo a tomar en cuenta es que el sensor trabaja con lógica invertida, lo que significa que entre más luz detecte, la señal OS tendrá un valor menor, y entre menos luz detecte OS será mayor. Para cambiar la lógica se restó a 4095 (que es el valor digital máximo que puede tomar el ADC del microcontrolador por su resolución de 12 bits) el valor leído como se puede ver una la figura 3.36.

```
def read_and_plot(): 1 usage
    global data_array # Se usa global para modificar la lista fuera de la función
    CONT_ESPEC = 0
    while True:
        if ser.in_waiting >= NUM_SAMPLES * 2: # Asegura que haya suficientes datos antes de leer
            # Lee exactamente 3770 muestras (cada muestra son 2 bytes)
            raw_data = ser.read(NUM_SAMPLES * 2)

            # Convierte los datos crudos a enteros (asumiendo 16 bits por dato)
            int_data = 4095 - np.frombuffer(raw_data, dtype=np.uint16)

            # Suma los nuevos datos con los datos anteriores
            data_array += int_data
```

Figura 3.36: Recepción y conversión de los datos recibidos

3.3.2. Visualización y almacenamiento de espectros

Una vez se tienen los datos de manera correcta, esto se van acumulando en un buffer, y al mismo tiempo estos se van graficando hasta llegar al limite establecido como se puede ver en la imagen 3.37.

```
# Ajusta el limite del eje Y al máximo valor actual de la gráfica
y_max = np.max(data_array) * 1.1 # Agrega un 10% de margen visual
y_max = max(y_max, 100) # Evita que el limite sea demasiado pequeño

# Limpia la gráfica y vuelve a graficar
plt.clf()
plt.plot(data_array)
plt.ylim(*args: 0, y_max) # Límite dinámico del eje Y
plt.xlim(*args: 0, NUM_SAMPLES) # Mantiene siempre 3770 en X
plt.xlabel('Pixel')
plt.ylabel('Valor Acumulado')
plt.title('Datos Acumulados por USB')
plt.pause(INTEG_TIME) # Breve pausa para actualizar la gráfica
```

Figura 3.37: Graficar los datos

Después de ser promediados los datos, estos pasan a ser guardados en un

archivo ".csv" como se ve en la figura 3.38.

```
CONT_ESPEC += 1
if CONT_ESPEC == NUM_ESPEC:
    data_array = data_array / NUM_ESPEC
    # Guardar datos en un archivo Excel
    # Guardar solo los datos del índice 33 al 3681
    data_to_save = data_array[32:3680]
    try:
        df = pd.DataFrame(data_to_save, columns=['Valor'])
        df.to_csv(path_or_buf='Espectro_Naranja.csv', index=False)
        print("Datos guardados en datos_recibidos.xlsx")
    except Exception as e:
        print("Error al guardar los datos:", e)
    break
```

Figura 3.38: Graficar y guardar los datos

3.3.3. Calibración del equipo

Para este punto ya pueden obtenerse espectros sin ningún problema, sin embargo hay varias cosas que deben calibrarse para poder interpretar la información de manera correcta. Para probar el equipo se incidió una luz blanca genera por un led, cuyo espectro se puede ver en la figura 3.39.

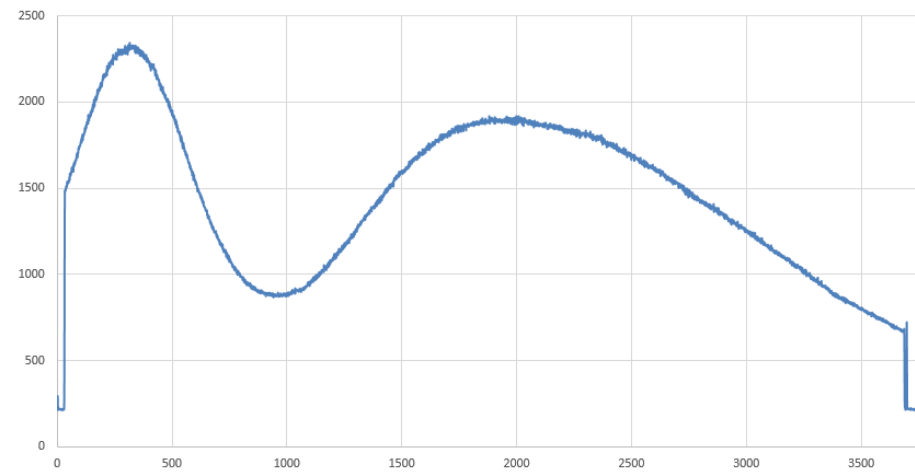


Figura 3.39: Espectro luz blanca Led

Lo primero que se puede notar es que a los extremos del espectro están presentes los dummy pixels generados por el sensor, lo que puede generar confusión respecto al comportamiento del espectro en esas zonas. Para eliminar a los

dummy pixels del espectro solo basta con identificar cuales muestras contienen a los dummy pixels para prescindir de estos al momento de graficar y guardar el espectro, en este caso desde la posición 32 hasta la 3680 del buffer que contiene los datos es información útil, por lo que los demás espacios contienen a los dummy pixels de los extremos, y solo basta con indicar el rango del buffer con el que se quiere trabajar entre corchetes en el código de python. De esta forma, el espectro esta ahora formado por 3648 muestras, las cuales coinciden con la cantidad de pixeles con los que cuenta el sensor, por lo que ahora cada dato contenido en el buffer se corresponde con el pixel en su misma posición.

Lo siguiente que debe ser tratado es un offset presente al rededor del valor de 250. Si bien la presencia de este no afecta la integración del espectro, su eliminación ayuda a visualizar en que zonas del sensor no se esta incidiendo luz. Para eliminarlo puede restarse el nivel de voltaje necesario a la señal OS usando un OPAM, pero hacer eso implicaría rehacer todo el circuito de control, por lo que la solución que se aplicó fue tomar un espectro en oscuridad total antes de tomar un espectro con una muestra, para que de esta manera se tenga la referencia del nivel del offset, y después restarle este espectro a los tomados con muestras, además de que hacer esto también elimina algo del ruido propio del sensor.

Lo único que resta es determinar que longitud de onda le corresponde a cada pixel, para lo cual se utiliza la interpolación lineal, que es un método matemático que permite estimar el valor de una función entre dos puntos conocido, asumiendo que el cambio sea lineal, como se explicó en la sección 2.9. Para lograrlo se tuvieron que tomar varios espectros de fuentes de luz monocromáticas de las que se conozca su longitud de onda, para después, mediante un análisis de los espectros, determinar cuales pixeles son los que presentan mayor amplitud, se utilizaron láseres y leds RGB para obtener los datos. Mediante un código en python, estos datos fueron utilizados para generar la interpolación, ya que la librería numpy cuenta con una función que la realiza, para después guardar los datos en un archivo ".CSV" como se puede ver en la figura 3.40. Algo importante a mencionar es que los extremos tuvieron que ser extrapolados, ya que no se contaba con fuentes de luz que tuvieran una longitud de onda más baja que 467nm y más alta que 650nm, y graficar con esos limites comprimiría los espectros, y al ver el comportamiento lineal de las longitudes de onda, se optó por utilizar las estimaciones de 455nm en el pixel 4 y 652nm en el pixel 3648 para mejorar el comportamiento (Para ver el código completo para la obtención de la interpolación lineal para las longitudes de onda consulte el apéndice B.1).

```

import numpy as np
import matplotlib.pyplot as plt
import pandas as pd

# Datos conocidos
pixels = np.array([4, 163, 848, 1057, 2666, 3555, 3648])
wavelengths = np.array([455, 467.5, 518.5, 532, 622.5, 650, 652])

# Array de pixeles del sensor CCD
total_pixels = 3648
pixel_array = np.arange(total_pixels)

# Interpolación lineal
wavelength_array = np.interp(pixel_array, pixels, wavelengths)

# Guardar en un archivo Excel
df = pd.DataFrame({'Pixel': pixel_array, 'Longitud de Onda (nm)': wavelength_array})
df.to_csv('path_or_buf: 'Longitudes_de_Onda_CCD.csv', index=False)
print("Datos guardados en Longitudes_de_Onda_CCD")

```

Figura 3.40: Código interpolación lineal para obtener longitudes de onda

La interpolación lineal de la longitud de onda del sensor CCD puede verse en la figura 3.41, donde el eje X son los pixeles y el eje Y son las longitudes de onda. Con esto ya se tiene calibrado el equipo por completo, por lo que ahora está listo para tomar espectros de muestras, lo que será abordado en la sección 4.

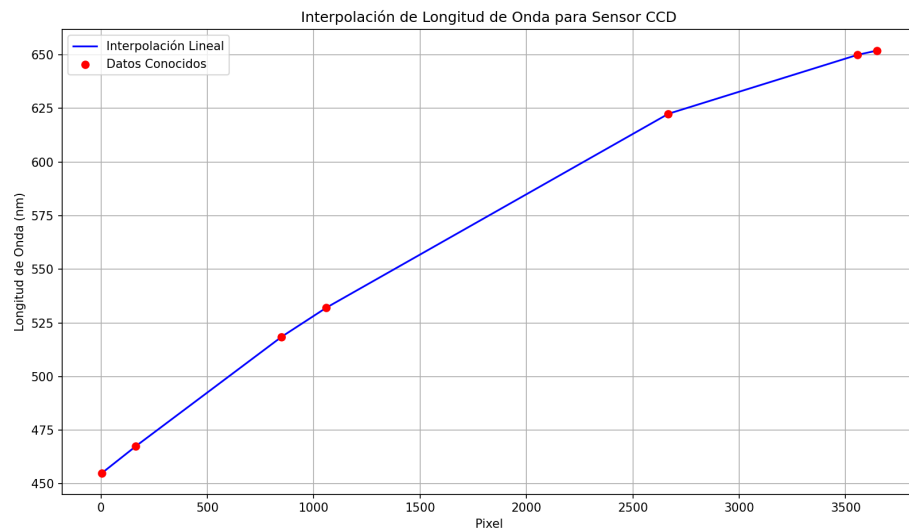


Figura 3.41: Interpolación lineal para obtener longitudes de onda

Ahora si se vuelve a capturar la luz blanca generada por un led, y se grafica utilizando el arreglo de longitudes de onda generadas por la interpolación lineal, se obtiene un espectro que es más fácil de interpretar al ya no contar con los dummy pixels y el offset, además de poder identificar que longitudes contribuyen más a generar esa luz, como se puede ver en la figura 3.42, donde el eje X es la amplitud captada por el ADC del microcontrolador y el eje Y la longitud de onda en nanómetros.

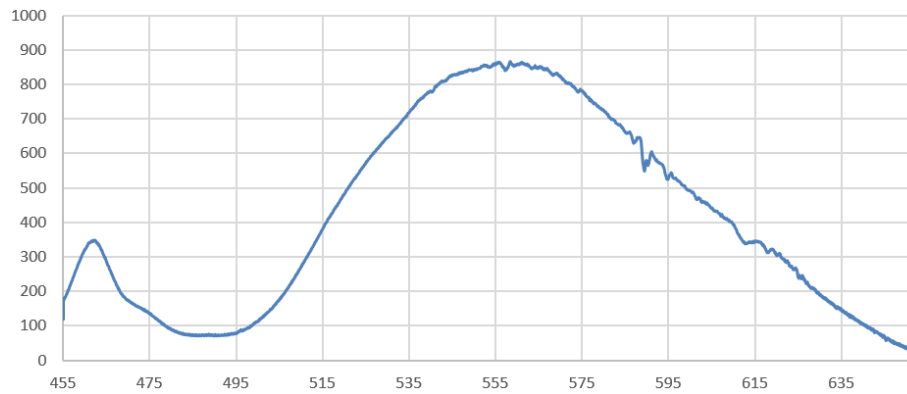


Figura 3.42: Espectro luz blanca Led con equipo calibrado

3.4. Conclusiones

En este capítulo se abordaron los pasos que se siguieron para la construcción del prototipo del espectrómetro, dividiendo en diferentes secciones las etapas principales del desarrollo del hardware y software para que este funcione correctamente. A pesar de que para este punto el equipo es capaz de capturar espectros, aún falta hablar sobre las muestras con las que el equipo fue probado y de las técnicas de filtrado de ruido que fueron aplicadas, temas que serán abordados en el capítulo 4.

Capítulo 4

Captura de espectros

4.1. Introducción

En este capítulo se muestran los espectros obtenidos con el espectrómetro construido a partir de diferentes muestras, además de mostrar que métodos de filtrado fueron utilizados, como es que estos fueron implementados en el código de python del receptor para realizar el filtrado del ruido durante la integración de los espectros, y como es que estos se ven afectados.

4.2. Espectros de muestras capturados

Como se mencionó en la sección anterior, ahora el espectrómetro es capaz de capturar espectros de diferentes muestras, sin embargo, hay una condición con la que estas deben cumplir para que estas puedan ser consideradas, y es que deben ser translúcidas. Anteriormente se mencionó que las muestras estarían contenidas en una cubeta de cuarzo, la cual permite el paso de la luz hacia la fibra óptica del espectrómetro, por lo que la muestra a analizar también debe permitir el paso de la luz. Por lo anterior, es que la mayoría de las muestras analizadas son líquidos, ya que es bastante fácil encontrar líquidos translúcidos, sin embargo, también se analizaron muestras líquidas opacas, las cuales tuvieron que ser disueltas en un poco de agua para permitir el paso de la luz. También se analizaron muestras sólidas, las cuales eran minerales que permitían el paso de la luz. A continuación se mostrarán los espectros de las muestras capturados.

En la figura 4.1 se pueden apreciar los espectros de las primeras cuatro muestras utilizadas, los cuales son: Aceite de oliva, licor de amaranto, brandy y café soluble. Algo presente en todos los espectros es el ruido de alta frecuencia, el cual impide la integración de espectros bien definidos en zonas específicas o en todo el rango de longitudes de onda. Algo que también es notorio en los espectros de aceite de oliva y de café soluble, es que en ciertas zonas la amplitud tiene valores negativos, la razón de esto es porque al no estar el equipo montado en una estructura de forma permanente, específicamente la fuente de luz, la luz

que llega al sensor puede variar en intensidad, siendo en ocasiones demasiado tenue que llega a estar por debajo del nivel del offset que tiene el sensor, y al momento de restarlo provoca que parte del espectro capturado tenga amplitud negativa.

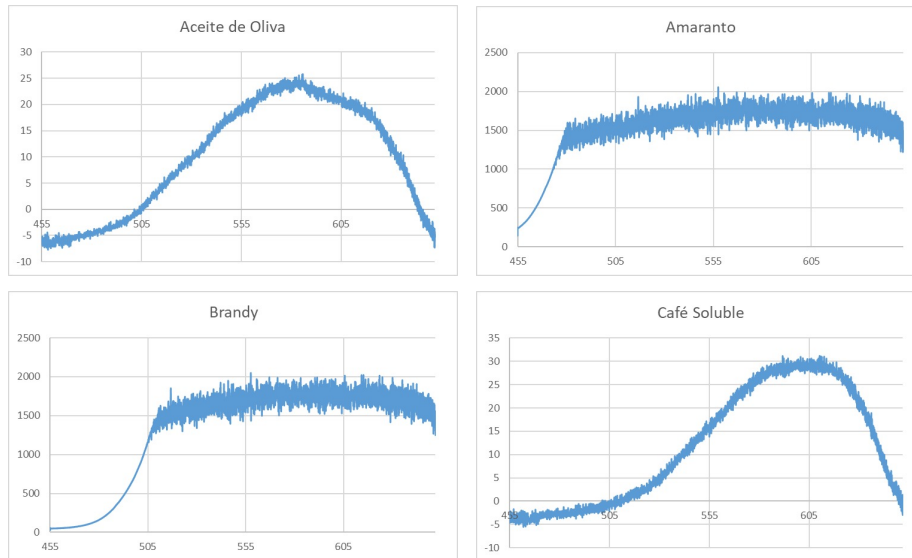


Figura 4.1: Espectros aceite de oliva, licor de amaranto, brandy y café soluble

La figura 4.2 muestras los espectros de clorofila para consumo humano, detergente de ropa, enjuague bucal y agua de jamaica. Nuevamente esta presente el ruido de alta frecuencia, sin embargo hay espectros donde afecta más que en otros, por ejemplo en espectro del detergente de ropa es casi nulo, permitiendo formar un espectro bien definido en todo el rango de longitudes de onda, caso contrario con la clorofila, la cual presenta ruido en todas las zonas, aunque aún es distinguible la silueta que forma el espectro. Algo importante a resaltar es que la clorofila y la jamaica tuvieron que ser preparadas de una forma especial para que sus muestras pudieran ser captadas por el espectrómetro. La clorofila, al ser un liquido muy opaco tuvo que ser disuelto en un poco de agua para aclarar la muestra y permitir el paso de la luz, y la flor de jamaica tuvo que ser remojada en agua caliente, para que el resultado de la infusión pudiera ser tomada como muestra.

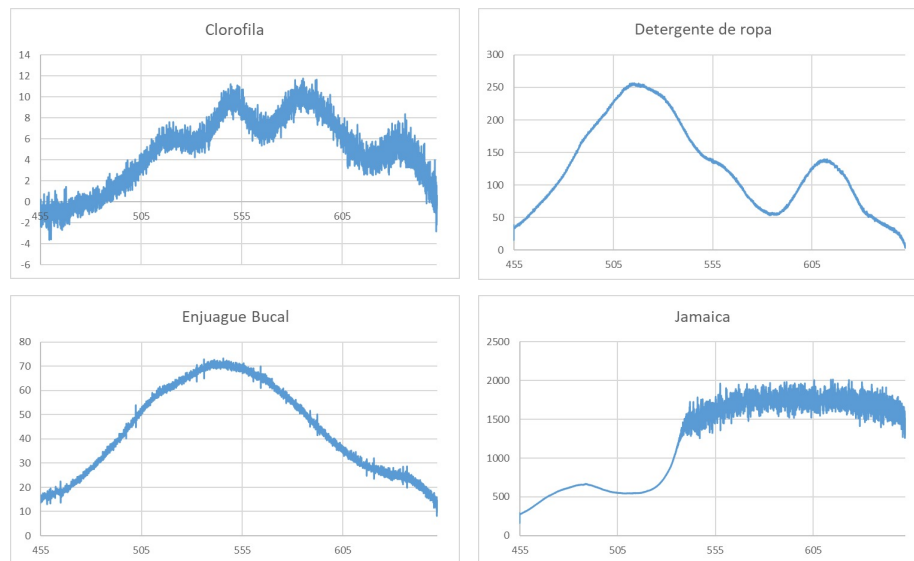


Figura 4.2: Espectros clorofila, detergente de ropa, enjuague bucal y jamaica

En la figura 4.3 se muestran los espectros de leche (la cual tuvo que ser diluida en un poco de agua para permitir el paso de la luz), anticongelante automotriz, líquido limpia vidrios y solución de mertiolate para curar heridas. Con los espectros de la solución de mertiolate y del líquido limpia vidrios es bastante suponer el color del líquido, siendo el primero una mezcla de rojo y naranja, lo que se corresponde a que tener una repuesta plana en longitudes de onda menor a 555nm, mientras que el limpia vidrios tiene un espectro con el efecto contrario al ser de un color azul intenso. Algo igual de interesante de observar es la respuesta espectral de la leche al ser un líquido blanco. De igual forma el anticongelante, el cual era una mezcla de color verde y amarillo, por lo que las longitudes de onda más cercanas al violeta son prácticamente inexistentes.

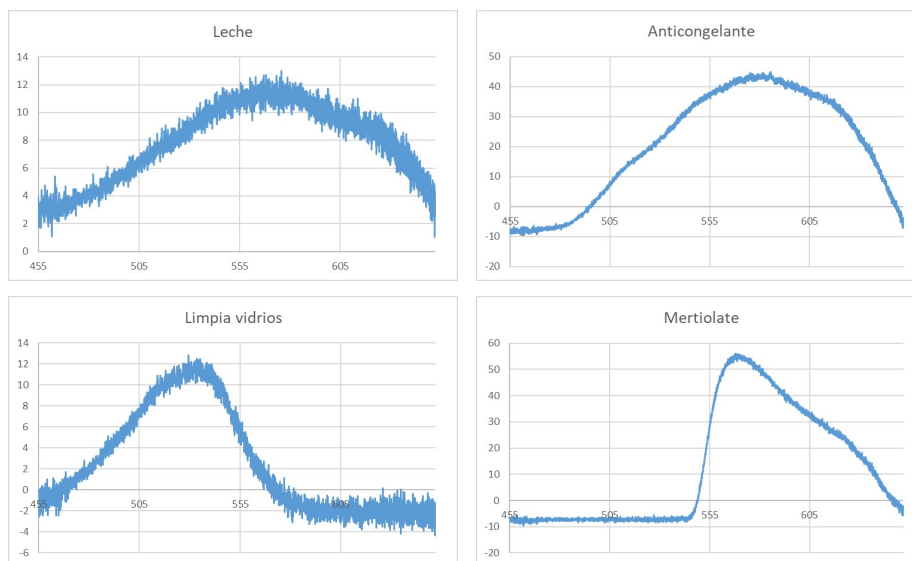


Figura 4.3: Espectros leche, anticongelante, líquido limpia vidrios y mertiolate

La figura 4.4 muestra los espectros obtenidos de jugo de naranja, solución antiácida pepto bismol, líquido limpia pisos pinol y shampoo. En los cuatro espectros esta bastante presente el ruido de alta frecuencia, además que en todos los espectros, a excepción del líquido limpia pisos, la amplitud es bastante baja, lo que provoca que sea difícil distinguir la forma del espectro.

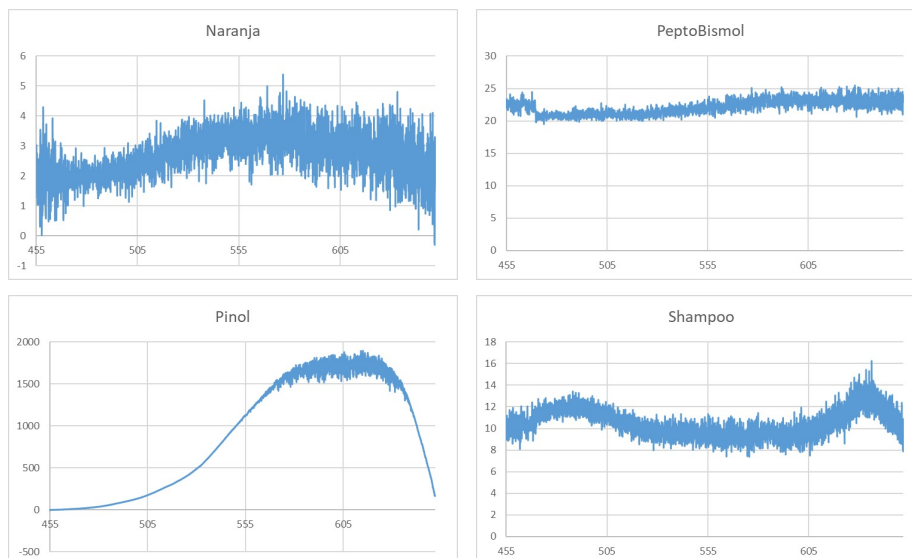


Figura 4.4: Espectros naranja, pepto bismol, pinol y shampoo

En la figura 4.5 se muestran los espectros de whisky (cuya respuesta es bastante parecida a la del brandy y a la del licor de amaranto), la de un láser rojo de longitud de onda de 650nm y la de un láser verde de 532nm. Estos dos últimos forman parte de las fuentes de luz utilizadas para calibrar el espectrómetro como de explicó en la sección 3.3.3.

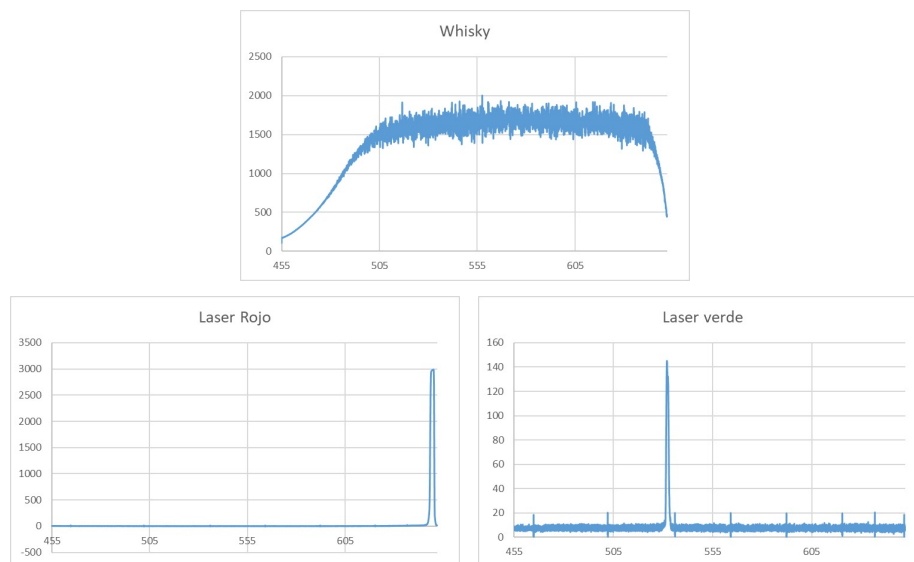


Figura 4.5: Espectros whisky, láser rojo y láser verde

La figura 4.6 muestra los espectros obtenidos de los colores individuales generados por un led RGB. En este caso los picos obtenidos no son igual de estrechos que los generados por los láseres, ya que las luces generadas contiene un rango mas amplio de longitudes de onda, siendo la que coincide con el punto más alto de amplitud en el espectro la que tiene mayor presencia. Estos espectros también fueron utilizados para calibrar el equipo y calcular que longitud de onda le correspondía a cada punto del eje X en el espectro.

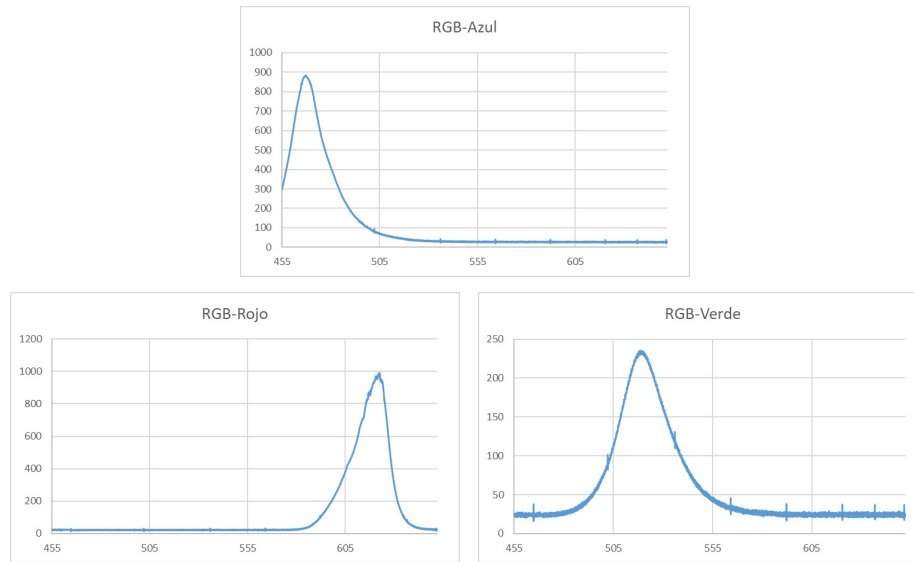


Figura 4.6: Espectros colores Led RGB

La figura 4.7 muestra los espectros de muestras de selenita y fluorita, los cuales a pesar de ser muestras sólidas, permiten el paso de la luz al ser translúcidas, por lo que es posible utilizarlas en el espectrómetro.

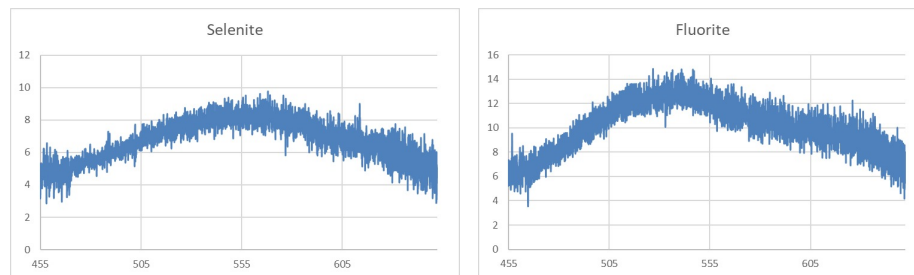


Figura 4.7: Espectros selenita y fluorita

4.3. Filtrado de ruido en espectros

Algo que estuvo presente en todos los espectros mostrados en la sección 4 fu el ruido de alta frecuencia, el cual afecta la forma del espectro, volviendo complicado identificar su comportamiento a lo largo del rango de las longitudes de onda. Afortunadamente, existen muchas técnicas de filtrado que permiten eliminar el ruido de alta frecuencia. Para este caso se utilizaron dos, el filtrado con wavelets y el filtrado con el algoritmo Savitzky-Golay, los cuales fueron explicados en las secciones 2.8.7 y 2.8.6 respectivamente. Esto se hace así para

poder comparar sus efectos sobre el espectro y determinar si una técnica es más efectiva que otra.

4.3.1. Filtrado con wavelets

Para realizar es filtrado con transformada wavelet, se realizó un código en Python, como se puede ver en la figura 4.8. En este se utiliza como wavelet madre a la Daubechies 1, la cual es la más simple de todas, y también se realizan 8 niveles de descomposición.

```
import pywt

WAVELET= 'db1'

def denoise(time_series, absolute=True): 1 usage
    coeffs = pywt.wavedec(time_series,WAVELET , level=8, mode="symmetric")
    sigma = np.median(np.abs(coeffs[8]))/0.6745
    for i in range(6,9):
        D = coeffs[i]
        n = len(D)
        tVisu = np.sqrt(2*np.log(n))*sigma
        coeffs[i] = pywt.threshold(D, tVisu, mode='hard')
        time_series_rec = pywt.waverec(coeffs,WAVELET , mode="symmetric")
        if absolute:
            time_series_rec = np.abs(time_series_rec)
    return time_series_rec

file = "Clorofila/Espectro_Clорofila2.csv"
```

Figura 4.8: Código de filtrado con transformada wavelet

El espectro utilizado para realizar las pruebas de filtrado es el de la clorofila, ya que fue de las muestras que presentaron más ruido. La figura 4.9 se puede apreciar al espectro de la clorofila antes de ser filtrado (gráfica de la izquierda) y después de ser filtrado (gráfica de la derecha). Se puede observar una reducción significativa del ruido de alta frecuencia después de realizar el filtrado, generando una silueta más definida y fácil de interpretar. Sin embargo, en la grafica del espectro filtrado aparece un efecto conocido como error por cuantización, lo cual provoca lineas no tan suaves formando el espectro, y si bien esto no resulta algo grave en espectroscopía visible, si puede afectar en otros tipos de espectroscopías, donde este efecto pueden ser considerados como picos detectados por el equipo, lo que alteraría la interpretación del usuario.

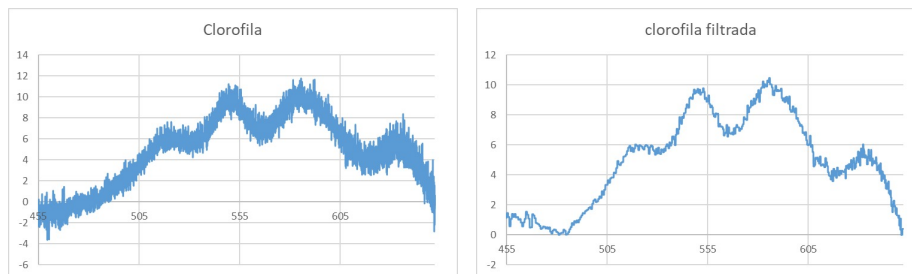


Figura 4.9: Espectros clorofila antes y después de ser filtrada con transformada wavelet

4.3.2. Filtrado con Savitzky-Golay

Para utilizar el algoritmo de Savitzky-Golay, python ya cuenta con una librería especializada, en donde solamente se deben ingresar los parámetros deseados por el usuario en la llamada a la función del filtro, como se puede ver en la figura 4.10.

```
import pandas as pd
import matplotlib.pyplot as plt
from scipy.signal import savgol_filter

# Cargar datos del CSV
df = pd.read_excel('LectorEspectrometro3/CS61.xlsx') # Cambia 'datos.csv' por el nombre real
# Suponiendo que tienes columnas 'tiempo' y 'valor'
Longitud_onda = df['LongitudDeOnda']
valor = df['Valor']

# Aplicar filtro Savitzky-Golay
valor_suavizado = savgol_filter(valor, window_length=91, polyorder=3)
```

Figura 4.10: Código de filtrado con Savitzky-Golay

Para este caso se decidió que el polinomio fuera de orden 3, y que el valor de la ventana fuera de 21. Con esto se obtuvo una buena respuesta, reduciendo significativamente el ruido del espectro, aunque no eliminándolo por completo, como se puede ver en la figura 4.11.

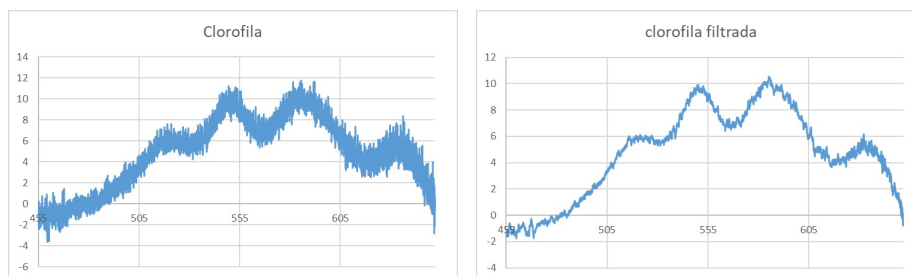


Figura 4.11: Espectros clorofila antes y después de ser filtrada con Savitzky-Golay

El suavizado de la señal resultante después del filtrado puede mejorarse incrementando el valor de la ventana, aunque se debe ser cuidadoso con esto, ya que se podría llegar a eliminar picos importantes del espectro. La figura 4.12 muestra los resultados obtenidos con diferentes valores de ventana para el filtro.

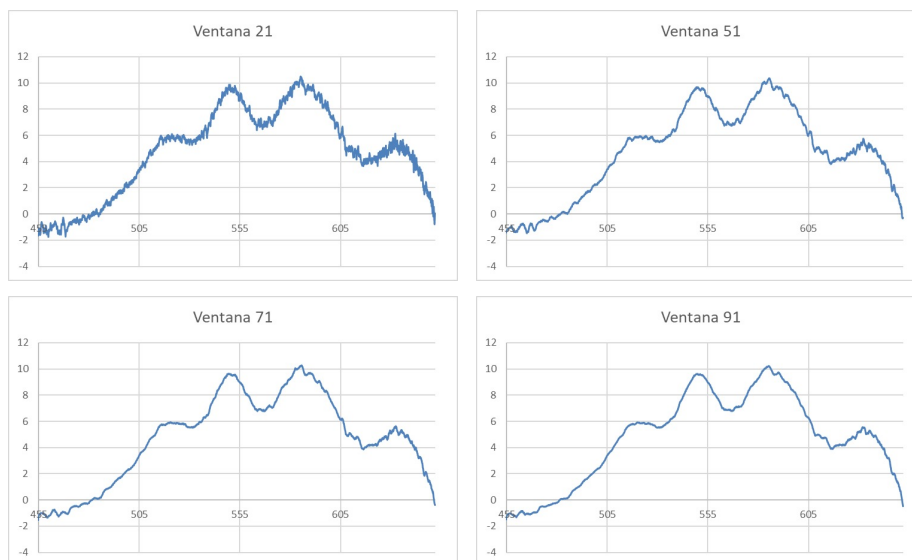


Figura 4.12: Espectros de clorofila filtrados con valores de ventana de 21, 51, 71 y 91

4.4. Implementación de técnicas de filtrado durante la integración del espectro

Como se mencionó en la sección 1.5.1, lo que se busca es realizar el filtrado del ruido durante la integración del espectro, para lo cual se debe adaptar este

proceso en el código de Python del receptor. Esto es bastante sencillo, pues con una simple función if se puede seleccionar entre una técnica de filtrado u otra, como se ve en la figura 4.13. Si la variable "Filtro" es igual a 1 entonces se hace el filtrado con wavelets y si es 2 se hace con Savitzky-Golay, y los resultados obtenidos se irán acumulando hasta llegar al número de espectros determinados por el usuario, para después finalmente promediarlos (para ver el código completo del receptor consulte el apéndice C.1).

```
# Suma los nuevos datos con los datos anteriores
if Filtro == 1:
    data_array_Temporal += denoise(time_series=int_data)
    data_array_Temporal += int_data
    data_array = data_array_Temporal[32:3680]

if Filtro == 2:
    data_array_Temporal += savgol_filter(int_data, window_length=91, polyorder=3)
    data_array = data_array_Temporal[32:3680]
```

Figura 4.13: Selección del tipo de filtro a utilizar

Por lo anterior, los pasos para capturar un espectro son en el siguiente orden:

- Seleccionar y configurar el tipo de filtro en el código: Este paso es muy importante, ya que con esto establecemos la técnica con la que se va a trabajar y como es que se va a aplicar a los espectros.
- Configurar el número de espectros y el tiempo de integración: Se debe tener un correcto manejo de estos dos parámetros, pues establecer valores muy altos para ambos implicaría que la captura del espectro tomara demasiado tiempo, lo que también puede dar oportunidad a captar más interferencias que alteren el resultado final. Estos se deben establecer dependiendo de las necesidades del usuario.
- Captura de espectro de oscuridad: Para garantizar una buena calibración es necesario primero capturar un espectro sin incidir ningún tipo de luz al equipo, además de que también es importante que este pasa por el filtrado seleccionado en el paso anterior, esto con el objetivo de restar esos datos a los espectros de las muestras capturados para eliminar el offset y cualquier otro tipo de distorsión que pueda estar presente en los espectros.
- Capturar el espectro de la muestra: En este paso es donde se realiza la captura del espectro de la muestra y sea realizan los procesos de filtrado, mostrando en una grafica como es que se va formando el espectro y guardando los datos en un archivo CSV.

La figura 4.14 muestra la comparación de los espectros obtenidos de la clorofila al realizar el filtrado con wavelets después (gráfica de la izquierda) y durante (gráfica de la derecha) la integración del espectro. Se puede apreciar como el

error de cuantización presente en el espectro filtrado después de la integración se redujo en el espectro filtrado durante la integración, lo que indica una clara mejoría al capturar los espectros de esta forma.

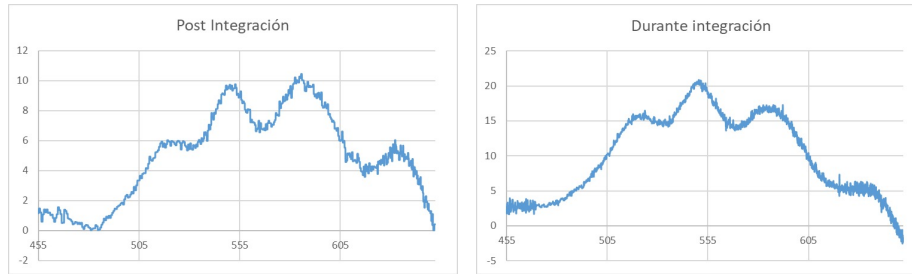


Figura 4.14: Clorofila filtrada con wavelets después y durante la integración del espectro

Un efecto similar se puede apreciar en la figura 4.15, en donde se aplica el filtrado con Savitzky-Golay a la clorofila después y durante la integración del espectro.

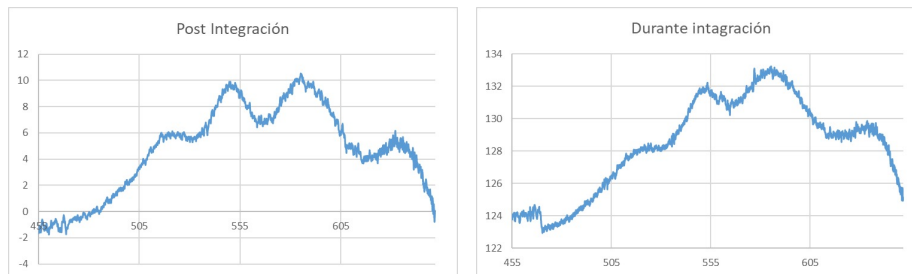


Figura 4.15: Clorofila filtrada con Savitzky-Golay después y durante la integración del espectro

Se debe tener en cuenta que la respuesta de los filtros aplicados puede cambiarse dependiendo de como se configuren sus parámetros, por lo que para obtener el mejor resultado posible se deben realizar varias pruebas hasta llegar a obtener un espectro que cubra con lo que busca el usuario.

4.5. Conclusiones

En este capítulo se mostraron los resultados obtenidos con el equipo construido, utilizando diferentes muestras, mayormente líquidas. Además se identificó el ruido presente en los espectros y como es que este puede ser reducido utilizando dos técnicas diferentes, comparando sus resultados al aplicarlos durante y después de la integración de los espectros.

Capítulo 5

Interpretación de resultados

5.1. Introducción

En este capítulo se realiza un análisis de los resultados obtenidos en la sección anterior mediante el uso de los métodos estadísticos del SNR y el PCA, para así determinar si hay una mejoría en la reducción del ruido al aplicar los filtros durante la integración del espectro, y si es posible clasificar las muestras según su espectro.

5.2. Obtención del SNR

Como se mencionó en la sección 2.5.3, el SNR es bastante útil en la espectroscopía para identificar la calidad de los espectros obtenidos. En este caso, el SNR es utilizado para realizar una comparativa de la eficiencia de las técnicas de filtrado utilizadas, wavelets y Savitzky-Golay, además de también comparar su eficiencia según el momento en que son aplicadas, durante y después de la integración del espectro.

El obtener el SNR es bastante sencillo, pues solo consiste en dividir el valor promedio de la señal entre su desviación estándar, para lo cual se realizó un código de python para hacerlo, como se puede ver en la figura 5.1 (para ver el código completo para calcular el SNR consulte el apéndice E.1). En este se extraen los datos del espectro del archivo CSV creado, para después obtener su promedio cuadrático y su desviación estándar mediante el uso de funciones disponibles en la librería numpy, después estos datos se dividen y se calcula el valor en decibelios.

```

import numpy as np
import pandas as pd

# Cargar los datos desde un archivo CSV
file_path = "LectorEspectrometro3/CSG1.CSV" # Cambia esta ruta a tu archivo CSV
df = pd.read_excel(file_path)

# Extraer los datos de la columna 'valor3'
data = np.array(df['Valor'])

# 1. Calcula la potencia de la señal (promedio cuadrático)
signal_power = np.mean(data**2)

# 2. Calcula el ruido (desviación estándar)
noise_power = np.std(data)

# 3. Calcula el SNR (relación señal-ruido)
snr = signal_power / noise_power

# 4. SNR en dB
snr_db = 10 * np.log10(snr)

print(f"SNR = {snr_db:.4f} dB")

```

Figura 5.1: Código de python para obtención del SNR

Para las pruebas se decidió seguir utilizando los espectros obtenidos de la muestra de clorofila, mostrados en la sección 4.3. En la tabla 5.1 se muestran los valores de SNR obtenidos para cada tipo de filtrado aplicado a los espectros durante y después de la integración.

Tipo de filtrado	SNR
Sin filtrar	10.33dB
Wavelets post integración	10.76dB
Wavelets durante integración	13.24dB
Savitzky-Golay post integración	10.38dB
Savitzky-Golay durante integración	37.89dB

Tabla 5.1: Resultados SNR filtros aplicados

Como se puede apreciar, la técnica que presenta un mejor SNR es el espectro filtrado con Savitzky-Golay durante la integración del espectro, lo que indica que realizar el filtrado durante la integración del espectro genera una mejoría en la reducción del ruido presente, esto incluso es claro con el filtrado usando wavelets, al comparar los SNR del espectro sin filtrar, filtrado después y durante la integración. También es importante entender que el valor del SNR puede variar según el tipo de muestra a analizar, ya que el ruido en el espectro puede

ser mayor o menor dependiendo de esta, y es por esto mismo que la configuración de los filtros también puede cambiar, requiriendo que estos actúen en mayor o menor medida hasta obtener un espectro que sea útil para el usuario.

5.3. Obtención del PCA

Como se mencionó en la sección 2.5.3, el PCA es una herramienta que puede dar una visión más clara de como se agrupan en zonas específicas de una gráfica los datos de los espectros dependiendo del tipo de muestra utilizada. Esto significaría que con la información obtenida con el espectrómetro se puede clasificar los espectros según la muestra al que pertenecen.

Para poder obtener el PCA, primero se debe contar con datos para poder realizar la comparativa, en esta caso se decidió utilizar tres muestras, la clorofila, el café soluble y el líquido limpia vidrios, a las cuales se les tomó cinco espectros sin realizar ningún tipo de filtrado, cinco espectros filtrados con wavelets y cinco con Savitzky-Golay, lo que daría un total de 45 espectros para trabajar.

Para realizar el PCA se desarrollo un código de Python, donde mediante el uso de funciones disponibles en la librería sklern se realizó todo el proceso necesario(para ver el código para la obtención del PCA consulte el apéndice D.1). La figura 5.2 muestra la gráfica PCA obtenida de los espectros no filtrados, en la cual se pude apreciar puntos de colores diferentes, azules, naranjas o verdes, los cuales corresponden al tipo de muestra, siendo clorofila, café soluble y liquido limpia vidrios respectivamente. Se puede apreciar como es que los puntos se van agrupando en diferentes zonas de la grafica, quedando mayormente juntos los que son del mismo color.

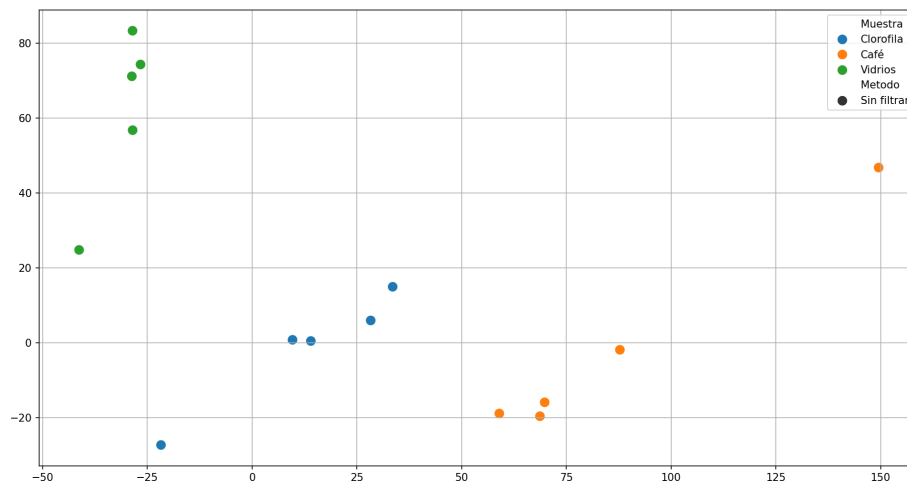


Figura 5.2: PCA sin filtrar

De forma similar, en la figura 5.3 se muestra la gráfica PCA de los espectros

cuando son filtrados con wavelets durante la integración de los mismos, cambiando ligeramente la manera en que los puntos se agruparon según el espectro de la muestra a la que pertenecen.

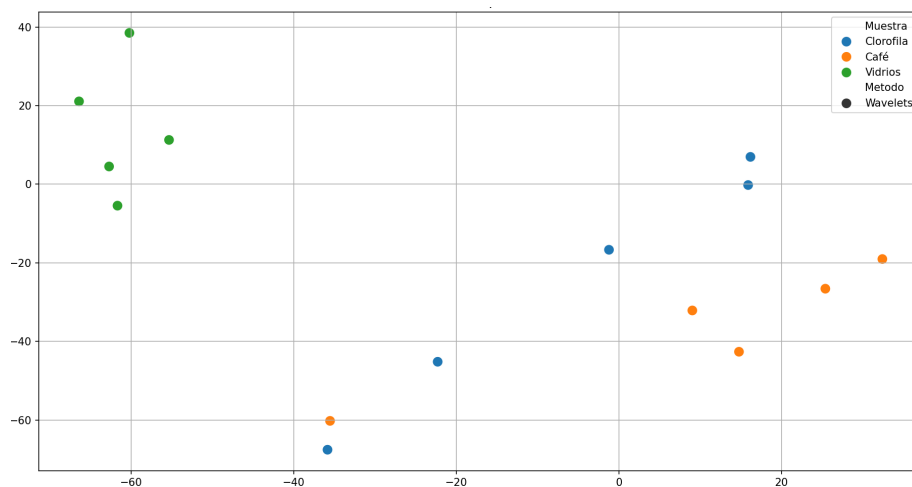


Figura 5.3: PCA Wavelets

La figura 5.4 muestra la gráfica PCA de los espectros filtrados con el algoritmo Savitzky-Golay durante la integración del espectro. En este caso se puede apreciar que los resultados del café soluble están más dispersos que en la gráfica anterior.

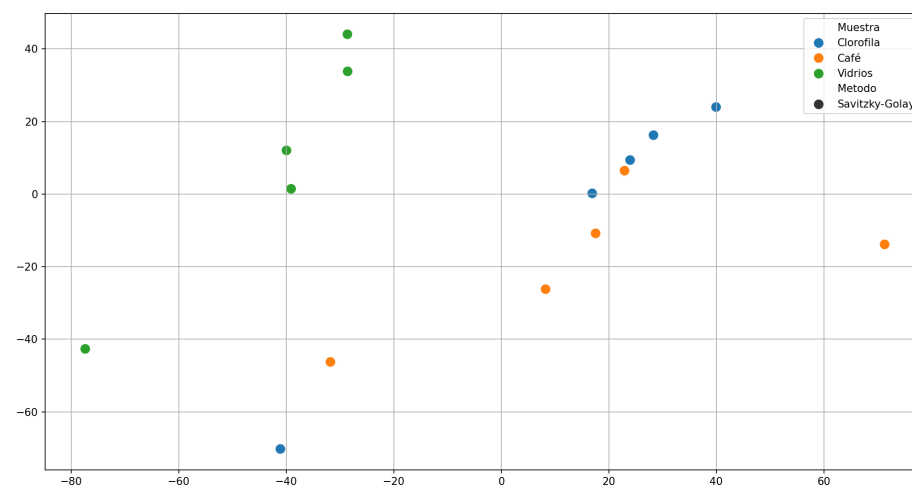


Figura 5.4: PCA Savitzky-Golay

Finalmente, en la figura 5.5 se colocan los restados de las tres gráficas PCA

anteriores, en donde, además de los tres colores mostrados anteriormente, se puede apreciar figuras de tres tipos, círculos, cruces y cuadrados, los cuales representan el tipo de filtrado, siendo sin filtrar, con wavelets y con Savitzky-Golay respectivamente.

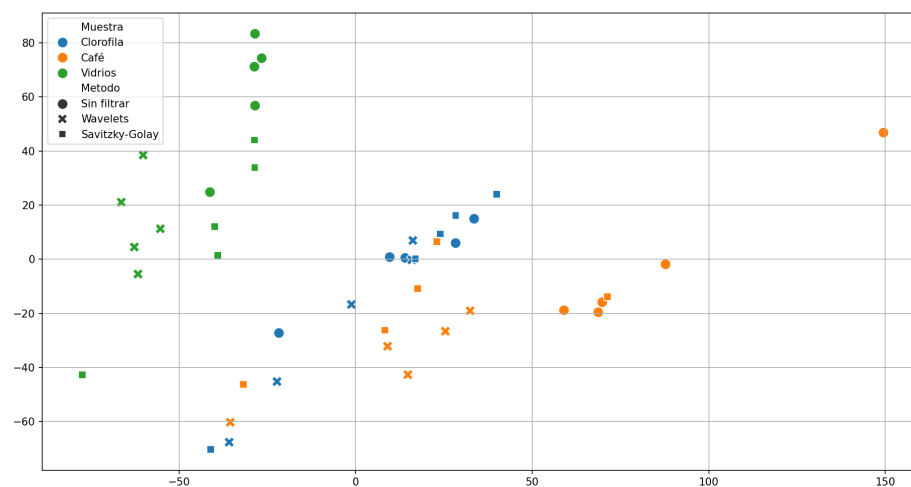


Figura 5.5: PCA global

En esta grafica se puede apreciar que tan bien los espectros se pudieron agrupar según la muestra a la que pertenecen dependiendo del filtro aplicado, siendo los espectros del líquido de limpia vidrios los que mejor lo lograron, mientras que los espectros del café soluble son los que se encuentran mas dispersos, llegan a invadir en la zona en donde se encuentran los de la clorofila. Algo similar pasa al observar como se agruparon los espectros según la técnica de filtrado aplica, siendo nuevamente el liquido limpia vidrios con el que se ve mas claro al agrupar de forma más precisa los espectros sin filtrar y a los filtrados con wavelets, siendo los filtrados con Savitzky-Golay los que se encuentran más dispersados, mientras que en el caso de la clorofila y del café soluble es más difícil notar como es que se agrupan los espectros. A pesar de haber utilizado pocos espectros los resultados fueron satisfactorios, ya que el poder conseguir una mejor respuesta al agrupar la información requeriría tomar una mayor cantidad de espectros, además de tener que montar todo el sistema óptico del espectrómetro sobre alguna estructura que fije la posición de los elementos, ya que al no estar fijos se pueden generar cambios que afecten a los espectros, haciendo que varíen demasiado entre ellos mismos.

5.4. Conclusiones

En este capítulo se realizó un análisis de los espectros obtenidos con el espectrómetro desarrollado, haciendo uso de las técnicas SNR y PCA, y con las

cuales se pudo llegar a la conclusión que el trabajo desarrollado cumple con los objetivos propuestos, quedando incluso la posibilidad de mejorar aún más los resultados obtenidos al trabajar con diferentes formas de configuración de los parámetros de las técnicas de filtrado aplicadas junto con la cantidad de espectros capturados y el tiempo de integración de los mismos.

Capítulo 6

Comentarios Finales

6.1. Introducción

El objetivo de este capítulo final es dar un cierre a todo lo desarrollado en las secciones previas, hablando de los puntos clave y de las dificultades presentadas y como es que se logró superarlas para desarrollar este trabajo, además de explicar como es que se cumplieron los objetivos establecidos y hacer mención del trabajo futuro que deben seguir desarrollándose.

6.2. Cumplimiento de los objetivos

Recordando lo mencionado en la sección 1.5, el objetivo general que se había establecido era el desarrollo de un espectrómetro de rango visible que contara con el hardware y algoritmos necesarios para la eliminación del ruido de alta frecuencia durante la integración de espectros de absorción aplicado a materiales translúcidos con fines de clasificación automática. Se considera que este objetivo se a cumplido de forma satisfactoria, ya que en los capítulos 3, 4 y se muestra el proceso de desarrollo del hardware y software, en el cual se implementaron dos técnicas de filtrado, además de comparar los resultados de filtrar el ruido de los espectros después y durante la integración. De igual manera, gracias a lo mostrado en estos capítulo junto con el capítulo 5, los objetivos particulares son cubiertos de forma satisfactoria, algunos de estos son el desarrollo del hardware y software, implementar las técnicas de filtrado, hacer comparaciones de los resultados, implementar un clasificador básico, etc.

6.3. Dificultades presentadas

Varias dificultades se presentaron durante el desarrollo de este trabajo, muchas de las cuales estaban relacionadas con la disponibilidad de los elementos necesarios para la construcción de el espectrómetro, como el sensor CCD, el

optical bench, incluso las placas PCB para la implementación del circuito de control ya que todo esto eran productos de importación, lo implicaba tener que cubrir los costos de envío y los impuestos para que estos pudieran ingresar al país, sin mencionar el largo tiempo de espera, lo que retrasaba significativamente los avances hasta que los productos llegaran.

Otra etapa en la que se presentaron varias dificultades fue durante la programación del microcontrolador y del receptor, ya que el hacer que estos dos pudieran trabajar correctamente en conjunto llevó a largos procesos de experimentación, teniendo que realizar cambios constantes en la programación hasta conseguir el resultado presentado.

El capturar los espectros fue otra de las dificultades presentadas durante el desarrollo, ya que al no estar fijado el espectrómetro en alguna estructura, la captura de los espectros era una actividad algo complicada debido a las variaciones de las posiciones de los elementos ópticos y de la fuente de luz, haciendo que los datos obtenidos no siempre fueran útiles, lo que llevaba a tener que hacer varias capturas hasta obtener un buen espectro de la muestra.

A pesar de todo, el trabajo realizado cumple de manera satisfactoria con lo que se había propuesto, dejando abierta la posibilidad de seguir mejorando lo ya realizado.

6.4. Trabajo futuro

A pesar de haber cumplido con los objetivos, y de haber superado las dificultades presentadas durante el desarrollo del trabajo, aún hay bastantes puntos que deben seguir desarrollándose, pero que debido a que ya no se cuenta con tiempo suficiente ya no pudo seguir trabajándose.

El primero de estos puntos es el modificar el circuito de control, ya que como se mencionó en el capítulo 3 y 4, los espectros capturados presentan un offset, el cual es eliminado mediante software en la programación del receptor, pero que en ocasiones los niveles de amplitud son tan bajos que al momento de restar el offset genera que se tengan valores negativos en algunas secciones del espectro, para lo cual se proponía modificar el circuito de control, específicamente la sección del OPAM encargado de amplificar la señal de salida del circuito, para arreglar este problema. Otra razón para modificar el circuito de control, es para poderlo hacer más pequeño, y de esta manera que este implementado en una sola PCB montada sobre el optical bench en lugar de dos PCB's separadas.

Otro punto que quedó trabajo futuro es probar el espectrómetro con otros tipos de espectroscopía, ya que algo que se busca es que dispositivo pueda ser compatible para trabajar más allá del espectro visible, aún cuando el el objetivo del trabajo se limitaba a que funcionara solamente dentro de este rango. Durante la etapa de pruebas se probó el espectrómetro con una configuración Raman, lamentablemente no se logró capturar ningún espectro de forma satisfactoria, ya que el TCD1304DG no cuenta con la sensibilidad suficiente para captar dichos espectros, por lo que sería necesario probar con otro sensor CCD, pero debido a que ya no se contaba con más tiempo no pudo realizarse ese cambio y probarlo.

La última cosa que quedó como trabajo futuro es fijar todo el arreglo óptico y la fuente de luz en alguna estructura o carcasa para evitar desajustes de los elementos, ya que esto provocaba variaciones en los espectros capturados y obligaba a tener que capturarlos de nuevo.

6.5. Conclusión

Para el desarrollo de este trabajo se aplicaron los conocimientos adquiridos en el programa de maestría en ciencias de la ingeniería del Instituto Tecnológico de Aguascalientes, y gracias a los cuales se brindó una alternativa a la manera en que son capturados los datos en la espectroscopía, mostrando el desarrollo de un dispositivo sencillo pero capaz de trabajar en uno de los tipos de espectroscopía más utilizadas, pudiendo reducir significativamente las dificultades que traen consigo los equipos comerciales, como lo es el difícil manejo, la portabilidad y la incapacidad de implementar técnicas de filtrado durante la integración de los espectros.

Bibliografía

- [1] Peter J. Larkin. *Infrared and raman spectroscopy: principles and spectral interpretation*. Elsevier, Amsterdam, 2e ed edition.
- [2] David W. Ball. *The basics of spectroscopy*. Number v. TT 49 in Tutorial texts in optical engineering. SPIE- The International Society for Optical Engineering, Bellingham, WA, 2001.
- [3] Richard L. MacCreery. *Raman spectroscopy for chemical analysis*. Number v. 157 in Chemical analysis. John Wiley & Sons, New York, 2000.
- [4] Isabel Cristina Castellanos Cuéllar, Javier Ricardo Velandia Cabra, Miguel Ángel González Curbelo, Diana Angélica Varela Martínez, and Eduardo Ramírez Valencia. *Aplicaciones y generalidades de un espectrofotómetro UV-VIS UV-1800 de Shimadzu*. Universidad EAN, Bogotá, Colombia, 2018.
- [5] Jorge Castro Ramos, Adrián E Villanueva Luna, and Carlos M Ortiz Lima. Espectroscopia raman y sus aplicaciones. *Opt. Pura. Apl*, pages 83–95, 2013.
- [6] Yohan Pérez-Moret. *Diseño y construcción de un espectrómetro óptico por USB*. Yohan Pérez-Moret, 2012.
- [7] Victor Manuel Soto Alexander. *Espectroscopia Raman: Tratamiento de ruido asociado con fluorescencia*. PhD thesis, Universidad Nacional Autónoma de México, 2020.
- [8] Sergio Iván Padrón-Ortega, Ernestina Elizabeth Banda-Cruz, and Nohra Violeta Gallardo-Rivas. Application of savitzky-golay and fast fourier transform filters in the processing of derivative spectra obtained from asphaltene solutions espectroscopía uv-vis derivada para asfaltenos. *CienciaUAT*, 18(2):170–182, 2024.
- [9] Ángel Gabriel Vega Artiles et al. Análisis y tratamiento de datos representativos del instrumento supercam de la mision mars 2020. 2022.
- [10] Antonio Bri Pérez. Eliminación de ruido en señales de espectroscopía infrarroja usando redes neuronales. 2022.

- [11] Yanru Bai and Quan Liu. Denoising raman spectra by wiener estimation with a numerical calibration dataset. *Biomedical Optics Express*, 11(1):200–214, 2019.
- [12] Sanjay D Deshmukh, Rahul Pandya, Tarush Singh, Arpit Singh, and Ashish Sharma. Building a low-cost, low-power and portable spectrophotometer for chemical analysis. *Solid State Technology*, 64(2):3079–3089, 2021.
- [13] Jorge Andrés Jiménez Salazar and Solange Ivette Rivera Manrique. Propuesta de diseño para la construcción de un espectrofotómetro uv-vis a base arduino y fibra óptica. *JÓVENES EN LA CIENCIA*, 28:1–16, 2024.
- [14] Omid Badkoobe Hezave and Seyed Hassan Tavassoli. Design and fabrication of a dual laser raman spectrometer with a single one-dimensional ccd detector. *Scientific Reports*, 14(1):28184, 2024.
- [15] Francisco Jacob Avila-Camacho, Leonardo Miguel Moreno-Villalba, Genaro Iván Cerón-Montes, Aristeo Garrido-Hernández, Cesar Andrés Cardoso-Chávez, Carlos Jossef Pacheco-Piña, et al. Diseño de un espectrofotómetro uv-vis de bajo costo para la industria bioquímica: Una revisión. *Pädi Boletín Científico de Ciencias Básicas e Ingenierías del ICBI*, 9:19–28, 2021.
- [16] Vilma Nohemí Yanchapanta Bastidas, Rosa Maricela Ormaza Hugo, Miguel Ángel Sáez Paguay, and Richard Willians Pachacama Choca. Construcción de un espectrómetro casero para el análisis de espectros de luz, mediante el software spectral workbench. *Polo del Conocimiento: Revista científico-profesional*, 5(7):209–222, 2020.
- [17] Eugene Hecht. *Optics*. Pearson, Boston Columbus Indianapolis New York San Francisco Amsterdam Cape Town Dubai London Madrid Milan Munich, 5 ed/fifth edition, global edition edition, 2017.
- [18] Joseph R. Lakowicz. *Principles of fluorescence spectroscopy*. Springer, New York, 3rd ed edition, 2006.
- [19] Steven W. Smith. *Digital Signal Processing: A Practical Guide for Engineers and Scientists*. Newnes, Oxford, UK, 2002.
- [20] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning: with Applications in R*. Springer Texts in Statistics. Springer, 2013.
- [21] Sasan Karamizadeh, Shahidan M Abdullah, Azizah A Manaf, Mazdak Zamani, and Alireza Hooman. An overview of principal component analysis. *Journal of signal and information processing*, 4(3):173–175, 2013.
- [22] Heinz-Helmut Perkampus. *UV-VIS Spectroscopy and Its Applications*. Springer, 1992.

- [23] John R. Ferraro, Kazuo Nakamoto, and Chris W. Brown. *Introductory Raman spectroscopy*. Academic Press, Amsterdam ; Boston, 2nd ed edition, 2003.
- [24] Richard G. Lyons. *Understanding Digital Signal Processing*. Prentice Hall, Upper Saddle River, NJ, 3rd edition, 2010.
- [25] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, 3rd edition, 2007.
- [26] David L. Donoho. De-noising by soft-thresholding. *IEEE Transactions on Information Theory*, 41(3):613–627, 1995.
- [27] Juan A Conesa. Interpolación. *Técnicas de Cálculo en Ingeniería Química*, 2011.
- [28] Ivet Challenger-Pérez, Yanet Díaz-Ricardo, and Roberto Antonio Becerra-García. El lenguaje de programación python. *Ciencias Holguín*, 20(2):1–13, 2014.
- [29] eBay Seller: spectrophoton. Optical bench czerny turner, 2025. Consultado el 6 de junio de 2025.
- [30] Sony Corporation. Ilx511 2048-pixel cdd linear image sensor (b/w). Technical report, Sony Corporation, 2025. Datasheet consultado el 6 de junio de 2025.
- [31] Hamamatsu Photonics. Linear sensors for spectrophotometry, 2025. Consultado el 11 de junio de 2025.
- [32] Toshiba Electronic Devices & Storage Corporation . Tcd1304dg toshiba ccd linear image sensor. Technical report, Toshiba Electronic Devices & Storage Corporation, 2025. Datasheet consultado el 11 de junio de 2025.
- [33] STMicroelectronics. Stm32f401cc, 2025. Consultado el 13 de junio de 2025.
- [34] WeActStudio. Blackpill, 2025. Repositorio en GitHub.

Apéndice A

Código fuente STM32

```
1  /* USER CODE BEGIN Header */
2  /**
3   *
4   * @file          : main.c
5   * @brief         : Main program body
6   *
7   * @attention
8   *
9   * Copyright (c) 2024 STMicroelectronics.
10  * All rights reserved.
11  *
12  * This software is licensed under terms that can be found in
13  * the LICENSE file
14  * in the root directory of this software component.
15  * If no LICENSE file comes with this software, it is provided
16  * AS-IS.
17  *
18  *
19  *
20  *
21  *
22  *
23  *
24  *
25  *
26  *
27  *
28  *
29  *
30  *
31  *
32  */
33  /* USER CODE END Header */
34  /* Includes -----*/
35  #include "main.h"
36  #include "usb_device.h"
37  /* Private includes -----
38  */
39  /* USER CODE BEGIN Includes */
40
41  /* USER CODE END Includes */
42
43  /* Private typedef
44  -----
45  */
46
47  /* USER CODE BEGIN PTD */
48
49  /* USER CODE END PTD */
```

```

33  /* Private define
    -----
    */
34  /* USER CODE BEGIN PD */
35
36  /* USER CODE END PD */
37
38  /* Private macro
    -----
    */
39  /* USER CODE BEGIN PM */
40
41  /* USER CODE END PM */
42
43  /* Private variables
    -----*/
44  ADC_HandleTypeDef hadc1;
45  DMA_HandleTypeDef hdma_adc1;
46
47  TIM_HandleTypeDef htim2;
48  TIM_HandleTypeDef htim3;
49  TIM_HandleTypeDef htim4;
50  TIM_HandleTypeDef htim5;
51
52  /* USER CODE BEGIN PV */
53
54  /* USER CODE END PV */
55
56  /* Private function prototypes
    -----*/
57  void SystemClock_Config(void);
58  static void MX_GPIO_Init(void);
59  static void MX_DMA_Init(void);
60  static void MX_TIM3_Init(void);
61  static void MX_TIM2_Init(void);
62  static void MX_TIM5_Init(void);
63  static void MX_ADC1_Init(void);
64  static void MX_TIM4_Init(void);
65  /* USER CODE BEGIN PFP */
66
67  /* USER CODE END PFP */
68
69  /* Private user code
    -----*/
70  /* USER CODE BEGIN 0 */
71  #define CCDBuffer 3770
72  volatile uint16_t CCDPixelBuffer[CCDBuffer];
73  /* USER CODE END 0 */
74
75  /**
76   * @brief The application entry point.
77   * @retval int
78   */
79  int main(void)
80  {
81      /* USER CODE BEGIN 1 */
82

```

```

83      /* USER CODE END 1 */
84
85      /* MCU Configuration
      -----
      */
86
87      /* Reset of all peripherals, Initializes the Flash
      interface and the Systick. */
88      HAL_Init();
89
90      /* USER CODE BEGIN Init */
91
92      /* USER CODE END Init */
93
94      /* Configure the system clock */
95      SystemClock_Config();
96
97      /* USER CODE BEGIN SysInit */
98
99      /* USER CODE END SysInit */
100
101      /* Initialize all configured peripherals */
102      MX_GPIO_Init();
103      MX_USB_DEVICE_Init();
104      MX_TIM3_Init();
105      MX_TIM4_Init();
106      MX_TIM2_Init();
107      MX_DMA_Init();
108      MX_ADC1_Init();
109      MX_TIM5_Init();
110      /* USER CODE BEGIN 2 */
111      HAL_TIM_PWM_Start(&htim3, TIM_CHANNEL_1); //fM
112      HAL_TIM_PWM_Start(&htim4, TIM_CHANNEL_4); //ADC
113      HAL_TIM_PWM_Start(&htim2, TIM_CHANNEL_1); //ICG
114      __HAL_TIM_SET_COUNTER(&htim2, 66); //600NS DELAY
115      HAL_TIM_PWM_Start(&htim5, TIM_CHANNEL_3); //SH
116      HAL_TIM_Base_Start_IT(&htim2);
117      /* USER CODE BEGIN 2 */
118
119      /* USER CODE END 2 */
120
121      /* Infinite loop */
122      /* USER CODE BEGIN WHILE */
123      while (1)
124      {
125          /* USER CODE END WHILE */
126
127          /* USER CODE BEGIN 3 */
128      }
129      /* USER CODE END 3 */
130  }
131
132  /**
133   * @brief System Clock Configuration
134   * @retval None
135   */
136  void SystemClock_Config(void)

```

```

137 | {
138 |     RCC_OscInitTypeDef RCC_OscInitStruct = {0};
139 |     RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};
140 |
141 |     /** Configure the main internal regulator output voltage
142 |     */
143 |     __HAL_RCC_PWR_CLK_ENABLE();
144 |     __HAL_PWR_VOLTAGESCALING_CONFIG(
145 |         PWR_REGULATOR_VOLTAGE_SCALE2);
146 |
147 |     /** Initializes the RCC Oscillators according to the
148 |         specified parameters
149 |         * in the RCC_OscInitTypeDef structure.
150 |     */
151 |     RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSE;
152 |     RCC_OscInitStruct.HSEState = RCC_HSE_ON;
153 |     RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
154 |     RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSE;
155 |     RCC_OscInitStruct.PLL.PLLM = 25;
156 |     RCC_OscInitStruct.PLL.PLLN = 336;
157 |     RCC_OscInitStruct.PLL.PLLP = RCC_PLLP_DIV4;
158 |     RCC_OscInitStruct.PLL.PLLQ = 7;
159 |     if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
160 |     {
161 |         Error_Handler();
162 |     }
163 |
164 |     /** Initializes the CPU, AHB and APB buses clocks
165 |     */
166 |     RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_HCLK |
167 |         RCC_CLOCKTYPE_SYSCLK
168 |         |RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;
169 |     RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
170 |     RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
171 |     RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV2;
172 |     RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV1;
173 |
174 |     if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_2
175 |         ) != HAL_OK)
176 |     {
177 |         Error_Handler();
178 |     }
179 | }
180 |
181 | /**
182 | * @brief ADC1 Initialization Function
183 | * @param None
184 | * @retval None
185 | */
186 | static void MX_ADC1_Init(void)
187 | {
188 |
189 |     /* USER CODE BEGIN ADC1_Init 0 */
190 |
191 |     /* USER CODE END ADC1_Init 0 */
192 |
193 |     ADC_ChannelConfTypeDef sConfig = {0};

```

```

190
191     /* USER CODE BEGIN ADC1_Init 1 */
192
193     /* USER CODE END ADC1_Init 1 */
194
195     /** Configure the global features of the ADC (Clock,
196         Resolution, Data Alignment and number of conversion)
197     */
198     hadc1.Instance = ADC1;
199     hadc1.Init.ClockPrescaler = ADC_CLOCK_SYNC_PCLK_DIV4;
200     hadc1.Init.Resolution = ADC_RESOLUTION_12B;
201     hadc1.Init.ScanConvMode = DISABLE;
202     hadc1.Init.ContinuousConvMode = DISABLE;
203     hadc1.Init.DiscontinuousConvMode = DISABLE;
204     hadc1.Init.ExternalTrigConvEdge =
205         ADC_EXTERNALTRIGCONVEDGE_RISING;
206     hadc1.Init.ExternalTrigConv = ADC_EXTERNALTRIGCONV_T4_CC4;
207     hadc1.Init.DataAlign = ADC_DATAALIGN_RIGHT;
208     hadc1.Init.NbrOfConversion = 1;
209     hadc1.Init.DMAContinuousRequests = ENABLE;
210     hadc1.Init.EOCSelection = ADC_EOC_SINGLE_CONV;
211     if (HAL_ADC_Init(&hadc1) != HAL_OK)
212     {
213         Error_Handler();
214     }
215
216     /** Configure for the selected ADC regular channel its
217         corresponding rank in the sequencer and its sample time
218     */
219     sConfig.Channel = ADC_CHANNEL_3;
220     sConfig.Rank = 1;
221     sConfig.SamplingTime = ADC_SAMPLETIME_3CYCLES;
222     if (HAL_ADC_ConfigChannel(&hadc1, &sConfig) != HAL_OK)
223     {
224         Error_Handler();
225     }
226     /* USER CODE BEGIN ADC1_Init 2 */
227
228     /* USER CODE END ADC1_Init 2 */
229
230 }
231
232 /**
233  * @brief TIM2 Initialization Function
234  * @param None
235  * @retval None
236  */
237 static void MX_TIM2_Init(void)
238 {
239     /* USER CODE BEGIN TIM2_Init 0 */
240
241     /* USER CODE END TIM2_Init 0 */
242
243     TIM_ClockConfigTypeDef sClockSourceConfig = {0};
244     TIM_MasterConfigTypeDef sMasterConfig = {0};

```

```

243     TIM_OC_InitTypeDef sConfigOC = {0};
244
245     /* USER CODE BEGIN TIM2_Init 1 */
246
247     /* USER CODE END TIM2_Init 1 */
248     htim2.Instance = TIM2;
249     htim2.Init.Prescaler = 0;
250     htim2.Init.CounterMode = TIM_COUNTERMODE_UP;
251     htim2.Init.Period = 630000-1;
252     htim2.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
253     htim2.Init.AutoReloadPreload =
        TIM_AUTORELOAD_PRELOAD_DISABLE;
254     if (HAL_TIM_Base_Init(&htim2) != HAL_OK)
255     {
256         Error_Handler();
257     }
258     sClockSourceConfig.ClockSource = TIM_CLOCKSOURCE_INTERNAL;
259     if (HAL_TIM_ConfigClockSource(&htim2, &sClockSourceConfig)
        != HAL_OK)
260     {
261         Error_Handler();
262     }
263     if (HAL_TIM_PWM_Init(&htim2) != HAL_OK)
264     {
265         Error_Handler();
266     }
267     sMasterConfig.MasterOutputTrigger = TIM_TRGO_ENABLE;
268     sMasterConfig.MasterSlaveMode = TIM_MASTERSLAVEMODE_ENABLE;
269     if (HAL_TIMEx_MasterConfigSynchronization(&htim2, &
        sMasterConfig) != HAL_OK)
270     {
271         Error_Handler();
272     }
273     sConfigOC.OCMode = TIM_OCMODE_PWM1;
274     sConfigOC.Pulse = 840-1;
275     sConfigOC.OCpolarity = TIM_OCPOLARITY_LOW;
276     sConfigOC.OCFastMode = TIM_OCFAST_DISABLE;
277     if (HAL_TIM_PWM_ConfigChannel(&htim2, &sConfigOC,
        TIM_CHANNEL_1) != HAL_OK)
278     {
279         Error_Handler();
280     }
281     /* USER CODE BEGIN TIM2_Init 2 */
282
283     /* USER CODE END TIM2_Init 2 */
284     HAL_TIM_MspPostInit(&htim2);
285
286 }
287
288 /**
289  * @brief TIM3 Initialization Function
290  * @param None
291  * @retval None
292  */
293 static void MX_TIM3_Init(void)
294 {
295

```

```

296      /* USER CODE BEGIN TIM3_Init 0 */
297
298      /* USER CODE END TIM3_Init 0 */
299
300      TIM_ClockConfigTypeDef sClockSourceConfig = {0};
301      TIM_MasterConfigTypeDef sMasterConfig = {0};
302      TIM_OC_InitTypeDef sConfigOC = {0};
303
304      /* USER CODE BEGIN TIM3_Init 1 */
305
306      /* USER CODE END TIM3_Init 1 */
307      htim3.Instance = TIM3;
308      htim3.Init.Prescaler = 0;
309      htim3.Init.CounterMode = TIM_COUNTERMODE_UP;
310      htim3.Init.Period = 42-1;
311      htim3.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
312      htim3.Init.AutoReloadPreload =
          TIM_AUTORELOAD_PRELOAD_DISABLE;
313      if (HAL_TIM_Base_Init(&htim3) != HAL_OK)
314      {
315          Error_Handler();
316      }
317      sClockSourceConfig.ClockSource = TIM_CLOCKSOURCE_INTERNAL;
318      if (HAL_TIM_ConfigClockSource(&htim3, &sClockSourceConfig)
          != HAL_OK)
319      {
320          Error_Handler();
321      }
322      if (HAL_TIM_PWM_Init(&htim3) != HAL_OK)
323      {
324          Error_Handler();
325      }
326      sMasterConfig.MasterOutputTrigger = TIM_TRGO_RESET;
327      sMasterConfig.MasterSlaveMode = TIM_MASTERSLAVEMODE_DISABLE
          ;
328      if (HAL_TIMEx_MasterConfigSynchronization(&htim3, &
          sMasterConfig) != HAL_OK)
329      {
330          Error_Handler();
331      }
332      sConfigOC.OCMode = TIM_OCMODE_PWM1;
333      sConfigOC.Pulse = 21-1;
334      sConfigOC.OCpolarity = TIM_OCPOLARITY_HIGH;
335      sConfigOC.OCFastMode = TIM_OCFAST_DISABLE;
336      if (HAL_TIM_PWM_ConfigChannel(&htim3, &sConfigOC,
          TIM_CHANNEL_1) != HAL_OK)
337      {
338          Error_Handler();
339      }
340      /* USER CODE BEGIN TIM3_Init 2 */
341
342      /* USER CODE END TIM3_Init 2 */
343      HAL_TIM_MspPostInit(&htim3);
344
345  }
346
347  /**

```

```

348 * @brief TIM4 Initialization Function
349 * @param None
350 * @retval None
351 */
352 static void MX_TIM4_Init(void)
353 {
354
355     /* USER CODE BEGIN TIM4_Init 0 */
356
357     /* USER CODE END TIM4_Init 0 */
358
359     TIM_ClockConfigTypeDef sClockSourceConfig = {0};
360     TIM_MasterConfigTypeDef sMasterConfig = {0};
361     TIM_OC_InitTypeDef sConfigOC = {0};
362
363     /* USER CODE BEGIN TIM4_Init 1 */
364
365     /* USER CODE END TIM4_Init 1 */
366     htim4.Instance = TIM4;
367     htim4.Init.Prescaler = 0;
368     htim4.Init.CounterMode = TIM_COUNTERMODE_UP;
369     htim4.Init.Period = 168-1;
370     htim4.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
371     htim4.Init.AutoReloadPreload =
372         TIM_AUTORELOAD_PRELOAD_DISABLE;
373     if (HAL_TIM_Base_Init(&htim4) != HAL_OK)
374     {
375         Error_Handler();
376     }
377     sClockSourceConfig.ClockSource = TIM_CLOCKSOURCE_INTERNAL;
378     if (HAL_TIM_ConfigClockSource(&htim4, &sClockSourceConfig)
379         != HAL_OK)
380     {
381         Error_Handler();
382     }
383     if (HAL_TIM_PWM_Init(&htim4) != HAL_OK)
384     {
385         Error_Handler();
386     }
387     sMasterConfig.MasterOutputTrigger = TIM_TRGO_RESET;
388     sMasterConfig.MasterSlaveMode = TIM_MASTERSLAVEMODE_DISABLE
389     ;
390     if (HAL_TIMEx_MasterConfigSynchronization(&htim4, &
391         sMasterConfig) != HAL_OK)
392     {
393         Error_Handler();
394     }
395     sConfigOC.OCMode = TIM_OCMODE_PWM1;
396     sConfigOC.Pulse = 42-1;
397     sConfigOC.OCpolarity = TIM_OCPOLARITY_HIGH;
398     sConfigOC.OCFastMode = TIM_OCFAST_DISABLE;
399     if (HAL_TIM_PWM_ConfigChannel(&htim4, &sConfigOC,
400         TIM_CHANNEL_4) != HAL_OK)
401     {
402         Error_Handler();
403     }
404     /* USER CODE BEGIN TIM4_Init 2 */

```

```

400
401     /* USER CODE END TIM4_Init 2 */
402
403 }
404
405 /**
406  * @brief TIM5 Initialization Function
407  * @param None
408  * @retval None
409  */
410 static void MX_TIM5_Init(void)
411 {
412
413     /* USER CODE BEGIN TIM5_Init 0 */
414
415     /* USER CODE END TIM5_Init 0 */
416
417     TIM_ClockConfigTypeDef sClockSourceConfig = {0};
418     TIM_SlaveConfigTypeDef sSlaveConfig = {0};
419     TIM_MasterConfigTypeDef sMasterConfig = {0};
420     TIM_OC_InitTypeDef sConfigOC = {0};
421
422     /* USER CODE BEGIN TIM5_Init 1 */
423
424     /* USER CODE END TIM5_Init 1 */
425     htim5.Instance = TIM5;
426     htim5.Init.Prescaler = 0;
427     htim5.Init.CounterMode = TIM_COUNTERMODE_UP;
428     htim5.Init.Period = 1680-1;
429     htim5.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
430     htim5.Init.AutoReloadPreload =
431         TIM_AUTORELOAD_PRELOAD_DISABLE;
432     if (HAL_TIM_Base_Init(&htim5) != HAL_OK)
433     {
434         Error_Handler();
435     }
436     sClockSourceConfig.ClockSource = TIM_CLOCKSOURCE_INTERNAL;
437     if (HAL_TIM_ConfigClockSource(&htim5, &sClockSourceConfig)
438         != HAL_OK)
439     {
440         Error_Handler();
441     }
442     if (HAL_TIM_PWM_Init(&htim5) != HAL_OK)
443     {
444         Error_Handler();
445     }
446     sSlaveConfig.SlaveMode = TIM_SLAVEMODE_TRIGGER;
447     sSlaveConfig.InputTrigger = TIM_TS_ITR0;
448     if (HAL_TIM_SlaveConfigSynchro(&htim5, &sSlaveConfig) !=
449         HAL_OK)
450     {
451         Error_Handler();
452     }
453     sMasterConfig.MasterOutputTrigger = TIM_TRGO_RESET;
454     sMasterConfig.MasterSlaveMode = TIM_MASTERSLAVEMODE_ENABLE;
455     if (HAL_TIMEx_MasterConfigSynchronization(&htim5, &
456         sMasterConfig) != HAL_OK)

```

```

453     {
454         Error_Handler();
455     }
456     sConfigOC.OCMode = TIM_OCMODE_PWM1;
457     sConfigOC.Pulse = 336-1;
458     sConfigOC.OCpolarity = TIM_OCPOLARITY_HIGH;
459     sConfigOC.OCFastMode = TIM_OCFAST_DISABLE;
460     if (HAL_TIM_PWM_ConfigChannel(&htim5, &sConfigOC,
461                                     TIM_CHANNEL_3) != HAL_OK)
462     {
463         Error_Handler();
464     }
465     /* USER CODE BEGIN TIM5_Init 2 */
466
467     /* USER CODE END TIM5_Init 2 */
468     HAL_TIM_MspPostInit(&htim5);
469 }
470
471 /**
472  * Enable DMA controller clock
473  */
474 static void MX_DMA_Init(void)
475 {
476
477     /* DMA controller clock enable */
478     __HAL_RCC_DMA2_CLK_ENABLE();
479
480     /* DMA interrupt init */
481     /* DMA2_Stream0_IRQn interrupt configuration */
482     HAL_NVIC_SetPriority(DMA2_Stream0_IRQn, 0, 0);
483     HAL_NVIC_EnableIRQ(DMA2_Stream0_IRQn);
484 }
485
486 /**
487  * @brief GPIO Initialization Function
488  * @param None
489  * @retval None
490  */
491
492 static void MX_GPIO_Init(void)
493 {
494     /* USER CODE BEGIN MX_GPIO_Init_1 */
495     /* USER CODE END MX_GPIO_Init_1 */
496
497     /* GPIO Ports Clock Enable */
498     __HAL_RCC_GPIOH_CLK_ENABLE();
499     __HAL_RCC_GPIOA_CLK_ENABLE();
500
501     /* USER CODE BEGIN MX_GPIO_Init_2 */
502     /* USER CODE END MX_GPIO_Init_2 */
503 }
504
505 /* USER CODE BEGIN 4 */
506 void HAL_ADC_ConvCpltCallback(ADC_HandleTypeDef* hadc){
507
508     CDC_Transmit_FS((uint8_t *)CCDPixelBuffer,CCDBuffer*2);

```

```

509     }
510
511
512     void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim){
513         if (htim->Instance == TIM2){
514
515             HAL_ADC_Start_DMA(&hadc1, (uint32_t *)CCDPixelBuffer,
516                               CCDBuffer);
517
518         }
519     }
520     /* USER CODE END 4 */
521
522     /**
523     * @brief This function is executed in case of error occurrence
524     *
525     * @retval None
526     */
527     void Error_Handler(void)
528     {
529         /* USER CODE BEGIN Error_Handler_Debug */
530         /* User can add his own implementation to report the HAL
531            error return state */
532         __disable_irq();
533         while (1)
534         {
535
536         }
537         /* USER CODE END Error_Handler_Debug */
538     }
539
540     #ifdef USE_FULL_ASSERT
541     /**
542     * @brief Reports the name of the source file and the source
543            line number
544            where the assert_param error has occurred.
545     * @param file: pointer to the source file name
546     * @param line: assert_param error line source number
547     * @retval None
548     */
549     void assert_failed(uint8_t *file, uint32_t line)
550     {
551         /* USER CODE BEGIN 6 */
552         /* User can add his own implementation to report the file
553            name and line number,
554            ex: printf("Wrong parameters value: file %s on line %d\r\n",
555                    file, line) */
556         /* USER CODE END 6 */
557     }
558     #endif /* USE_FULL_ASSERT */

```

Listing A.1: Código de programación microcontrolador

Apéndice B

Código fuente interpolación lineal en python

```
1
2 import numpy as np
3 import matplotlib.pyplot as plt
4 import pandas as pd
5
6 # Datos conocidos
7 pixels = np.array([4, 163, 848, 1057, 2666 , 3555, 3648])
8 wavelengths = np.array([455,467.5, 518.5, 532, 622.5 ,650,652])
9
10 # Array de pixeles del sensor CCD
11 total_pixels = 3648
12 pixel_array = np.arange(total_pixels)
13
14 # Interpolacion lineal
15 wavelength_array = np.interp(pixel_array, pixels, wavelengths)
16
17 # Guardar en un archivo Excel
18 df = pd.DataFrame({'Pixel': pixel_array, 'Longitud de Onda (nm)
19                    ': wavelength_array})
20 df.to_csv('Longitudes_de_Onda_CCD.csv', index=False)
21 print("Datos guardados en Longitudes_de_Onda_CCD")
22
23 # Graficar
24 plt.figure(figsize=(10, 6))
25 plt.plot(pixel_array, wavelength_array, label='Interpolacion
26           Lineal', color='blue')
27 plt.scatter(pixels, wavelengths, color='red', label='Datos
28           Conocidos', zorder=5)
29 plt.xlabel('Pixel')
30 plt.ylabel('Longitud de Onda (nm)')
31 plt.title('Interpolacion de Longitud de Onda para Sensor CCD')
32 plt.legend()
33 plt.grid(True)
34 plt.show()
```

```
33     # Opcional: Mostrar las longitudes de onda interpoladas  
34     print("Longitudes de onda interpoladas para cada pixel:")  
35     print(wavelength_array[:10]) # Muestra los primeros 10 valores  
        como ejemplo
```

Listing B.1: Código fuente interpolación lineal en python

Apéndice C

Código fuente del receptor en Python

```
1  import serial
2  import numpy as np
3  import matplotlib.pyplot as plt
4  import pandas as pd
5  import pywt
6  from scipy.signal import savgol_filter
7
8  #variable para seleccionar tipo de filtro 1:Wavelet, 2:Savitzky
   -Goaly
9  Filtro = 1
10
11  #Tipo de wavelet a usar
12  WAVELET = 'db1'
13
14  #Archivo para la calibracion dependiendo del filtro se usa uno
   u otro
15  file = "Calibracion_Oscuridad_Wavelet.csv"
16  #file = "Calibracion_Oscuridad_SG.csv"
17
18  # Configura el puerto USB CDC
19  ser = serial.Serial('COM5', timeout=1) # Cambia 'COM5' al
   puerto correcto
20
21  plt.ion() # Modo interactivo para la grafica
22
23  # Inicializa el arreglo con ceros para mantener siempre 3770
   datos
24  NUM_SAMPLES = 3770
25  data_array = np.zeros(NUM_SAMPLES, dtype=np.float64)
26  data_array_Temporal = np.zeros(NUM_SAMPLES, dtype=np.float64)
27  # Variables de control de captura de espectros
28  NUM_ESPEC = 100
29  INTEG_TIME = 0.001
30
31  #Leer archivo de calibracion
```

```

32 df = pd.read_excel(file)
33 column1 = "Valor"
34 column2 = "LongitudDeOnda"
35
36 #Vector para eliminacion del ruido en la oscuridad
37 Calibracion_Oscuridad = df[column1].values
38
39 #Vector de longitudes de onda
40 Longitudes_Onda = df[column2].values
41
42 def read_and_plot():
43     global data_array_Temporal #Se usa global para modificar la
44         lista fuera de la funcion
45     global data_array # Se usa global para modificar la lista
46         fuera de la funcion
47     CONT_ESPEC = 0
48     while True:
49         if ser.in_waiting >= NUM_SAMPLES * 2: # Asegura que haya
50             suficientes datos antes de leer
51         # Lee exactamente 3770 muestras (cada muestra son 2 bytes)
52         raw_data = ser.read(NUM_SAMPLES * 2)
53
54         # Convierte los datos crudos a enteros (asumiendo 16 bits por
55             dato)
56         int_data = 4095 - np.frombuffer(raw_data, dtype=np.uint16)
57
58         # Suma los nuevos datos con los datos anteriores
59         if Filtro == 1:
60             data_array_Temporal += denoise(time_serie=int_data)
61             data_array_Temporal += int_data
62             data_array = data_array_Temporal[32:3680]
63
64         if Filtro == 2:
65             data_array_Temporal += savgol_filter(int_data, window_length
66                 =91, polyorder=3)
67             data_array = data_array_Temporal[32:3680]
68
69         # Ajusta el limite del eje Y al maximo valor actual de la
70             grafica
71         y_max = np.max(data_array) * 1.1 # Agrega un 10% de margen
72             visual
73         y_max = max(y_max, 100) # Evita que el limite sea demasiado
74             pequeno
75
76         # Limpia la grafica y vuelve a graficar
77         plt.clf()
78         plt.plot(Longitudes_Onda, data_array)
79         plt.ylim(0, y_max) # Limite dinamico del eje Y
80         plt.xlim(Longitudes_Onda[0], Longitudes_Onda[-1])
81         plt.xlabel('Longitud de onda (nm)')
82         plt.ylabel('Valor Acumulado')
83         plt.title('Datos Acumulados por USB')
84         plt.pause(INTEG_TIME) # Breve pausa para actualizar la grafica
85
86     CONT_ESPEC += 1
87     if CONT_ESPEC == NUM_ESPEC:

```

```

81 data_array = data_array / NUM_ESPEC
82 # Guardar datos en un archivo csv
83 data_to_save = data_array - Calibracion_Oscuridad
84 try:
85     df = pd.DataFrame({
86         'LongitudDeOnda': Longitudes_Onda,
87         'Valor': data_to_save
88     })
89     df.to_csv('VN5.csv', index=False)
90     print("Datos guardados")
91 except Exception as e:
92     print("Error al guardar los datos:", e)
93
94     break
95
96
97 def denoise(time_serie, absolute=True):
98     coeffs = pywt.wavedec(time_serie, WAVELET, level=8, mode="
99         symmetric")
100     sigma = np.median(np.abs(coeffs[8]))/0.6745
101     for i in range(6,9):
102         D = coeffs[i]
103         n = len(D)
104         tVisu = np.sqrt(2*np.log(n))*sigma
105         coeffs[i] = pywt.threshold(D, tVisu, mode='hard')
106         time_serie_rec = pywt.waverec(coeffs, WAVELET, mode="symmetric"
107             )
108         if absolute:
109             time_serie_rec = np.abs(time_serie_rec)
110         return time_serie_rec
111
112 try:
113     read_and_plot()
114 except KeyboardInterrupt:
115     print("Proceso interrumpido por el usuario.")
116 finally:
117     ser.close()
118     plt.ioff()
119     plt.show()

```

Listing C.1: Código fuente del receptor en Python

Apéndice D

Código fuente PCA en python

```
1  import pandas as pd
2  from sklearn.decomposition import PCA
3  from sklearn.preprocessing import StandardScaler
4  import matplotlib.pyplot as plt
5  import seaborn as sns
6
7  # 1. Cargar archivo
8  df = pd.read_excel("espectros_transpuestos.xlsx", header=None)
9
10 # 2. Extraer ID y datos numericos
11 ids = df.iloc[:, 0]
12 X = df.iloc[:, 1:].values
13
14 # 3. Funciones para extraer clase y metodo desde ID
15 def obtener_muestra(id_str):
16     if id_str.startswith("C"):
17         return "Clorofila"
18     elif id_str.startswith("S"):
19         return "Cafe"
20     elif id_str.startswith("V"):
21         return "Vidrios"
22     else:
23         return "Desconocido"
24
25 def obtener_metodo(id_str):
26     if id_str.endswith("N" + id_str[-1]): # e.g., CN1, SN2, VN3
27         return "Sin filtrar"
28     elif "W" in id_str:
29         return "Wavelets"
30     elif "SG" in id_str in id_str: # para SSG, CSG, VSH
31         return "Savitzky-Golay"
32     else:
33         return "Desconocido"
34
35 # 4. Obtener etiquetas
```

```

36 muestras = ids.apply(obtener_muestra)
37 metodos = ids.apply(obtener_metodo)
38
39 # 5. Normalizar datos
40 X_norm = StandardScaler().fit_transform(X)
41
42 # 6. PCA
43 pca = PCA(n_components=2)
44 X_pca = pca.fit_transform(X_norm)
45
46 # 7. DataFrame de resultados
47 df_pca = pd.DataFrame({
48     'ID': ids,
49     'Muestra': muestras,
50     'Metodo': metodos,
51     'PC1': X_pca[:, 0],
52     'PC2': X_pca[:, 1]
53 })
54
55 # 8. Grafica
56 plt.figure(figsize=(10, 7))
57 sns.scatterplot(data=df_pca, x='PC1', y='PC2', hue='Muestra',
58                 style='Metodo', s=100)
59 plt.title("PCA de espectros")
60 plt.xlabel(f"PC1 ({pca.explained_variance_ratio_[0]*100:.1f}%
61             var. explicada)")
62 plt.ylabel(f"PC2 ({pca.explained_variance_ratio_[1]*100:.1f}%
63             var. explicada)")
64 plt.grid(True)
65 plt.tight_layout()
66 plt.show()
67
68 # 9. Guardar resultados
69 df_pca.to_excel("resultados_PCA_muestras.xlsx", index=False)

```

Listing D.1: Código fuente PCA en python

Apéndice E

Código fuente SNR en python

```
1  import numpy as np
2  import pandas as pd
3
4  # Cargar los datos desde un archivo CSV
5  file_path = "LectorEspectrometro3/CSG1.CSV" # Cambia esta ruta
6  # a tu archivo CSV
7  df = pd.read_excel(file_path)
8
9  # Extraer los datos de la columna 'valor3'
10 data = np.array(df['Valor'])
11
12 # 1. Calcula la potencia de la señal (promedio cuadrático)
13 signal_power = np.mean(data**2)
14
15 # 2. Calcula el ruido (desviación estándar)
16 noise_power = np.std(data)
17
18 # 3. Calcula el SNR (relación señal ruido)
19 snr = signal_power / noise_power
20
21 # 4. SNR en dB
22 snr_db = 10 * np.log10(snr)
23
24 print(f"SNR = {snr_db:.4f} dB")
```

Listing E.1: Código fuente SNR en python