DEPARTAMENTO DE
INGENIERÍA ELÉCTRICA
UNIVERSIDAD DE CONCEPCIÓN

AUTOMATIZACIÓN DEL CONTROL DE POLARIZACIÓN EN FIBRA ÓPTICA MEDIANTE ARDUINO

POR

Cristóbal Gabriel Vegas Moya

Memoria de Título presentada a la Facultad de Ingeniería de la Universidad de Concepción para optar al título profesional de Ingeniero Civil en Telecomunicaciones.

Profesor Guía
Gabriel Saavedra Mondaca, Ph.D.

Concepción,
2 de julio de 2025

© 2025 Cristóbal Gabriel Vegas Moya

© 2025 Cristóbal Gabriel Vegas Moya
Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento.

A todos los estudiantes de Telecom...

Resumen

La inestabilidad del estado de polarización en sistemas de fibra óptica degrada la calidad de la señal, y su corrección mediante sistemas comerciales resulta costosa. Este trabajo aborda dicho problema mediante el desarrollo de una solución de control automatizado, asequible y de código abierto. El objetivo fue diseñar, implementar y validar un sistema de bajo costo que controla el estado de polarización en tiempo real, utilizando un controlador mecánico gobernado por Arduino para maximizar la potencia en una de las salidas del divisor de haz polarizador y minimizarla en la ortogonal.

La metodología se basó en la construcción de un controlador de polarización mecánico, con piezas impresas en 3D y accionado por tres servomotores. El lazo de control se implementó conectando la salida de los dos fotodetectores en paralelo: una salida se dirigió a un osciloscopio para el monitoreo y la otra a un Arduino para el procesamiento de los datos y la automatización. Para la automatización, se desarrolló un algoritmo de optimización en Python que ejecuta una búsqueda de dos etapas (barrido grueso y ajuste fino) y se guía por una métrica de rendimiento normalizada para asegurar el máximo contraste entre las salidas. El rendimiento del sistema se evaluó experimentalmente bajo condiciones de potencia óptica de entrada baja (1 dBm), óptima (3 dBm) y alta (7.5 dBm).

Los resultados demostraron una excelente eficacia y repetibilidad del sistema en los rangos de operación no saturados, donde la métrica de rendimiento alcanzó valores consistentemente cercanos al ideal de 1. No obstante, se caracterizó una limitación fundamental del hardware a potencias elevadas, donde la saturación del conversor analógico-digital del Arduino degradó el rendimiento. Se concluye

que el sistema desarrollado es una alternativa funcional, precisa y viable frente a las soluciones comerciales, validando el uso de hardware de bajo costo para la instrumentación óptica en entornos de investigación y desarrollo.

Abstract

The instability of the state of polarization in optical fiber systems degrades signal quality, and its correction through commercial systems is costly. This work addresses said problem by developing an automated, affordable, and open-source control solution. The objective was to design, implement, and validate a low-cost system that controls the state of polarization in real time, using a mechanical controller governed by Arduino to maximize power at one of the polarization beam splitter outputs and minimize it at the orthogonal one.

The methodology was based on the construction of a mechanical polarization controller, with 3D-printed parts and actuated by three servomotors. The control loop was implemented by connecting the output of the two photodetectors in parallel: one output was directed to an oscilloscope for monitoring and the other to an Arduino for data processing and automation. For automation, a Python optimization algorithm was developed that executes a two-stage search (coarse scan and fine adjustment) and is guided by a normalized performance metric to ensure maximum contrast between the outputs. The system's performance was experimentally evaluated under conditions of low (1 dBm), optimal (3 dBm), and high (7.5 dBm) optical input power.

The results demonstrated excellent efficacy and repeatability of the system in non-saturated operating ranges, where the performance metric consistently reached values close to the ideal of 1. Nevertheless, a fundamental hardware limitation was characterized at high power levels, where the saturation of the Arduino's analog-to-digital converter degraded performance. It is concluded that the developed system is a functional, precise, and viable alternative to commercial solutions,

validating the use of low-cost hardware for optical instrumentation in research and development environments.

Agradecimientos

Quiero agradecer a todas las personas que me apoyaron, motivaron y alentaron durante todo este proceso.

En primer lugar, a mis padres, Ivon y Osvaldo, por apoyarme siempre en mis decisiones, por guiarme con sus consejos y por formarme como la persona que soy hoy. Jamás podré devolverles todo el cariño y el esfuerzo que me han entregado, y en especial, la infinita paciencia para soportarme. También, un agradecimiento especial a mis tíos y primos por su apoyo constante.

A mis amigos Guto, Pipe, Titin, Lucho, García, Pablo, Bentrix, Jeria, Chamorro, Hermo y Nico. Cada vez que nos juntábamos o hablábamos por Discord, las risas lograban que cualquier problema se olvidara. Gracias por hacer el día a día mucho más ligero y divertido.

A mis amigos de la carrera o la universidad, por estar siempre ahí cuando los necesité y por nunca dejar que me sintiera apartado: Efra, Alonso, Tello, Nivek, Galo, Kova, Checho, Dani y Lliva. Un agradecimiento también a mis amigos del laboratorio, Cristóbal, Leo y Harry; su ayuda fue muy importante cada vez que tenía dudas sobre la teoría, el uso de algún dispositivo o cómo mejorar mi algoritmo. Gracias por todo el apoyo, por las risas que compartimos y por los fifas.

A mi profesor guía, Gabriel, por su enorme paciencia y disposición. Ante cualquier duda, consulta o problema, siempre respondió con rapidez, y sus consejos sobre el tema me facilitó enormemente este trabajo. También quiero agradecer a los demás profesores que me ayudaron durante estos años de universidad y me

enseñaron acerca de este mundo, como el profesor Godoy, Sobarzo.

Gracias a todas las demás personas que, de una u otra forma, hicieron mis días más divertidos y me aconsejaron.

Índice General

Resumen I

Abstract III

Agradecimientos V

Índice de Figuras X

Índice de Tablas XII

1. Introducción **1**

1.1. Trabajos previos 3

1.1.1. Discusión 5

1.2. Definición del problema 6

1.3. Objetivos 7

1.3.1. Objetivo general 7

1.3.2. Objetivos específicos 7

1.4. Metodología 8

1.5. Alcances y limitaciones 12

1.5.1. Alcances 12

1.5.2. Limitaciones 13

2. Marco Teórico **14**

2.1. Introducción 14

2.2. Conceptos Fundamentales 15

2.2.1. Luz y sus propiedades. 15

2.2.2. Tipos de Polarización. 15
2.2.3. Fibra Óptica 17
2.2.4. Birrefringencia 18
2.3. Componentes Principales 20
2.3.1. Polarizing Beam Splitter (PBS) 20
2.3.2. Fotodetector 20
2.3.3. Servomotor 21
2.3.4. Arduino 21
2.3.5. LáserPRO800 22
2.3.6. Osciloscopio 23
2.3.7. Generador de Funciones arbitrarias 24
2.4. Teorías y Principios Relevantes 24
2.4.1. Cálculo de Jones 24
2.4.1.1. Vector de Jones 25
2.4.1.2. Matriz de Jones 26
2.4.2. Principio de Control de Polarización Mecánico 26
2.4.3. Controlador de polarización 27
2.4.4. Conversión de Potencia Óptica a Voltaje 28
2.4.5. Lectura del Arduino 31
2.5. Enfoque Metodológico 32

3. Desarrollo **33**
3.1. Introducción 33
3.2. Metodología de Implementación 33

4. Resultados **50**
4.1. Configuración Experimental 50
4.2. Análisis del comportamiento en tiempo real 51
4.2.1. Caso de Operación Óptima (3 dBm) 52
4.2.2. Casos de Baja Potencia y Saturación 53
4.3. Evaluación del Rendimiento del algoritmo 54
4.3.1. Rendimiento en Rango Óptimo y Bajo (1 dBm y 3 dBm) . . . 54
4.3.2. Rendimiento en Condiciones de Saturación (7.5 dBm) 56

4.4. Validar experimentalmente el sistema diseñado 57
4.5. Tiempo promedio de ejecución. 58

5. Conclusiones 60
5.1. Sumario 60
5.2. Conclusión 61
5.3. Trabajo a Futuro 62

Bibliografía 67

Appendices 68

A. ANEXO: Figuras relevantes pero no tanto para el cuerpo de la tesis 69

B. ANEXO: Código 70

Índice de Figuras

2.1. Onda electromagnetica con su campo electrico y campo magnetico. 15
2.2. Los 3 distintos tipos de polarización. 16
2.3. Fibra Optica SMF y MMF. 17
2.4. Esfera Poincaré 19
2.5. Polarizing Beam Splitter(PBS) 20
2.6. Fotodetector 21
2.7. Arduino UNO 22
2.8. LaserPRO800 23
2.9. Osciloscopio 23
2.10. Generador de funciones 24
2.11. Responsividad del fotodetector 29

3.1. Montaje del Setup. Fuente: Autoridad Propia 35
3.2. Diagrama del Setup. Fuente: Autoridad Propia 35
3.3. Interfaz Inicial. Fuente: Autoridad Propia. 36
3.4. Diagrama de Flujo Básico. Fuente: Autoridad Propia. 39
3.5. Diagrama de flujo automatización. Fuente: Autoridad Propia. 46
3.6. Interfaz final. Fuente: Elaboración propia. 48
3.7. Gráfico de los voltajes en tiempo real. Fuente: Elaboración propia. . 49

4.1. Gráfico en tiempo real con potencia de 3 dBm maximizando la polarización horizontal. Fuente: Elaboración propia. 52
4.2. Gráficos en tiempo real con potencia de 1 dBm. Fuente: Elaboración propia. 53

4.3. Gráficos en tiempo real con potencia de 7.5 dBm, mostrando saturación. Fuente: Elaboración propia. 54
4.4. Resultados de la automatización con una potencia de 3 dBm. 55
4.5. Resultados de la automatización con una potencia de 1 dBm. 55
4.6. Resultados de la automatización con una potencia de 7.5 dBm. . . . 56

Índice de Tablas

4.1. Promedios y desviacion estandar de las metricas. 57

Siglas

CMOS Semiconductor Complementario de Óxido Metálico

IoT Internet de las Cosas

PBS Divisor de Haz de Polarización

PO Perturbar y Observar

SiP Fotónica de Silicio

SOP Estado de Polarización

Capítulo 1

Introducción

En las últimas décadas, la fibra óptica se ha consolidado como principal de las telecomunicaciones globales, habilitando la transmisión de datos a velocidades muy altas. Su rol es fundamental para el despliegue de tecnologías emergentes como el 5G y el Internet de las Cosas (IoT), que demandan una infraestructura de red con alta capacidad y baja latencia [1]. Sin embargo, el rendimiento óptimo de estas redes está condicionado por un fenómeno físico fundamental: el estado de polarización (SOP) de la luz que se propaga por la fibra. Durante la instalación y operación, tensiones mecánicas, curvaturas y cambios de temperatura inducen birrefringencia en la fibra, alterando de forma inesperada el SOP [2]. Esta variable de la polarización puede perjudicar severamente la calidad de la señal, especialmente en sistemas que utilizan componentes sensibles a ella, como los divisores de haz de polarización (PBS), llevando a pérdidas de potencia y errores en la transmisión.

Tradicionalmente, la gestión de este fenómeno se ha abordado mediante ajustes manuales, un enfoque limitado por su lentitud e imprecisión. Aunque existen sistemas automatizados en el mercado, su alto costo dificulta su uso en contextos de investigación y desarrollo. Esta situación crea una brecha tecnológica para el desarrollo de soluciones de bajo costo, personalizables y de código abierto que permitan un control robusto y fiable de la polarización [3]. Esta memoria de título propone una solución asequible y eficiente que no solo resuelve un problema

técnico prevalente, sino que también aporta al conocimiento tecnológico en el área de la instrumentación óptica.

Esta memoria de título aborda este problema presentando el diseño, construcción y validación de un sistema automatizado de control de polarización en tiempo real. El objetivo general es estabilizar el SOP de una señal láser de 1550 nm para maximizar la potencia óptica en una de las salidas de un divisor de polarización[4] (ej. polarización horizontal) y, simultáneamente, atenuar al máximo la potencia en la salida ortogonal (vertical). Este objetivo dual es crucial para aplicaciones que requieren separar y optimizar estados de polarización específicos.

Para alcanzar dicho objetivo, se desarrolló un controlador de polarización mecánico de bajo costo, fabricado mediante impresión 3D e inspirado en un diseño existente [5]. El control activo del sistema se logra a través de tres servomotores, cada uno asociado a una paleta que ejerce presión sobre la fibra, operados por un Arduino [6]. Se implementó un lazo de control de retroalimentación, donde dos fotodetectores [7] miden la potencia óptica de las salidas vertical y horizontal del sistema. Esta información de potencia, leída como un voltaje por el Arduino, es procesada por un algoritmo de optimización que ajusta de manera autónoma y continua la posición de los servomotores, completando así el ciclo de automatización para mantener el estado de polarización deseado.

La presente Memoria de Título se estructura en cinco capítulos. El Capítulo 1 introduce el problema de la inestabilidad de la polarización en fibras ópticas, estableciendo los objetivos, alcances y una revisión del estado del arte que justifica la necesidad de este trabajo. A continuación, el Capítulo 2 profundiza en el marco teórico, presentando los fundamentos físicos y matemáticos de la polarización, la birrefringencia y los componentes clave del sistema. El Capítulo 3 se dedica al desarrollo experimental, detallando el montaje del sistema, el diseño de la interfaz gráfica, la implementación de la comunicación serial y la lógica del algoritmo de automatización. Posteriormente, el Capítulo 4 presenta y analiza los resultados cuantitativos obtenidos en las pruebas de laboratorio, evaluando el rendimiento del controlador en términos de estabilidad, velocidad y eficacia. Finalmente, el Capítulo 5 cierra la investigación, presentando primero un sumario

de los hallazgos más importantes, para luego exponer las conclusiones definitivas del trabajo y una discusión sobre las posibles líneas de trabajo y mejoras futuras.

1.1. Trabajos previos

La gestión del SOP en sistemas de fibra óptica es un desafío fundamental para asegurar la integridad de la señal en un amplio espectro de aplicaciones, desde las telecomunicaciones de alta velocidad hasta los sensores de precisión. Las perturbaciones mecánicas y térmicas sobre la fibra alteran el SOP de forma impredecible, lo que requiere sistemas de control activos para compensar estas variaciones en tiempo real [8]. La tecnología y los métodos para lograr este control han evolucionado significativamente, abarcando desde principios mecánicos hasta soluciones optoelectrónicas integradas, cada una con sus propios algoritmos de control [9].

Las primeras soluciones para automatizar este control se basaron en principios mecánicos, aplicando un estrés controlado sobre la propia fibra para inducir birrefringencia. Este enfoque ha sido validado a nivel industrial, como lo demuestra la patente que describe dispositivos basados en este método [10]. Una de las implementaciones más comunes de este principio son los compresores de fibra o “fiber squeezers”, donde un actuador aplica una presión precisa y controlada sobre la fibra. Se han utilizado diferentes tipos de actuadores para lograr esta compresión; por ejemplo, en [11] se describe un controlador comercial que utiliza compresores de fibra piezoeléctricos. De forma análoga, el trabajo [12] presenta un ”fiber squeezer”donde la fuerza de compresión es generada por un actuador electromagnético, en este caso, un solenoide.

Sobre esta base tecnológica de control mecánico, se han desarrollado sistemas modernos, como el descrito en [13], que son gobernados por algoritmos de optimización avanzados. Además del método de presión, se han explorado otras formas de manipulación mecánica, como la torsión controlada de la fibra [14], lo que demuestra la diversidad de enfoques que existen dentro del control mecánico

de polarización.

Paralelamente a la evolución de los dispositivos, las estrategias de control algorítmico también han avanzado significativamente. Estas abarcan desde algoritmos de búsqueda iterativa fundamentales, como los métodos de ascenso de colina (”hill-climbing”) [15], hasta la aplicación de técnicas de Inteligencia Artificial más sofisticadas. En este último ámbito, se ha explorado el uso de redes neuronales para el control de actuadores mecánicos como los ”fiber squeezers”[16] y arquitecturas de Aprendizaje Profundo con Control Predictivo (DL-MPC) para la auto-sintonización de sistemas ópticos complejos [17]. En contraste con estos enfoques de búsqueda, también se han desarrollado soluciones analíticas que calculan directamente la transformación de polarización, buscando una convergencia más rápida que los métodos iterativos tradicionales [15].

En la búsqueda de velocidades de compensación superiores, indispensables para las redes de telecomunicaciones de alta capacidad, una línea de investigación fundamental se ha centrado en el desarrollo de soluciones no mecánicas. Se exploraron tecnologías electro-ópticas que utilizan materiales como el Niobato de Litio ($LiNbO_3$) para fabricar controladores monolíticos en chip. Estos dispositivos, descritos en trabajos como [18] y [19], modifican la polarización mediante la aplicación de voltajes a guías de onda, utilizando componentes como convertidores de modo y desplazadores de fase. La principal ventaja de este enfoque es su alta velocidad, logrando tiempos de respuesta en el orden de los microsegundos, un salto significativo en rendimiento en comparación con los sistemas mecánicos.

La tendencia más reciente en este campo apunta hacia la integración a gran escala y la disminución del tamaño mediante la fotónica de silicio (SiP). Esta tecnología, compatible con los procesos de fabricación CMOS estándar, permite crear controladores dinámicos y complejos en un espacio muy reducido, como se detalla en [20]. Un ejemplo concreto se presenta en [21], donde se reporta un controlador con un ancho de banda de modulación de aproximadamente 30 kHz, lo que se traduce en tiempos de respuesta del orden de las decenas de

microsegundos. Este nivel de integración es considerado imperativo para el futuro de los sistemas de comunicación óptica de alta velocidad totalmente integrados.

El control del SOP no solo es fundamental para la transmisión de señales, sino que también es una herramienta crucial en otros ámbitos de la fotónica, como en el diseño y la operación de láseres de fibra. En este contexto, se utiliza para manipular las propiedades de la luz directamente en la fuente. Un ejemplo de esto es el control de los modos de emisión: mediante un ajuste preciso de la polarización, es posible generar modos de fibra específicos, incluyendo modos de momento angular orbital (OAM), como se demuestra en el trabajo [22]. Por otro lado, la polarización es clave para iniciar y optimizar el fenómeno de ”bloqueo de modos”(mode-locking) en láseres de fibra. Para ello, se han desarrollado sistemas automatizados que utilizan retardadores de cristal líquido (LCVRs) y algoritmos de optimización para compensar perturbaciones y mantener el láser en su rendimiento óptimo, como se detalla en [23].

La relevancia de abordar este tipo de problemas de control mediante software y hardware accesible también se refleja en el ámbito académico. Un ejemplo análogo al presente trabajo es la tesis [24], donde se desarrolla un sistema para la compensación automática de la deriva de polarización. Dicho trabajo, aunque utiliza un controlador comercial (General Photonics, PCD-M02), implementa un programa en Python para comunicarse con los dispositivos y optimizar la señal. Es destacable que el principio físico de su controlador es el mismo que se explora en esta memoria: el uso de ”fiber squeezers”piezoeléctricos que aplican tensión a la fibra para inducir birrefringencia y así controlar el SOP de manera precisa.

1.1.1. Discusión

Del análisis de los trabajos previos, se desprenden conclusiones clave sobre las estrategias dominantes para el control de polarización, las cuales revelan un compromiso fundamental entre rendimiento y complejidad. A nivel de hardware, se observan dos tendencias principales: los sistemas mecánicos, que manipulan la fibra mediante presión o torsión [10] [13] [14], ofrecen la ventaja de una mayor

simplicidad y accesibilidad, pero a expensas de una velocidad de respuesta más lenta. En contraste, las soluciones electro-ópticas e integradas [18] [21] ofrecen un rendimiento muy superior en velocidad e integración monolítica, aunque con una mayor complejidad de fabricación. De manera análoga, en el dominio de los algoritmos de control, los enfoques van desde búsquedas iterativas simples como Perturbar y Observar (PO) [11], que son directas de implementar pero pueden converger en óptimos locales, hasta técnicas de Inteligencia Artificial [16] [17], que prometen un control más robusto a costa de una mayor carga de procesamiento.

A partir de este análisis, se identifica una oportunidad de desarrollo en la intersección de estos enfoques. Existe un espacio para sistemas que aprovechen la simplicidad y accesibilidad del control mecánico, pero que usen un algoritmo de optimización más robusto que una simple búsqueda local, sin llegar a la complejidad de la Inteligencia Artificial. La literatura muestra una cierta polarización entre sistemas mecánicos con control simple y sistemas electro-ópticos con control complejo. Esta memoria de título aborda precisamente esa área intermedia, presentando el diseño y el método de un sistema mecánico de bajo costo, pero dotado de un algoritmo de automatización (basado en barridos exhaustivos y ajustes finos) diseñado para escapar de los óptimos locales y ofrecer un rendimiento eficiente.

1.2. Definición del problema

Como se ha mencionado anteriormente, la transmisión de señales en fibra óptica es altamente susceptible a la polarización. Este fenómeno, originado por la birrefringencia inducida por factores externos como la temperatura, la vibración y el estrés mecánico sobre la fibra, provoca una alteración aleatoria e impredecible en el SOP de la señal. Consecuentemente, esto degrada la calidad en componentes que dependen de la polarización, como los circuitos fotónicos integrados, llevando a significativas pérdidas de potencia y a una degradación general del rendimiento del enlace.

Adicionalmente, si bien se han propuesto diversos métodos, estos muestran un compromiso fundamental entre rendimiento y complejidad. Por un lado, los sistemas mecánicos como los fiber squeezers o los controladores de onda rotatorios son de implementación directa, pero su velocidad de respuesta es limitada. Por otro lado, los algoritmos de control varían desde enfoques iterativos simples como Perturbar y Observar (PO), que son fáciles de implementar pero pueden converger lentamente o en óptimos locales, hasta técnicas avanzadas basadas en Inteligencia Artificial, que ofrecen mayor robustez a costa de una mayor carga de procesamiento de datos. Así, esta Memoria de Título aborda el problema desde la perspectiva del control algorítmico en tiempo real, proponiendo y evaluando una nueva alternativa de compensación. El objetivo es desarrollar un sistema que sea eficiente y simple de implementar, pero que a la vez sea capaz de mantener un estado de polarización estable y un tiempo de respuesta eficiente, con el fin de mejorar el rendimiento, la estabilidad y la eficiencia de los enlaces de comunicación óptica.

1.3. Objetivos

1.3.1. Objetivo general

Desarrollar e implementar un sistema automatizado para el control del estado de polarización en una fibra óptica monomodo (SMF), mediante el uso de un controlador de polarización con servomotores controlados por Arduino. El propósito es mejorar la eficiencia y precisión en el ajuste de la polarización, eliminando la necesidad de ajustes manuales que resultan lentos e inexactos.

1.3.2. Objetivos específicos

En base al objetivo general dado, los objetivos específicos son:

1. Desarrollar un código en Python vinculado a Arduino que permita modificar la polarización de manera manual mediante el uso del control de la polarización, con una interfaz gráfica programada en Python.
2. Diseñar e implementar un algoritmo de control automático para optimizar el estado de polarización, con el propósito de maximizar la potencia óptica en una salida de polarización deseada y, simultáneamente, minimizarla en la salida ortogonal.
3. Desarrollar un gráfico en tiempo real para poder observar como varía el voltaje al momento de automatizar.
4. Validar experimentalmente el sistema diseñado.

1.4. Metodología

Ahora se explicará el paso a paso que se realizó para los objetivos específicos:

1. **Desarrollar un código en Python vinculado a Arduino que permita modificar la polarización de manera manual mediante el uso del control de la polarización, con una interfaz gráfica programada en Python.**

   Inicialmente, se llevó a cabo una selección y revisión de los componentes requeridos para el montaje experimental. Esta etapa fundamental incluyó el análisis de las hojas de datos de cada dispositivo para determinar sus especificaciones técnicas. El sistema se diseñó en torno a una fuente láser operando a una longitud de onda de 1550 nm, una fibra óptica monomodo (SMF) como medio de transmisión, y un controlador de polarización fabricado mediante impresión 3D, cuyo diseño se basó en el modelo MPC320 de Thorlabs, el cual estuvo controlado por un microcontrolador Arduino UNO. La separación y detección de la señal óptica se realizó mediante un divisor de haz polarizante (PBS) y dos fotodetectores, con un osciloscopio para la monitorización visual de las señales. El propósito de esta revisión fue asegurar la compatibilidad entre todos los componentes

y comprender sus límites operativos antes de proceder con su integración física. El montaje detallado y el modo de uso de estos dispositivos se describen en secciones posteriores

Posteriormente, se procedió al diseño conceptual y desarrollo de una interfaz gráfica de usuario (GUI) utilizando el lenguaje de programación Python y su librería Tkinter. El objetivo del diseño fue crear una herramienta intuitiva que permitiera al usuario gestionar la comunicación con el dispositivo arduino, incluyendo la selección del puerto y el establecimiento de la conexión, y, a su vez, poder usar el control manual de los ángulos de los servomotores del controlador de polarización

La etapa final para este objetivo consistió en la implementación de la comunicación serial entre el código en Python y el código para el microcontrolador Arduino. Se estableció un protocolo de comunicación que permitía el envío de comandos desde la interfaz gráfica de usuario hacia el Arduino, posibilitando así la actuación precisa de los servomotores. De manera complementaria, se desarrolló la funcionalidad para solicitar y recibir datos de voltaje desde el Arduino en tiempo real. Estos datos, provenientes de los fotodetectores, fueron comparados con las mediciones visualizadas en un osciloscopio para validar la correcta captura de la información.

2. **Diseñar e implementar un algoritmo de control automático para optimizar el estado de polarización, con el propósito de maximizar la potencia óptica en una salida de polarización deseada y, simultáneamente, minimizarla en la salida ortogonal.**

   Para lograr la automatización del control de polarización, se desarrolló e implementó un algoritmo de optimización en el código de Python. Para llegar a la solución de automatización final, se consideraron y pusieron a prueba inicialmente diversas estrategias. Entre ellos, se consideró una maximización directa del voltaje objetivo con lógica de reversión (rollback) [20], donde el sistema intentaba incrementar la potencia en un detector y volvía a un estado anterior si se superaban ciertos límites o no había mejora. Otro concepto explorado fue una variación del método de ”Modificar

y Observar”[11] simple, donde los ajustes se realizaban en orden para cada paleta observando el impacto directo en el voltaje. Si bien estos métodos ofrecieron aprendizajes iniciales, presentaron limitaciones en cuanto a la velocidad de convergencia y la capacidad de evitar óptimos locales.

Dadas estas observaciones, se procedió a implementar la estrategia de ajuste final, que constó de múltiples etapas para cada servomotor. Inicialmente, se realizaba un barrido angular grueso sobre un rango amplio de movimiento del servo para identificar las regiones de operación más prometedoras. Una vez identificada una posición angular que ofrecía una mejora en el rendimiento del sistema, se ejecutaba un procedimiento de ajuste fino en un rango más acotado alrededor de dicha posición, utilizando pasos angulares más pequeños para refinar el resultado. Este proceso de barrido grueso seguido de ajuste fino se repetía para cada uno de los tres servomotores. Para asegurar la búsqueda de un estado óptimo global, todo el ciclo de optimización de los tres servomotores se podía ejecutar en múltiples pasadas completas.

El objetivo principal de este algoritmo fue ajustar de manera automática las paletas del controlador de polarización para alcanzar un estado óptimo, definido por la maximización de la potencia óptica en una de las salidas del divisor de haz polarizante (PBS) y, simultáneamente, la atenuación de la potencia en la salida de polarización ortogonal.

Para cuantificar este objetivo dual de manera efectiva, el método de automatización se basó en una métrica de rendimiento normalizada, calculada a partir de los voltajes medidos. Esta métrica fue diseñada para evaluar con precisión la eficacia con la que se lograba la maximización de la potencia en el canal deseado frente a la atenuación en el canal no deseado.

Durante el desarrollo del algoritmo de automatización, se estableció un conjunto de criterios definidos para la detención del algoritmo. El proceso se consideraba completado o era interrumpido si se cumplía alguna de las siguientes condiciones: el alcance de un valor umbral para la métrica de rendimiento considerada ideal, la finalización del número predefinido de

pasadas de optimización, o una intervención manual por parte del usuario.

Junto con estas condiciones de término, la búsqueda del algoritmo operaba dentro de restricciones mecánicas, estableciéndose un rango de barrido para los servomotores de 0 a 120 grados. Adicionalmente, se implementaron mecanismos de seguridad activos, como el análisis continuo de los voltajes, para evitar exceder los límites operativos (establecidos en 5V) que pudieran comprometer la integridad del microcontrolador Arduino.

3. **Desarrollar un gráfico en tiempo real para poder observar como varía el voltaje al momento de automatizar.**

   Para visualizar de forma continua cómo cambiaban los voltajes durante el funcionamiento del algoritmo, se desarrolló una herramienta gráfica. El objetivo era mostrar en tiempo real la evolución de los voltajes medidos en los dos fotodetectores, facilitando así la observación directa y el efecto de los ajustes de polarización.

   Esta herramienta de visualización se implementó utilizando el lenguaje de programación Python. Se recurrió a una biblioteca ampliamente utilizada para generar los diagramas visuales. Estos diagramas se integraron directamente en una ventana independiente dentro de la aplicación principal, permitiendo que el gráfico apareciera como una extensión de la interfaz de usuario. Este gráfico fue diseñado para presentar dos líneas, cada una representando el voltaje en el tiempo de uno de los fotodetectores.

   Con el fin de mostrar cómo variaban los voltajes a lo largo del tiempo, el algoritmo fue programado para guardar las mediciones más recientes. Se implementó un mecanismo para mantener un historial de un número limitado de las últimas mediciones de voltaje y los instantes de tiempo en que fueron tomadas. De esta forma, a medida que llegaban nuevas mediciones y el historial alcanzaba su capacidad máxima, la medición más antigua era borrada para dar lugar a la más nueva, asegurando que el gráfico siempre mostrara la información más relevante sin consumir recursos de memoria. Dado que el proceso de automatización que generaba los datos de voltaje

operaba de forma paralela e independiente al proceso que actualizaba el gráfico, se implementaron algunos tiempos adicionales para asegurar que el intercambio de datos entre estas dos tareas se realizara de manera ordenada y segura, evitando posibles errores en la visualización.

4. **Validar experimentalmente el sistema diseñado.**

   La validación experimental del sistema diseñado se llevó a cabo mediante la ejecución de diversas pruebas controladas. El proceso implicó la obtención y el análisis de los datos de voltaje de polarización en distintas condiciones de operación. Estos resultados fueron cruciales para determinar la efectividad global del sistema, evaluando tanto su comportamiento como sus limitaciones en diferentes escenarios. En particular, la métrica de rendimiento sirvió como una herramienta fundamental para cuantificar la eficacia con la que el sistema lograba los objetivos de polarización.

1.5. Alcances y limitaciones

1.5.1. Alcances

El objetivo de esta memoria de título es la optimización del estado de polarización (SOP). Específicamente, el sistema busca alcanzar de forma autónoma una configuración que maximice la potencia en una salida de polarización deseada, mientras atenúa simultáneamente la potencia en la salida ortogonal. Este proceso fue diseñado para ser eficiente, con la capacidad de operar independientemente de la potencia óptica de entrada y del estado físico inicial de la fibra. Todo el desarrollo y la validación experimental se llevaron a cabo utilizando los dispositivos en el laboratorio de la Universidad de Concepción.

1.5.2. Limitaciones

Esta memoria de titulo presenta limitaciones, como la velocidad de respuesta, que, al depender de servomotores, es del orden de los segundos. Adicionalmente, existen limitaciones mecánicas inherentes al diseño impreso en 3D del controlador, las cuales restringieron el rango de rotación de los servomotores a 120° para garantizar un funcionamiento fiable y evitar errores.

Otra limitación fundamental está definida por el rango de potencia óptica de entrada. Si bien el algoritmo demuestra ser robusto a bajas potencias, su rendimiento se ve comprometido a potencias elevadas. Esto no se debe a un fallo en la lógica del algoritmo, sino a una limitación física del Arduino.

Capítulo 2

Marco Teórico

2.1. Introducción

El presente capítulo establece el marco teórico y conceptual que sustenta este proyecto. Para ello, se abordarán los principios físicos fundamentales que rigen el sistema, comenzando por la naturaleza de la luz y la polarización, para luego profundizar en la birrefringencia inducida como mecanismo de control. Asimismo, se describirán los componentes tecnológicos clave y se introducirán los modelos matemáticos, como el Cálculo de Jones, que son esenciales para analizar y fundamentar el diseño. El capítulo concluye con la presentación y justificación del enfoque metodológico de optimización iterativa seleccionado, proporcionando así el contexto integral necesario para comprender la implementación y los resultados expuestos posteriormente.

2.2. Conceptos Fundamentales

2.2.1. Luz y sus propiedades.

La luz presenta una naturaleza dual, comportándose tanto como una corriente de partículas (fotones) como una onda. Para el estudio de los fenómenos ópticos abordados en esta memoria de título, y en particular para comprender la polarización, el modelo ondulatorio es el que más se relaciona con la memoria.

Según este modelo, la luz es una onda electromagnética transversal. Esto significa que está compuesta por un campo eléctrico ($\vec{E}$) y un campo magnético ($\vec{B}$) que oscilan de forma sinusoidal en el tiempo y en el espacio. Una característica fundamental de estas ondas es que ambos campos, el eléctrico y el magnético, son perpendiculares entre sí y, a su vez, son perpendiculares a la dirección de propagación de la onda. Para el análisis de la polarización, el comportamiento del campo eléctrico es importante, ya que la orientación de su oscilación en el plano transversal a la dirección de propagación es lo que define el estado de polarización de la luz, como se puede observar en la figura 2.1.

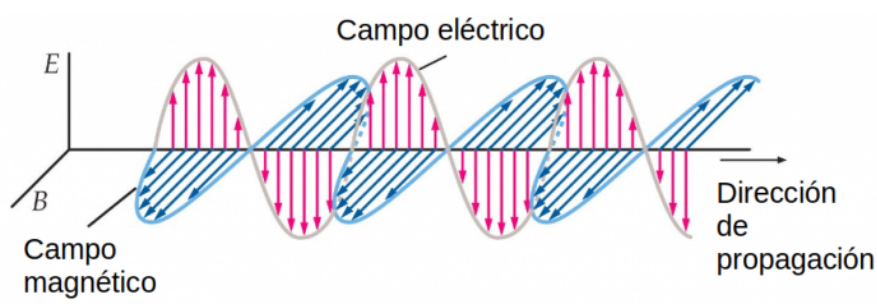


Fig. 2.1: Onda electromagnetica con su campo electrico y campo magnetico.

2.2.2. Tipos de Polarización.

El estado de polarización de la luz, que corresponde a la trayectoria descrita por el vector del campo eléctrico en el plano transversal a su propagación, se puede

clasificar en tres tipos principales, a pesar de que existen infinitas configuraciones:

1. **Polarización Lineal:** [25] Se define como aquella en la que el vector del campo eléctrico oscila en una dirección fija y recta dentro del plano transversal a la propagación. La orientación de esta puede variar, dando lugar a casos específicos como la polarización horizontal, la polarización vertical o la polarización lineal oblicua. la cual es de 45°.

2. **Polarización Circular:** [26] En este caso, el vector del campo eléctrico rota en un círculo a medida que la onda se propaga, manteniendo una magnitud constante. La rotación puede ser en dos sentidos: polarización circular derecha (RHCP) o polarización circular izquierda (LHCP), dependiendo de la dirección de giro. Este estado se produce cuando la luz está compuesta por dos componentes lineales ortogonales de igual amplitud, pero con una diferencia de fase de exactamente 90°.

3. **Polarización Elíptica:** [27] Ocurre cuando el vector del campo eléctrico traza una elipse en el plano perpendicular a la dirección de propagación. Este estado resulta de la superposición de dos componentes lineales ortogonales que tienen amplitudes desiguales y/o una diferencia de fase arbitraria (que no sea 0°, 180°, o 90° con amplitudes iguales).

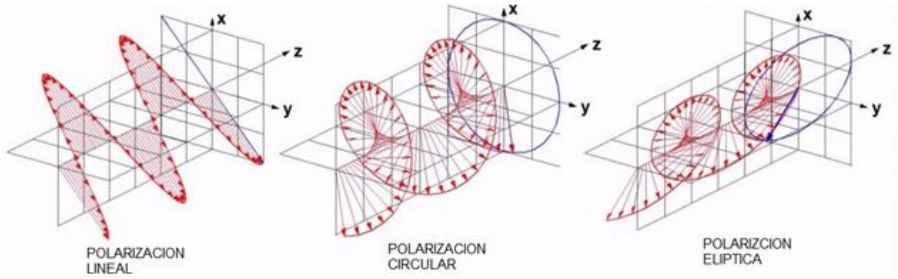


Fig. 2.2: Los 3 distintos tipos de polarización.

En la figura 2.2 se pueden observar los tres tipos de polarización, y cabe mencionar que estos estados no son independientes. De hecho, la polarización elíptica representa el caso más general, mientras que la polarización lineal y la circular son casos particulares de esta que ocurren bajo condiciones específicas de fase y amplitud.

2.2.3. Fibra Óptica

La fibra óptica es una guía de onda dieléctrica que transmite luz a lo largo de su eje. Su funcionamiento se basa en el principio de reflexión interna total. La fibra está compuesta por un núcleo (core) central de un material con un índice de refracción ($n_{\text{núcleo}}$) y un revestimiento (cladding) que lo rodea, con un índice de refracción ligeramente inferior ($n_{\text{revestimiento}}$). Cuando un rayo de luz viaja por el núcleo e incide en la interfaz entre ambos con un ángulo mayor que el llamado "ángulo crítico", la luz no se refracta, sino que se refleja completamente de vuelta hacia el núcleo. Este fenómeno se repite a lo largo de la fibra, confinando la luz y guiándola eficazmente a largas distancias con muy poca pérdida de señal.

Existen principalmente dos tipos de fibra óptica, cuya diferencia fundamental reside en el diámetro de su núcleo: la fibra monomodo (Single-Mode Fiber, SMF) y la multimodo (Multimode Fiber, MMF), como se ilustra en la Figura 2.2. La fibra SMF se caracteriza por tener un núcleo extremadamente pequeño, del orden de 9 µm, mientras que la fibra MMF posee un núcleo considerablemente más grande, típicamente de 50 µm o más. Esta diferencia geométrica es la que define el comportamiento de la luz en su interior.

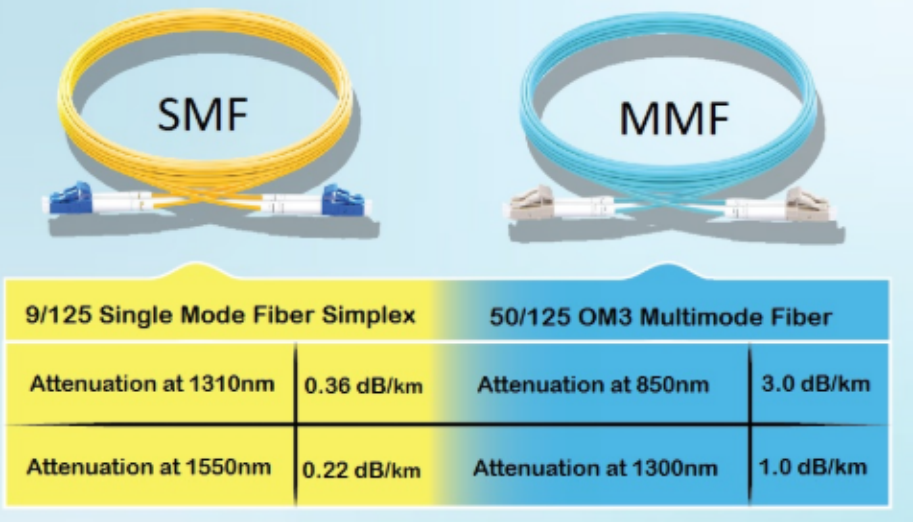


Fig. 2.3: Fibra Optica SMF y MMF.

La principal ventaja de la fibra monomodo, y la razón de su elección en

este trabajo, radica en que su núcleo reducido restringe la propagación de la luz a un único camino o ”modo”fundamental. Esto elimina por completo la dispersión modal, un efecto adverso presente en las fibras multimodo donde las diferentes trayectorias de la luz (múltiples modos) provocan un ensanchamiento y degradación de los pulsos de señal con la distancia, limitando así el ancho de banda y el alcance. Gracias a la ausencia de esta dispersión, la fibra SMF es el estándar para aplicaciones de alta velocidad y largas distancias, como las telecomunicaciones. Como se detalla en la Figura 2.3, la fibra SMF presenta una atenuación excepcionalmente baja en la ventana de operación de 1550 nm, con un valor típico de **0.22 dB/km**, lo que la hace ideal para la transmisión eficiente de señales en este rango. Por estas razones, se utilizó una fibra óptica monomodo estándar para la longitud de onda de 1550 nm en el desarrollo de este proyecto.

2.2.4. Birrefringencia

La birrefringencia es una propiedad óptica de ciertos materiales anisótropos en los que el índice de refracción depende de la orientación de la polarización de la luz que los atraviesa. Cuando un rayo de luz no polarizada incide sobre un material birrefringente, este se divide en dos rayos que viajan a velocidades distintas y están polarizados de forma ortogonal entre sí. Estas dos direcciones de polarización privilegiadas se conocen como los ejes ópticos del material (eje rápido y eje lento). Esta diferencia de velocidades introduce un desfase relativo entre las dos componentes de polarización. Este concepto es crucial, ya que es el principio físico que permite la existencia de componentes como las placas de onda y el funcionamiento del controlador de polarización.

Mientras que algunos materiales cristalinos son naturalmente birrefringentes, esta propiedad también puede ser inducida en materiales normalmente isótropos, es decir, un material cuyas propiedades son iguales en todas las direcciones. La birrefringencia inducida en una fibra óptica se genera al someterla a estrés mecánico, como presión, doblado o torsión [5] . Este fenómeno es la base del funcionamiento del controlador de polarización desarrollado en esta memoria. Las

paletas del controlador, al presionar la fibra, crean puntos de estrés localizado que actúan como pequeñas placas de onda variables, alterando de forma controlada el estado de polarización de la luz que se propaga por ella.

Para visualizar el efecto de la birrefringencia, se emplea la Esfera de Poincaré, un modelo donde cada punto sobre la superficie de una esfera representa un estado de polarización único. El paso de la luz a través de un material birrefringente (un retardador) se traduce en una rotación del estado de polarización sobre la esfera, describiendo siempre una trayectoria circular, como se ilustra en la Figura 2.4. Este modelo es útil para entender cómo las tres paletas del controlador, al actuar como retardadores variables, inducen rotaciones combinadas para transformar un estado de polarización de entrada en cualquier otro estado deseado, logrando así un control total.

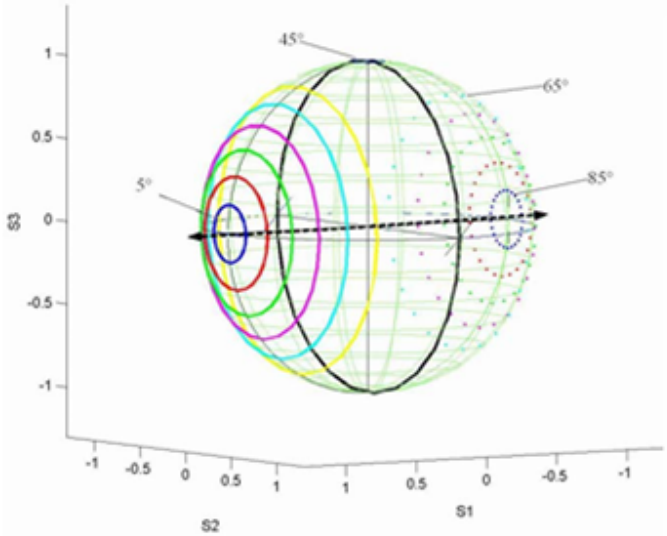


Fig. 2.4: Esfera Poincaré.[28]

2.3. Componentes Principales

2.3.1. Polarizing Beam Splitter (PBS)

Es un componente óptico cuya función es separar un haz de luz incidente en dos haces de salida con estados de polarización lineales y ortogonales entre sí. En este proyecto, el PBS recibe la luz proveniente del controlador de polarización y la divide en una componente de polarización horizontal y otra vertical, dirigiendo cada una hacia un fotodetector distinto[4].

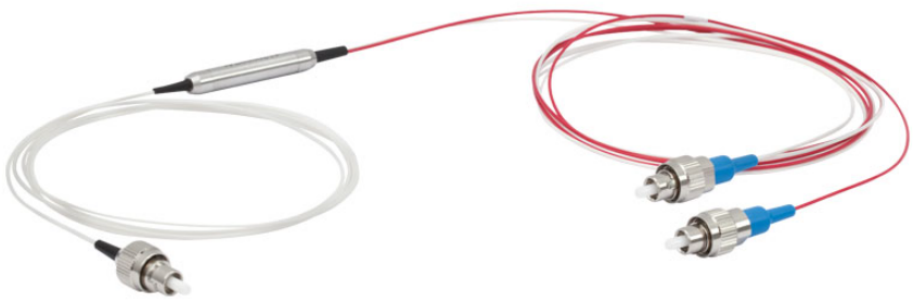


Fig. 2.5: Polarizing Beam Splitter(PBS). [4]

2.3.2. Fotodetector

Es un transductor optoelectrónico que convierte una señal óptica (luz) en una señal eléctrica (generalmente un voltaje o una corriente). Su respuesta es idealmente proporcional a la potencia óptica incidente. En este caso, los dos fotodetectores son los sensores que miden la potencia en cada una de las dos salidas del PBS, proporcionando la retroalimentación necesaria para el algoritmo de control[29].

Fig. 2.6: Fotodetector PDA05CF2 Thorlabs. [29]

2.3.3. Servomotor

Es un tipo de motor de corriente continua que permite un control preciso de su posición angular. A diferencia de un motor convencional que gira continuamente, un servomotor puede ser comandado para moverse y mantenerse en un ángulo específico dentro de su rango de operación. Esto se logra mediante un mecanismo de retroalimentación interno que compara la posición actual con la posición comandada. En este trabajo, los servomotores son los actuadores que mueven las paletas del controlador de polarización, ejecutando los ajustes calculados por el algoritmo de automatización[5].

2.3.4. Arduino

La placa Arduino es un dispositivo de código abierto, basado en un microcontrolador. En este trabajo, el Arduino UNO actúa como la unidad de control. Su función es recibir los comandos enviados desde el código en Python a través de la comunicación serial, y traducir estas instrucciones en señales eléctricas precisas (PWM) para manipular la posición angular de los servomotores. Adicionalmente, es el encargado de realizar la adquisición

de datos, leyendo los voltajes analógicos de los fotodetectores mediante su conversor analógico-digital (ADC) y transmitiendo esta información de vuelta al computador para el lazo de control[6] .

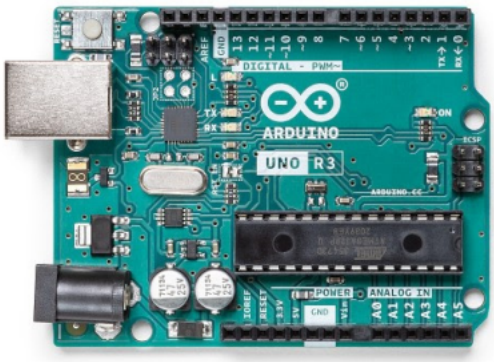


Fig. 2.7: Arduino UNO. [6]

2.3.5. LáserPRO800

La fuente de luz del sistema es un láser PRO800, un dispositivo que emite un haz de alta coherencia en una longitud de onda de 1550 nm. En el contexto de este trabajo, su rol es fundamental: proveer la señal óptica de referencia, la cual debe tener propiedades estables y un estado de polarización (SOP) conocido. Esta es la señal que se transmite a través de la fibra para ser modificada por el controlador y analizada por el sistema de detección, permitiendo así evaluar la eficacia del algoritmo de control[30].

Fig. 2.8: LaserPRO800. [30]

2.3.6. Osciloscopio

El osciloscopio es un instrumento de medición electrónico que permite visualizar señales eléctricas, representando gráficamente el voltaje en función del tiempo. En este trabajo, el osciloscopio fue una herramienta fundamental para la monitorización y validación. Se utilizó para observar en tiempo real la señal de voltaje proveniente de los fotodetectores, permitiendo una verificación visual inmediata del efecto de los ajustes manuales de las paletas. Además, fue crucial durante la etapa de calibración para contrastar y confirmar la exactitud de las lecturas de voltaje realizadas por el microcontrolador Arduino[31].

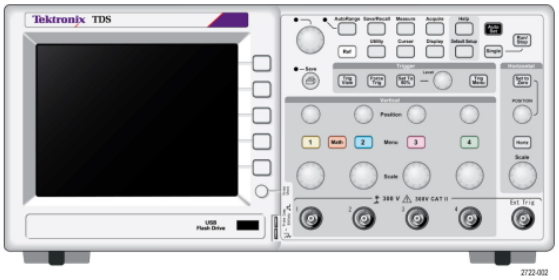


Fig. 2.9: Osciloscopio. [31]

2.3.7. Generador de Funciones arbitrarias

El generador de funciones es un instrumento capaz de producir diversas formas de onda eléctricas con parámetros controlables. Para los fines de este proyecto, su capacidad para generar niveles de voltaje de corriente continua (DC) precisos fue la de mayor utilidad. Su función en este trabajo fue exclusivamente para la etapa de calibración del sistema de adquisición de datos. Se empleó como una fuente de voltaje de referencia, conocida y estable, para verificar que las lecturas del conversor analógico-digital (ADC) del Arduino correspondían con exactitud a los valores de entrada medidos simultáneamente en el osciloscopio [32].

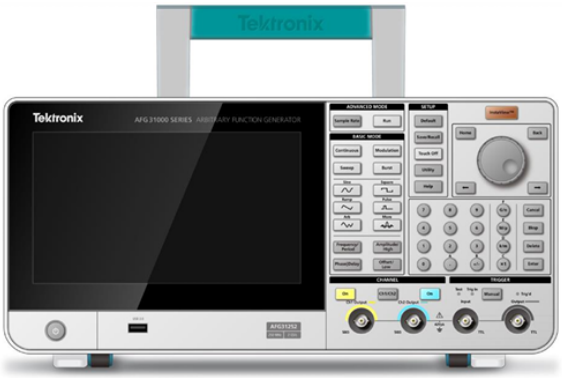


Fig. 2.10: Generador de funciones. [32]

2.4. Teorías y Principios Relevantes

2.4.1. Cálculo de Jones

Para describir y predecir de manera cuantitativa cómo un estado de polarización es modificado al pasar por diferentes componentes ópticos, se utiliza un formalismo matemático conocido como el Cálculo de Jones. Propuesto por R. C. Jones, este método fue diseñado para simplificar los cálculos en sistemas con varios elementos ópticos, que de otra manera podrían volverse “abrumadoramente complicados”. El cálculo reduce las ecuaciones a una forma

matricial y se distingue de otros métodos porque tiene en cuenta la fase absoluta de la onda luminosa.

Este modelo es aplicable a la luz monocromática y se basa en dos elementos principales: los vectores de Jones, que representan el vector de luz en forma compleja, y las matrices de Jones, que son operadores matriciales que se desarrollan para los distintos elementos ópticos. La gran ventaja de este formalismo es que el efecto de una serie de componentes se puede representar mediante el producto de sus matrices correspondientes, lo que "simplifica enormemente la tarea de calcular el efecto de muchos elementos". Además, este enfoque permite derivar teoremas generales sobre las combinaciones de dichos elementos [33].

2.4.1.1. Vector de Jones

Un estado de polarización cualquiera puede ser representado como un vector columna de 2x1, conocido como vector de Jones ($\vec{J}$). Este vector describe las amplitudes (A_x, A_y) y las fases (ϕ_x, ϕ_y) de las componentes del campo eléctrico a lo largo de dos ejes ortogonales (x e y):

$$\vec{J} = \begin{pmatrix} E_x \\ E_y \end{pmatrix} = \begin{pmatrix} A_x e^{i\phi_x} \\ A_y e^{i\phi_y} \end{pmatrix} \tag{2.1}$$

Normalizando la intensidad total a la unidad, los estados de polarización lineal más relevantes para este trabajo se representan de la siguiente manera:

- Polarización Lineal Horizontal: $\vec{J}_H = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$
- Polarización Lineal Vertical: $\vec{J}_V = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$
- Polarización Lineal a +45°: $\vec{J}_{45} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix}$

2.4.1.2. Matriz de Jones

Un componente óptico que altera la polarización (como una placa de onda o un polarizador) se representa mediante una matriz de 2x2, llamada matriz de Jones (**T**). La relación entre el estado de polarización de entrada ($\vec{J}_{\text{in}}$) y el de salida ($\vec{J}_{\text{out}}$) tras pasar por el componente se describe mediante una multiplicación matricial:

$$\vec{J}_{\text{out}} = \mathbf{T} \cdot \vec{J}_{\text{in}} \tag{2.2}$$

Este formalismo permite modelar sistemas ópticos complejos simplemente multiplicando las matrices de Jones de cada componente en el orden en que la luz los atraviesa.

2.4.2. Principio de Control de Polarización Mecánico

El controlador de polarización utilizado en este trabajo, basado en la aplicación de estrés mecánico sobre la fibra, funciona teóricamente como una serie de placas de onda (o retardadores) variables y rotables [28]. El principio físico se basa en la birrefringencia inducida previamente descrita.

Cuando una de las paletas del controlador presiona la fibra óptica, induce un estrés localizado. Este estrés provoca que el vidrio de la fibra se vuelva birrefringente en ese punto, creando un “eje lento” y un “eje rápido”. La magnitud de la presión ejercida por el servomotor determina la cantidad de desfase (ϵ) entre las componentes de la luz, mientras que el ángulo físico de la paleta determina la orientación (θ) de estos ejes. Por lo tanto, cada paleta actúa como una placa de onda de retardo y orientación variables.

El poder de un controlador de tres paletas, como el implementado, radica en la cascada de estos efectos. Utilizando el Cálculo de Jones [33], el efecto total del controlador sobre la luz se puede modelar como el producto de las matrices de Jones de cada una de las tres paletas:

$$\mathbf{T}_{\text{total}} = \mathbf{T}_{\text{paleta3}}(\theta_3, \epsilon_3) \cdot \mathbf{T}_{\text{paleta2}}(\theta_2, \epsilon_2) \cdot \mathbf{T}_{\text{paleta1}}(\theta_1, \epsilon_1) \tag{2.3}$$

De esta forma, el estado de polarización final a la salida del controlador es $\vec{J}_{\text{out}} = \mathbf{T}_{\text{total}} \cdot \vec{J}_{\text{in}}$. Al poder ajustar de forma independiente los parámetros de cada paleta, es posible construir una matriz de transformación total que puede convertir cualquier estado de polarización de entrada en cualquier estado de polarización de salida deseado. Este es el principio fundamental que permite la automatización, ya que el algoritmo busca la combinación de ajustes que genere la transformación necesaria para optimizar la señal.

2.4.3. Controlador de polarización

El controlador de polarización desarrollado en este trabajo es un dispositivo de fibra óptica del tipo mecánico, cuyo diseño se fabricó mediante impresión 3D basándose en la estructura de los controladores de paletas comerciales, como los modelos MPC220 y MPC320 de Thorlabs[5] . El principio de funcionamiento de este tipo de dispositivo se basa en la **birrefringencia inducida por estrés**, un concepto detallado anteriormente.

Estos controladores funcionan enrollando la fibra óptica en bucles sobre paletas independientes. Cada una de estas paletas actúa como una **placa de onda fraccional** (retardador), alterando el estado de polarización (SOP) de la luz. La cantidad de retardo de fase (ϕ) que introduce cada paleta no es arbitraria, sino que depende de parámetros físicos bien definidos como el número de vueltas de fibra (N), el diámetro del bucle (D), el diámetro del revestimiento de la fibra (d) y la longitud de onda de operación (λ). Esta relación se puede estimar mediante la siguiente ecuación:

$$\phi = a \cdot \frac{Nd^2}{\lambda D} \tag{2.4}$$

donde 'a' es una constante que depende del material de la fibra (para la sílice, $a \approx 0{,}133$). El control efectivo del SOP se logra mediante la rotación de estas paletas. Al girar una paleta, se cambia la orientación del eje rápido de la birrefringencia inducida con respecto a la polarización de la luz incidente, lo que permite transformar el estado de polarización.

Una configuración de tres paletas, como la del modelo MPC320 en la que se

inspira este trabajo, es particularmente versátil. Teóricamente, puede funcionar como una placa de cuarto de onda, una de media onda y otra de cuarto de onda (configuración QHQ). Esta combinación permite, en principio, transformar cualquier estado de polarización de entrada en cualquier otro estado de polarización de salida deseado. La primera paleta convierte la polarización de entrada en una polarización lineal; la segunda rota esta polarización lineal al ángulo requerido; y la tercera la transforma al estado final deseado.

Para la implementación específica de esta memoria de título, se determinó experimentalmente la configuración de las vueltas de fibra en cada paleta para la longitud de onda de operación de 1550 nm. Se optó por una configuración de 1-2-1 vueltas (una vuelta en la primera paleta, dos en la central y una en la última). Esta decisión surgió de la necesidad de encontrar un punto medio entre las especificaciones de diferentes modelos comerciales y las limitaciones prácticas del montaje. Durante las pruebas, se constató que un número excesivo de vueltas, si bien induce un mayor retardo de fase, también introduce pérdidas de potencia óptica (pérdidas de inserción) significativas. Por lo tanto, la configuración 1-2-1 se estableció como un compromiso óptimo entre la obtención de un rango de retardo de fase suficiente para un control completo del SOP y la minimización de las pérdidas para el sistema.

2.4.4. Conversión de Potencia Óptica a Voltaje

Para que nuestro sistema de control funcione, el microcontrolador Arduino necesita saber cuánta potencia óptica está llegando a los sensores. Sin embargo, el Arduino no mide potencia óptica, sino voltaje. Por lo tanto, fue fundamental establecer una relación fiable y matemática entre la potencia óptica (en Watts) que incide en el fotodetector y el voltaje (en Voltios) que el Arduino finalmente lee [29].

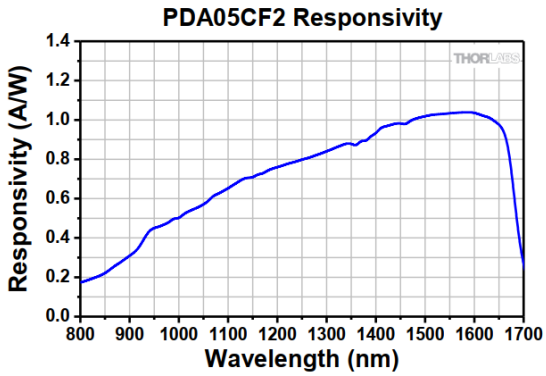


Fig. 2.11: Responsividad del fotodetector. [29]

El fotodiodo de InGaAs dentro del detector genera una fotocorriente (I_p) proporcional a la potencia óptica incidente. La eficiencia de esta transformación a la longitud de onda de trabajo (1550 nm) viene definida por la Responsividad ($R(\lambda)$) del detector. Consultando el gráfico de la responsividad del fotodetector 2.11, estimamos este valor en 1.05 Amperios por Watt (A/W).

Esta fotocorriente es extremadamente pequeña, por lo que el detector la pasa por una segunda etapa: un amplificador de transimpedancia (TIA). La misión de este circuito es convertir esa minúscula corriente en un nivel de voltaje manejable y proporcional. La magnitud de esta amplificación, conocida como Ganancia de Transimpedancia (G_{TI}), no es un valor universal; depende críticamente de las características del circuito al que se conecta, un detalle técnico que resultó ser clave en este montaje.

Esto conduce a una consideración fundamental: la impedancia de carga. La salida del fotodetector no se conecta a un solo dispositivo, sino a dos en paralelo: la entrada analógica del Arduino, con una impedancia altísima de 100 MΩ, y un canal del osciloscopio, con una impedancia de 1 MΩ. Para determinar el comportamiento del detector, es necesario conocer la impedancia de carga total o equivalente ($R_{carga,eq}$) que este ve.

Cuando dos resistencias se conectan en paralelo, la resistencia equivalente se calcula con la fórmula:

$$\frac{1}{R_{carga,eq}} = \frac{1}{R_{Arduino}} + \frac{1}{R_{Osciloscopio}} \tag{2.5}$$

Sustituyendo los valores de los instrumentos utilizados:

$$\frac{1}{R_{carga,eq}} = \frac{1}{100 \times 10^6\,\Omega} + \frac{1}{1 \times 10^6\,\Omega} \tag{2.6}$$

Al resolver la suma de fracciones, nos da un valor de $1{,}01 \times 10^{-6}\,\Omega^{-1}$. Al invertir este resultado para encontrar $R_{carga,eq}$, obtenemos:

$$R_{carga,eq} = \frac{1}{1{,}01 \times 10^{-6}} \approx 0{,}99 \times 10^6\,\Omega \tag{2.7}$$

Es decir, una impedancia equivalente de aproximadamente 0.99 MΩ. En el contexto de la instrumentación óptica, este valor se considera sin lugar a dudas como alta impedancia (Hi-Z). Según el manual del fabricante, operar en este régimen de Hi-Z nos obliga a usar una ganancia de transimpedancia específica de 1×10^4 **V/A**.

Con estos dos factores definidos, se puede ensamblar la ecuación que gobierna la conversión en nuestro sistema[29] :

$$V_{OUT} = P \times R(\lambda) \times G_{TI} \tag{2.8}$$

Al sustituir los valores que se justifican para este montaje experimental:

$$V_{OUT}(V) \approx P(W) \times 1{,}05(A/W) \times 10000(V/A) \tag{2.9}$$

Se obtiene la relación matemática que el software utilizará para interpretar las mediciones:

$$V_{OUT} \approx 10500 \cdot P \tag{2.10}$$

Esta fórmula se convirtió en la piedra angular del lazo de control, lo que permite calcular la potencia óptica a partir del voltaje medido, simplemente despejando P:

$$P(W) \approx \frac{V_{OUT}}{10500} \tag{2.11}$$

2.4.5. Lectura del Arduino

La función principal del microcontrolador Arduino en este sistema es la adquisición y preprocesamientode las señales eléctricas. Este proceso se desglosa en tres etapas fundamentales: detección, traducción y envío de la información[6].

En la etapa de detección, el Arduino capta las señales eléctricas del fotodetector a través de sus pines de entrada analógica (A0 y A1). Internamente, el microcontrolador utiliza un Convertidor Analógico-Digital (ADC) con una resolución de 10 bits. Esto significa que el ADC puede transformar el voltaje continuo (analógico) de la señal en un valor digital discreto, que va desde 0 (correspondiente a 0V) hasta 1023 (correspondiente a 5V, que es el voltaje de referencia de la placa Arduino). La función analogRead() es la encargada de realizar esta lectura y obtener el valor crudo del sensor.

Una vez que el Arduino ha detectado el valor crudo, procede a su traducción o procesamiento. Este valor digital en el rango 0-1023 se convierte a una escala de voltaje comprensible (0-5V) utilizando la siguiente fórmula de escalado:

$$\text{voltaje} = \frac{\text{valor_crudo}}{1023{,}0} \times 5{,}0$$

Esta operación es crucial, ya que el resultado es directamente interpretable y utilizable para calcular la potencia óptica. Este paso de ”traducción”se aplica individualmente a cada señal proveniente de los fotodetectores conectados a los pines A0 y A1.

Finalmente, el Arduino se encarga del envío de la información procesada al código de Python a través de la comunicación serie. Utilizando la función Serial.print(), el Arduino transmite el valor de voltaje del primer fotodetector, seguido de una

coma (que actúa como un delimitador). Inmediatamente después, el valor de voltaje del segundo fotodetector es enviado, seguido de un carácter de nueva línea mediante Serial.println(). Esta secuencia asegura que el script de Python pueda leer una línea completa de datos que contenga ambos valores de voltaje, listos para su procesamiento posterior y utilización en la interfaz o en la lógica de automatización. La comunicación se establece a una velocidad de 115200 baudios, garantizando un flujo de datos rápido y confiable.

2.5. Enfoque Metodológico

Para abordar el control automatizado de la polarización, se empleará una metodología de optimización iterativa ”modificar y observar”. Se opta por este enfoque empírico debido a la complejidad de modelar analíticamente la relación no lineal entre la rotación de las paletas del controlador y la potencia óptica resultante, permitiendo así que el sistema converja a una solución óptima de forma adaptativa.

La implementación de este ciclo de control se apoyará en un protocolo de comunicación serie entre un script en Python y un microcontrolador Arduino. Esta plataforma centralizará la lógica del algoritmo de optimización y el procesamiento de las señales de los fotodetectores. Para esto último, se utilizarán técnicas de promediado de lecturas para asegurar la fiabilidad de los datos de retroalimentación, mitigando el efecto del ruido inherente a las mediciones experimentales.

Capítulo 3

Desarrollo

3.1. Introducción

Este capítulo detalla el proceso de diseño y construcción del sistema de control de polarización. Se describe el montaje de los componentes de los dispositivos y el desarrollo del software de control, poniendo especial énfasis en la evolución del algoritmo de optimización. Se expone el recorrido metodológico que, partiendo de estrategias iniciales, permitió identificar sus limitaciones y culminó en el diseño de una solución de dos etapas más robusta y eficaz. Así, este capítulo narra el proceso de ingeniería que transformó los fundamentos teóricos en un prototipo funcional y validado.

3.2. Metodología de Implementación

1. **Revisión de Componentes y Configuración del Montaje Experimental (Setup).**

   Primero, se tuvo que montar el setup, donde se verificó que era necesario fusionar una fibra con conectores FC/APC en un extremo y FC/UPC en

el otro, debido a la falta de otra fibra disponible. Para realizar la fusión, se utilizaron los kits MiniLite-Strip [34], la guillotina (GW-750) [35] y la fusionadora (KL-300T) [36]. Después de que la fusión se completó con éxito, se montó la fibra en el setup.

En segundo lugar, se realizó el montaje del setup una vez que se obtuvieron todos los instrumentos necesarios. Se utilizó un láser (PRO-800), que se conectó mediante fibra al controlador de polarización, el cual se basó en el MPC320 de Thorlabs, ya que esa base y forma se usó para imprimir un modelo en la impresora 3D, el cual cuenta con tres servomotores. El dispositivo impreso es controlado mediante Arduino, que a su vez está vinculado a una notebook. Desde la notebook, se ejecuta el código de Python y Arduino.

Dado que la longitud de la fibra era considerablemente mayor en relación con el espacio, se usó un carrete para la fibra óptica, para mantenerla ordenada y evitar daños. El carrete se ancló firmemente a la mesa, para que no se mueva y no se corte la fibra.

Después de que la fibra pasa por el controlador, la señal llegó hasta un divisor de haz polarizante (PBS). Este componente óptico separaba la luz en dos componentes de polarización ortogonales (horizontal y vertical), generando así dos salidas de fibra independientes. Cada una de estas fibras se conectó a un fotodetector individual, encargado de convertir la señal óptica incidente en una señal eléctrica (voltaje) proporcional a su potencia. Para el monitoreo y la adquisición de estas señales eléctricas, la salida de cada fotodetector (mediante un cable BNC) se conectó a la entrada de un conector en T. Una salida de cada conector en T se dirigió a un canal del osciloscopio, permitiendo la visualización directa de la variación del voltaje. La otra salida de cada conector en T se enlazó, mediante un adaptador BNC a pines, a las entradas analógicas del microcontrolador Arduino. Esta configuración posibilitó la lectura simultánea de los voltajes tanto por el osciloscopio como por el Arduino para el procesamiento de datos y el control de un lazo cerrado.

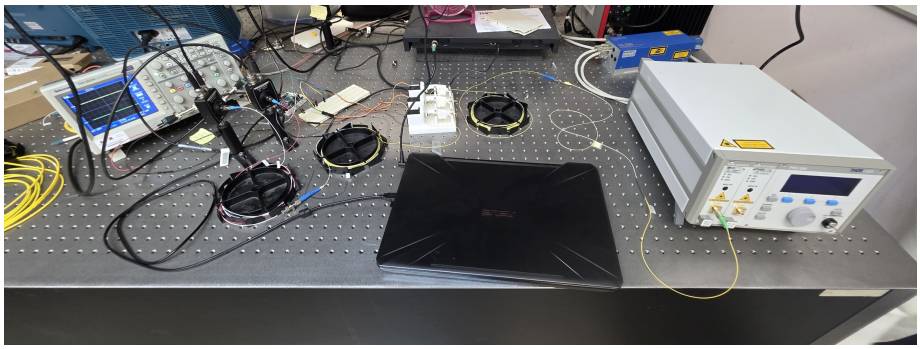


Fig. 3.1: Montaje del Setup. Fuente: Autoridad Propia

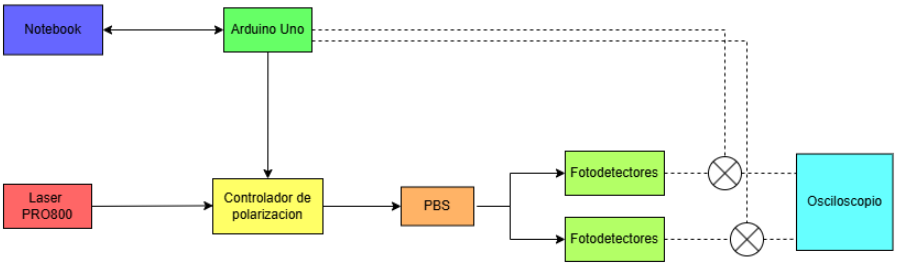


Fig. 3.2: Diagrama del Setup. Fuente: Autoridad Propia

En la figura 3.1 y el diagrama 3.2 se observa el montaje de los dispositivos implementado en el laboratorio. Se destaca que, para cada dispositivo, se consultó su respectiva hoja de datos.

2. **Diseño y Desarrollo Inicial de la Interfaz Gráfica de Usuario (GUI) en Python.**

   El desarrollo inicial del código comenzó con la creación de la Interfaz Gráfica de Usuario (GUI) en Python, utilizando la biblioteca Tkinter. En esta primera etapa de diseño, se definieron los elementos visuales y la disposición de los botones, planificando su funcionalidad antes de la implementación.

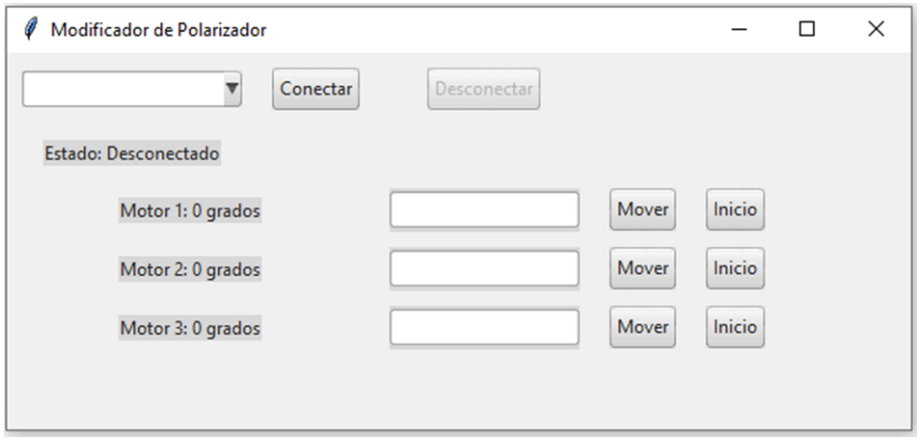


Fig. 3.3: Interfaz Inicial. Fuente: Autoridad Propia.

Como se puede observar en la imagen, la estructura de la GUI contempló una sección de la conexión con el dispositivo Arduino, esta incluía un selector de puertos seriales y los botones visuales para “Conectar” y “Desconectar”. Se diseñó también un área dedicada al control manual de los tres servomotores, donde para cada servo se dispuso un espacio para ingresar el ángulo deseado, un botón para enviar la instrucción de movimiento y un botón de “Inicio” destinado a retornar el servo a la posición de cero grados. Posteriormente, se incorporaron botones con la intención de que, más adelante, permitieran solicitar y visualizar en la interfaz el voltaje detectado para cada una de las polarizaciones de forma individual. Todos estos elementos se crearon inicialmente como componentes visuales, para posteriormente emplearlos.

3. **Establecimiento de la Comunicación Serial Python-Arduino y lectura de voltaje.**

   Para lograr la manipulación precisa de las paletas del controlador de polarización, se estableció un sistema de comunicación serial entre el código principal por Python y el microcontrolador Arduino. La preparación de esta comunicación comenzaba en el propio Arduino, donde, al iniciarse, se configuraba el puerto serial para operar a una velocidad de 115200 baudios. De forma paralela, se inicializaban los tres servomotores, asignando cada

uno a un pin digital específico en el Arduino y estableciendo una posición angular inicial de cero grados, mediante la librería Servo.h.

Desde el código en Python, su interfaz gráfica de usuario (GUI) ofrecía al usuario las herramientas para gestionar esta conexión. El primer paso era la selección del puerto serial correcto, facilitada por una lista desplegable que enumeraba los puertos detectados en el sistema. Una vez seleccionado el puerto, al accionar un botón de “Conectar”, el código de Python procedía a establecer formalmente la comunicación con el Arduino. La recepción de un mensaje de confirmación desde el Arduino, servía como señal de que el enlace bidireccional estaba operativo.

El control efectivo de los servomotores se basó en un protocolo de mensajes de texto simples. Cuando el usuario, a través de la GUI, ingresaba un ángulo deseado para un servomotor específico y activaba la orden de movimiento, el código en Python validaba primero la entrada. Luego, construía una cadena de texto con un formato predefinido, por ejemplo: “set_angle S A”, donde S identificaba el servo y A el ángulo. Esta cadena se codificaba y se transmitía al Arduino a través del enlace serial. El código de Python también actualizaba inmediatamente la interfaz gráfica para mostrar el nuevo ángulo que se había modificado.

El código desarrollado para el microcontrolador Arduino operaba dentro de un bucle de ejecución continua, verificando constantemente la disponibilidad de datos en el puerto serial. Al recibir una cadena de texto desde el computador, el Arduino la analizaba para determinar si correspondía a un comando de movimiento de servomotor válido. Si la instrucción era reconocida, el programa extraía de ella el identificador del servomotor y el valor del ángulo solicitado. Antes de ejecutar la acción, se aplicaba una restricción a este valor para asegurar que se mantuviera dentro del rango operativo de los servomotores, establecido entre 0 y 120 grados. Finalmente, a través de las funciones de la librería Servo.h, se generaba y enviaba la señal de control al servomotor indicado, provocando su desplazamiento físico al ángulo especificado. Este mecanismo de

interacción permitió un ajuste manual y directo de cada servomotor del controlador de polarización.

Una vez establecido el control manual de los servomotores mediante la comunicación serial entre el computador y el Arduino, el siguiente paso fundamental fue asegurar la correcta y fiable adquisición de datos de voltaje por parte del microcontrolador. Esta validación era muy importante antes de utilizar el Arduino para leer las señales provenientes de los fotodetectores en el sistema de control de polarización.

Para este propósito, se implementó un procedimiento de calibración y verificación utilizando un generador de funciones como fuente de voltaje controlada y conocida. La salida del generador de funciones se conectó de forma simultánea tanto a un canal del osciloscopio como a una de las entradas analógicas del Arduino. Esta conexión en paralelo se facilitó mediante el uso de un conector en T BNC. El osciloscopio proporcionaba una lectura de referencia precisa y visual del nivel de voltaje suministrado por el generador de funciones.

El proceso consistió en configurar el generador de funciones para emitir diferentes niveles de voltaje de corriente continua (DC) dentro del rango operativo del Arduino (típicamente 0-5V). Para cada nivel de voltaje configurado, se observaba la medición en el osciloscopio y, al mismo tiempo, se programaba el Arduino para leer el valor de su pin analógico y convertir esta lectura digital (un valor entre 0 y 1023 para un ADC de 10 bits) nuevamente a una representación de voltaje mediante la fórmula estándar:

$$V_{calculado} = \frac{\text{lectura_ADC}}{1023{,}0} \times V_{\text{referencia_ADC}} \tag{3.1}$$

Los valores de voltaje calculados por el Arduino se enviaban al computador a través de la comunicación serial y se comparaban con los valores mostrados en el osciloscopio. Este “entrenamiento” o proceso de comparación permitió verificar la precisión de la conversión analógico-digital del Arduino y confirmar la correcta implementación de la fórmula de conversión en el

código.

Este procedimiento aseguró que el Arduino fuera capaz de interpretar y reportar con precisión los niveles de voltaje, una capacidad importante para el monitoreo de las señales de los fotodetectores.

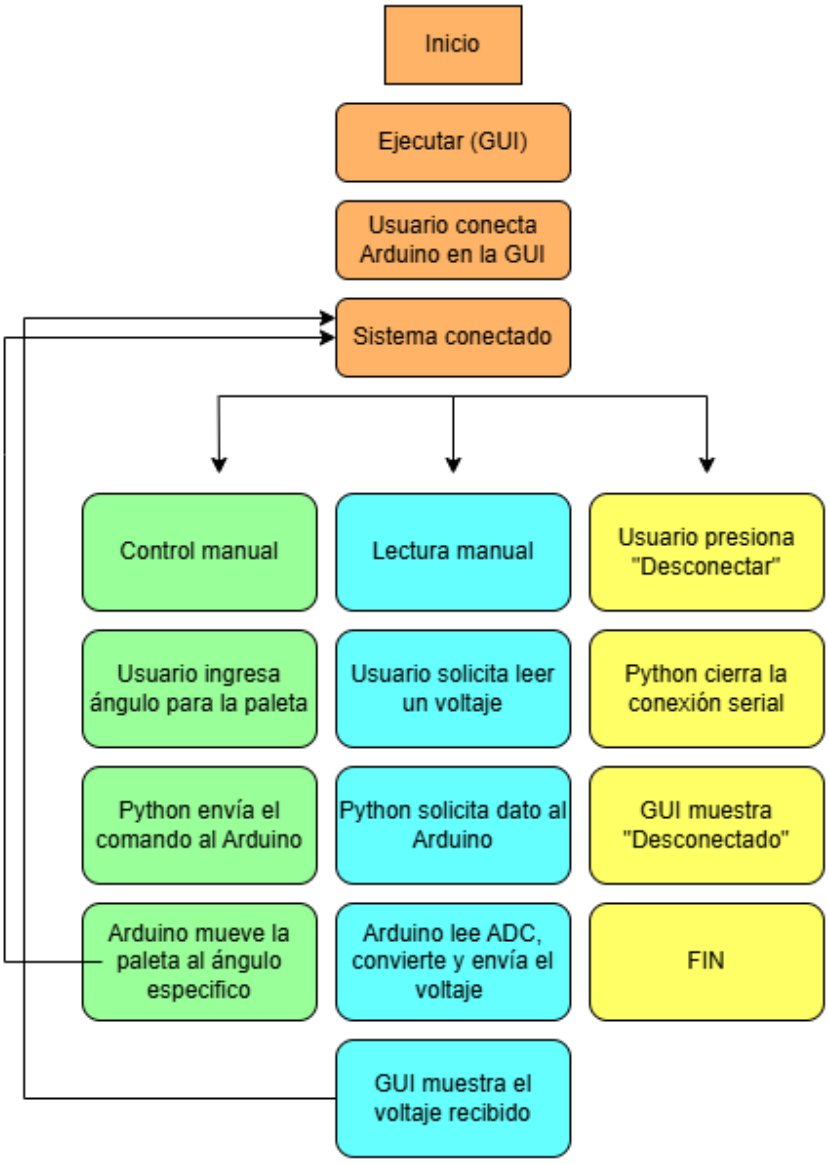


Fig. 3.4: Diagrama de Flujo Básico. Fuente: Autoridad Propia.

El diagrama 3.4 detalla el flujo del proceso de operación manual del sistema.

El código comienza con la iniciación y el establecimiento de la comunicación serial entre la interfaz gráfica (GUI) y el Arduino, llevando al sistema a un estado central y operacional de ”Sistema conectado”. Desde este punto, se divide en tres acciones disponibles para el usuario: el control manual para el ajuste físico de los servomotores, la lectura manual para el monitoreo de los voltajes y la desconexión para finalizar la sesión de forma controlada.

Es muy importante destacar el diseño cíclico, donde las flechas de retorno al estado central indican que el usuario puede ejecutar acciones de control y lectura de forma repetida e indefinida.

4. **Desarrollo y Evolución del Algoritmo de Automatización.**

Con el sistema de control manual y detección de voltajes operativos, el siguiente desafío fue desarrollar un algoritmo para automatizar el ajuste del estado de polarización. El objetivo principal era lograr que el sistema, de forma automática, configurara los servomotores del controlador para maximizar la potencia óptica en una salida de polarización seleccionada, mientras simultáneamente minimizaba la potencia en la salida ortogonal. Este proceso de desarrollo fue iterativo, explorando diversos métodos antes de definir el método final implementado.

Para facilitar estas pruebas iniciales de automatización, la interfaz gráfica de usuario (GUI) ya había sido adaptada para permitir al operador definir dos parámetros clave: el ”ángulo de paso”, para determinar la magnitud de los incrementos angulares, y el “número de barridos” o pasadas completas. Se utilizaron en esta primera fase para controlar uno de los primeros enfoques implementados: una estrategia de búsqueda local secuencial, conceptualmente similar a un algoritmo de ascenso de colina (Hill Climbing), aplicado de forma individual a cada servomotor. La lógica de este método consistía en mover un servomotor un pequeño incremento angular, utilizando el ”ángulo de paso”definido en la GUI. Tras este ajuste, se medía el voltaje y, si no resultaba en una mejora, se probaba un movimiento de igual magnitud en la dirección opuesta. Si ninguna de

las dos perturbaciones producía un incremento en el voltaje deseado, el algoritmo consideraba que ese servomotor había alcanzado un valor óptimo y procedía a aplicar el mismo proceso al siguiente servomotor.

Sin embargo, pronto se observó que este enfoque, si bien era relativamente simple de implementar, presentaba una limitación significativa, ya que tendía a finalizar su búsqueda en máximos locales de potencia. Dicho de otro modo, el algoritmo podía detenerse en un ángulo que era óptimo únicamente con respecto a sus ángulos vecinos cercanos, pero no necesariamente la mejor solución global posible en todo el rango de operación. Esta incapacidad para distinguir entre un óptimo local y el óptimo global deseado evidenció la necesidad de explorar otro tipo de estrategias.

Se investigó un cambio de enfoque con el fin de mejorar los resultados de la automatización. Una consideración inicial fue la optimización de los parámetros temporales. Se procedió a analizar y ajustar tanto el tiempo de detección de voltaje como el tiempo asignado para el movimiento de los servomotores. Tras diversas pruebas experimentales, se establecieron rangos de tiempo que demostraron ser razonables para que la operación sea fiable, ya que con menores o mayores tiempos se detectaron muchos problemas. En este sentido, cuando se inicia una lectura de voltaje, el código en Python primero envía una solicitud al Arduino, introduciendo una breve pausa de 0.05 segundos. Antes de leer la respuesta, el código espera 0.1 segundos adicionales, permitiendo la transmisión, conversión y retorno de los datos. Así, cada operación de lectura de voltaje conlleva una demora de aproximadamente 0.15 segundos. De forma análoga, para el movimiento de un servomotor, al enviar el comando existe la misma pausa inicial de 0.05 segundos, luego, para permitir el desplazamiento físico completo del servomotor, se introduce una pausa sustancial de 0.3 segundos. En consecuencia, cada comando de movimiento de servo implica una demora total de 0.35 segundos. Estas pausas deliberadas resultaron cruciales para la estabilidad, evitando que el código intentara leer datos prematuramente o emitir nuevos comandos antes de la finalización de los anteriores.

Durante la investigación de alternativas, se encontró el concepto de Optimización por Coordenadas (Coordinate Descent) o Optimización Alterna. El principio fundamental de este tipo de métodos radica en descomponer un problema de optimización multivariable complejo en una serie de subproblemas más simples, donde cada variable se optimiza individualmente mientras las demás se mantienen temporalmente fijas, repitiendo el proceso. Se decidió adoptar la idea general de esta optimización como guía para el diseño del nuevo algoritmo de automatización.

A diferencia de los enfoques anteriores que alternaban entre los servomotores en busca de mejoras de voltaje, el nuevo método se centró en optimizar completamente un servomotor antes de proceder con el siguiente. Es decir, el algoritmo buscaba el ángulo que produjera el voltaje más cercano al óptimo o el óptimo, para el primer servomotor y, solo una vez encontrado y fijado esa posición, comenzaba el proceso de optimización para el segundo, y así sucesivamente. Este razonamiento se materializó en un barrido angular más sistemático para cada servomotor de forma individual. Para un servomotor seleccionado, el algoritmo lo desplazaba a través de su rango operativo completo (de 0 a 120 grados), registrando el voltaje del detector objetivo en cada posición. Al finalizar la búsqueda de un servomotor, se identificaba el ángulo que había producido el mayor voltaje, y el servomotor se posicionaba en dicho punto.

Un aspecto crucial de este método fue la incorporación de umbrales de voltaje para optimizar la eficiencia y la seguridad. Si durante el barrido el voltaje medido alcanzaba un valor dentro del rango considerado óptimo y seguro (establecido entre 4.8V y 4.99V), la exploración para esa paleta específica se detenía de forma anticipada. Dicha posición se registraba entonces como la óptima para esa pasada, y el algoritmo procedía a analizar el siguiente servomotor. Por otro lado, si se registraba un valor de voltaje superior o igual a 5V, indicativo de una posible saturación o riesgo para el Arduino, ese ángulo era descartado y no se consideraba para la selección

del óptimo. Este método demostró una mejora significativa en la capacidad de encontrar valores de voltaje más altos en comparación con los enfoques de búsqueda local iniciales.

No obstante, incluso esta estrategia de barrido por servomotor, aunque más efectiva en la búsqueda de voltajes elevados, presentaba inicialmente una carencia en términos de eficiencia. El algoritmo estaba programado para completar de forma obligatoria el análisis de las tres paletas en cada pasada, aun cuando ya se hubiera encontrado un voltaje dentro del rango óptimo en una de las primeras paletas. Se reconoció que continuar la búsqueda en las paletas restantes, una vez alcanzado un nivel de voltaje suficientemente alto y seguro, no siempre conducía a una mejora y podía representar una inversión de tiempo innecesaria.

Por lo tanto, la lógica del algoritmo fue refinada para incorporar una condición de detención anticipada. Con esta modificación, si en cualquier punto del barrido de un servomotor el sistema alcanzaba un estado de rendimiento considerado óptimo y seguro, el proceso de automatización se terminaba por completo, sin necesidad de continuar con el resto de la búsqueda. Esta interrupción prematura evitó la necesidad de analizar los servomotores siguientes o de ejecutar pasadas adicionales, optimizando así el tiempo de operación.

Finalmente, a pesar de las mejoras, se detectó una nueva forma para incrementar la precisión del algoritmo. Se observó que, en ocasiones, al finalizar el barrido y posicionar un servomotor en el ángulo identificado como el mejor, el valor de voltaje medido no siempre correspondía con exactitud al máximo registrado durante el barrido general. Para disminuir esta pequeña discrepancia, se implementó un ajuste fino. Este procedimiento se activaba después de que el barrido grueso identificara la región angular más prometedora. En lugar de simplemente aceptar ese ángulo, el algoritmo realizaba una segunda búsqueda mucho más localizada, la cual exploraba el rango comprendido entre 3 grados por debajo y 3 grados por encima del ángulo seleccionado, utilizando incrementos de tan solo 1 grado. En cada

uno de estos ángulos, se volvía a medir el voltaje, seleccionando el ángulo que ofrecía el mejor rendimiento dentro de este pequeño intervalo.

Para maximizar la precisión y la eficacia del algoritmo, la etapa de ajuste fino se diseñó para activarse de forma dinámica en dos escenarios clave durante el proceso de optimización. Se aplicaba de manera intermedia si, durante el transcurso del barrido grueso, se identificaba un nuevo ángulo con un valor de voltaje superior al máximo registrado hasta ese momento; en dicha situación, la exploración principal se pausaba para refinar la búsqueda alrededor de este nuevo punto prometedor. Asimismo, se realizaba un ajuste fino final al concluir por completo la exploración de una paleta. Este último ajuste se aplicaba sobre el mejor ángulo global encontrado durante todo ese barrido, con el fin de verificar la posición óptima para esa pasada. En ambos escenarios, el ángulo que ofrecía el rendimiento más alto dentro de este pequeño intervalo de búsqueda de precisión era seleccionado como la nueva posición óptima.

Si bien la estrategia de barrido de dos etapas (grueso y fino) demostró ser efectiva para encontrar un alto voltaje en la polarización objetivo, se observó que este resultado por sí solo no siempre cumplía con el objetivo completo. Se presentaron casos en los que, a pesar de alcanzar un rango de voltaje óptimo para la señal a maximizar, el voltaje en el canal no objetivo permanecía considerablemente elevado; es decir, no estaba suficientemente mitigado o atenuado.

La hipótesis de trabajo inicial se basaba en que ambos canales de polarización tendrían un comportamiento mayoritariamente inverso; es decir, que al maximizar la potencia en el canal objetivo, la potencia en el canal ortogonal se minimizaría de forma natural. Sin embargo, la experimentación reveló que esta relación no era siempre directa. Se identificaron configuraciones angulares donde el voltaje no objetivo se atenuaba de manera casi perfecta, pero esto ocurría a niveles de potencia que llevaban al detector objetivo a la saturación, superando el límite de seguridad de 5V y haciendo que el estado no fuera válido.

De forma inversa, al encontrar configuraciones que mantenían el voltaje objetivo en un nivel alto pero seguro, se observaba con frecuencia que el voltaje en el canal no objetivo también aumentaba, perdiéndose la atenuación deseada. Esto demostró que una estrategia basada únicamente en el valor absoluto de un solo voltaje era insuficiente. Se requería un criterio de optimización que evaluara la relación entre ambos canales independientemente de los niveles de potencia absolutos. Para abordar este objetivo dual y resolver las limitaciones observadas, se introdujo una métrica de rendimiento normalizada como la función objetivo principal para el algoritmo de automatización.

Esta métrica se usó para abordar el objetivo dual durante la optimización:

$$\text{Métrica} = \frac{V_{\text{objetivo}} - V_{\text{no objetivo}}}{V_{\text{objetivo}} + V_{\text{no objetivo}}} \tag{3.2}$$

Donde V_{objetivo} es el voltaje de la polarización que se desea maximizar y $V_{\text{no objetivo}}$ es el voltaje de la polarización que se desea minimizar. El uso de esta fórmula ofrece una representación normalizada del rendimiento del sistema, con un valor ideal que tiende a 1, lo cual indica una alta potencia en el canal objetivo y una muy baja en el canal no objetivo. Con la implementación de esta métrica, todo el proceso de optimización, incluyendo tanto el barrido grueso como el ajuste fino, fue modificado para que sus decisiones se basaran en la maximización de esta métrica, en lugar de simplemente maximizar un único voltaje. Esto aseguró que el algoritmo buscara activamente la configuración del servomotor que ofreciera el mejor contraste posible entre las dos salidas de polarización.

Cabe mencionar que, mientras se realizaban las pruebas de los códigos, se confirmó que la limpieza rigurosa de los conectores de fibra óptica constituía un requisito fundamental para una transmisión de señal estable. Del mismo modo, se estableció la necesidad de mantener la fibra óptica en un estado estático, libre de perturbaciones mecánicas cuando se esté realizando la automatización, dado que se comprobó que cualquier movimiento o

tensión sobre la fibra introducía variaciones incontroladas en el estado de polarización, afectando directamente la validez y el rendimiento del algoritmo de automatización.

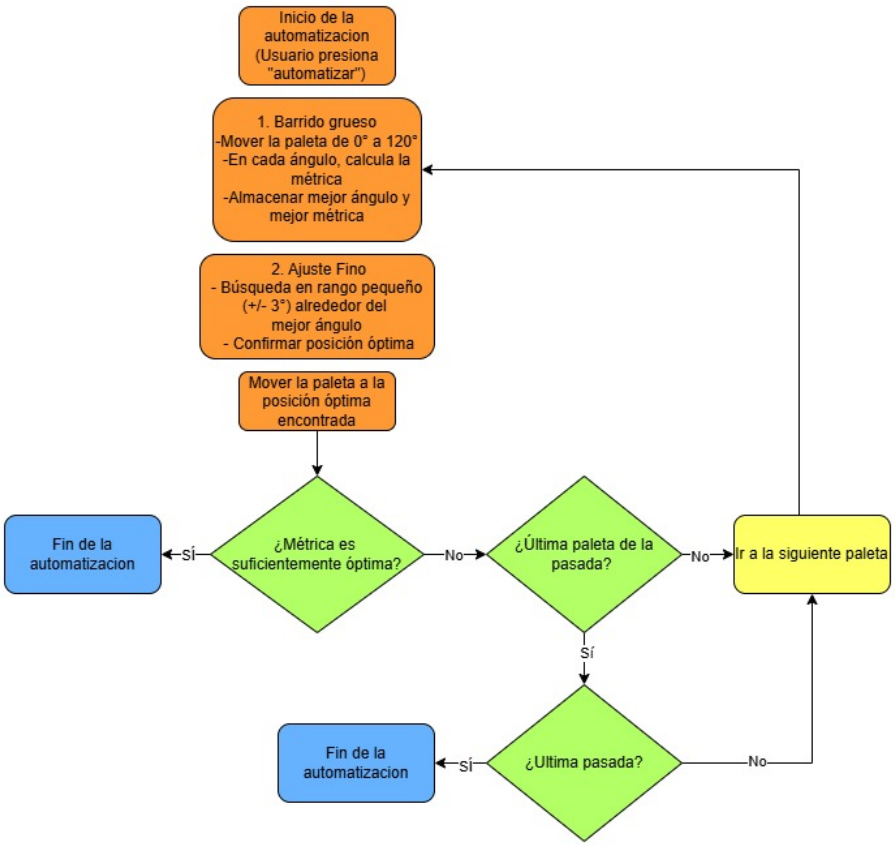


Fig. 3.5: Diagrama de flujo automatización. Fuente: Autoridad Propia.

En la imagen 3.5 se observa el diagrama de flujo del algoritmo de automatización implementado. El proceso, iniciado por el usuario, se basa en una estrategia de optimización por ángulos, organizada en un bucle que recorre secuencialmente cada uno de los tres servomotores del controlador a lo largo de un número predefinido de pasadas. Para cada servomotor, el algoritmo ejecuta un método de dos etapas: primero, un “barrido grueso” explora el rango angular completo para identificar la región de máximo rendimiento según la métrica objetivo; a continuación, un “ajuste fino” realiza una búsqueda de alta precisión en un intervalo reducido alrededor de ese

punto para determinar y establecer la posición óptima final.

El proceso de automatización cuenta con tres condiciones de término que garantizan tanto la eficiencia como el control del usuario. Las dos primeras son automáticas: la primera es la detención por éxito, donde el algoritmo concluye de forma anticipada si la métrica de rendimiento alcanza el umbral óptimo, es decir, un valor mayor o igual a 0.95, tras la optimización de cualquier servomotor. La segunda es la finalización por completo, en la que el ciclo termina de forma natural tras ejecutar el número total de pasadas programadas si no se encontró antes una solución óptima. Finalmente, la tercera y crucial condición es la interrupción manual, que permite al usuario detener el proceso en cualquier momento a través de la interfaz, otorgando flexibilidad y control total sobre la operación.

5. **Desarrollo del Gráfico de Visualización en Tiempo Real.**

   Para poder observar el comportamiento del algoritmo durante su operación, se desarrolló una gráfica en tiempo real. El objetivo principal de esta funcionalidad fue observar la evolución de los voltajes medidos a lo largo del tiempo. Esto permite un análisis directo de la respuesta del sistema a los ajustes de los servomotores, la velocidad de convergencia del algoritmo hacia una solución óptima y la estabilidad del sistema una vez que ha alcanzado dicho estado.

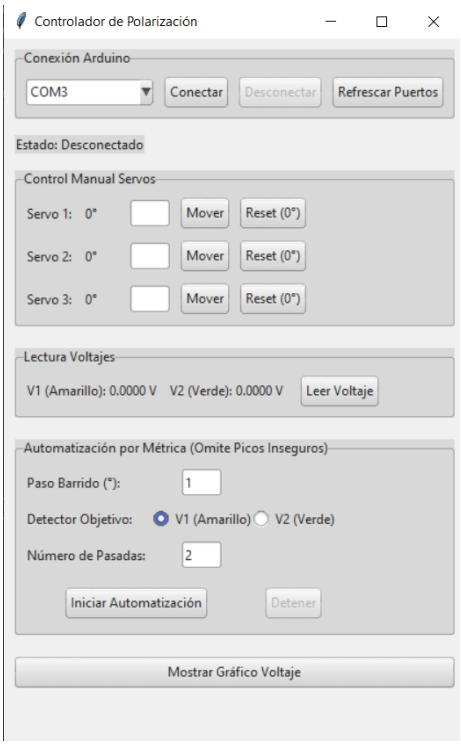


Fig. 3.6: Interfaz final. Fuente: Elaboración propia.

La implementación se llevó a cabo dentro del código principal en Python, utilizando una biblioteca de gráficos científicos para la generación de los diagramas. Para integrar esta visualización en la aplicación, se incorporó un botón denominado “Mostrar Gráfico Voltaje” en la Interfaz Gráfica de Usuario (GUI), como se aprecia en la vista general de la Figura 3.6. Al presionar este botón, se despliega una nueva ventana independiente dedicada exclusivamente a mostrar el gráfico en tiempo real.

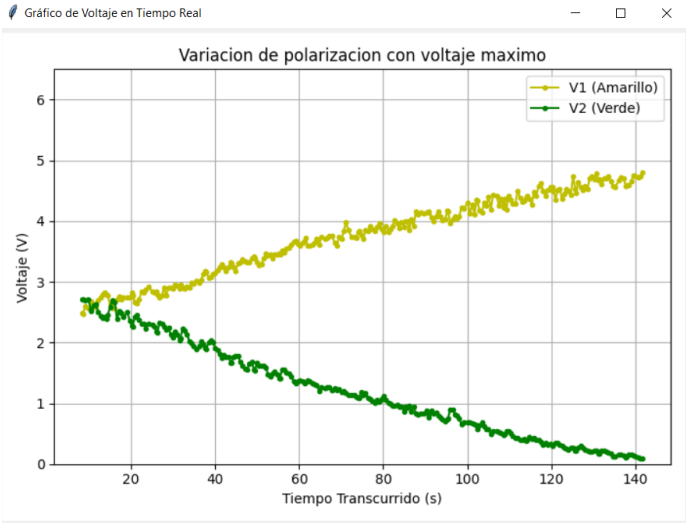


Fig. 3.7: Gráfico de los voltajes en tiempo real. Fuente: Elaboración propia.

La Figura 3.7 presenta un ejemplo de este gráfico, el cual representa la evolución temporal de los voltajes en ambas salidas de polarización. Para garantizar un funcionamiento fluido y estable, especialmente durante ejecuciones prolongadas, se implementaron dos optimizaciones clave. Primero, se utilizó un sistema de almacenamiento temporal de tamaño fijo (búfer) que registra las mediciones más recientes y descarta las más antiguas, evitando así un consumo excesivo de memoria. Segundo, para asegurar tanto la integridad de los datos como la capacidad de respuesta de la interfaz, se estableció una frecuencia de actualización del gráfico de 150 milisegundos. Este intervalo demostró ser un balance óptimo entre un monitoreo efectivo y la prevención de interferencias con la interacción del usuario. El análisis detallado y la interpretación de los resultados específicos obtenidos mediante esta herramienta de visualización se abordarán en el capítulo siguiente.

Capítulo 4

Resultados

4.1. Configuración Experimental

A continuación, se presentan los resultados obtenidos en el laboratorio. El procedimiento consistió en variar la potencia de salida de la fuente láser para evaluar el comportamiento y la eficacia del algoritmo de optimización bajo diferentes niveles de potencia óptica de entrada.

Una consideración fundamental en los experimentos fue el límite de voltaje de 5V en la entrada del microcontrolador Arduino. Para proteger el dispositivo y evitar la saturación de su conversor analógico-digital, fue necesario identificar una potencia de operación óptima. Mediante un proceso iterativo, se estableció un rango de potencias que permitía alcanzar voltajes elevados, favoreciendo así una métrica de rendimiento ideal, sin riesgo de saturación o daño al hardware. Con base en este análisis, se definieron tres escenarios de prueba:

- **Potencia Baja (1 dBm):** Para evaluar el rendimiento del sistema con una señal de entrada débil.
- **Potencia Óptima (3 dBm):** Situada dentro del rango de operación ideal.
- **Potencia Alta (7.5 dBm):** Configurada intencionalmente para llevar el sistema a la saturación y evaluar su comportamiento en condiciones

extremas.

El principio de funcionamiento del controlador mecánico se basa en inducir birrefringencia en la fibra óptica mediante la aplicación de un estrés controlado. Al presionar la fibra con una paleta, se crean localmente un eje lento y un eje rápido, lo que introduce un desfase de fase entre las componentes ortogonales de la luz. En efecto, cada paleta se comporta como una placa de onda artificial, donde la magnitud del retardo y la orientación de los ejes se controlan mediante la presión y el ángulo de la paleta. Al combinar las 3 paletas, es posible generar cualquier estado de polarización requerido.

4.2. Análisis del comportamiento en tiempo real

Como se mencionó anteriormente, estos gráficos sirven para observar la dinámica de la convergencia del algoritmo.

4.2.1. Caso de Operación Óptima (3 dBm)

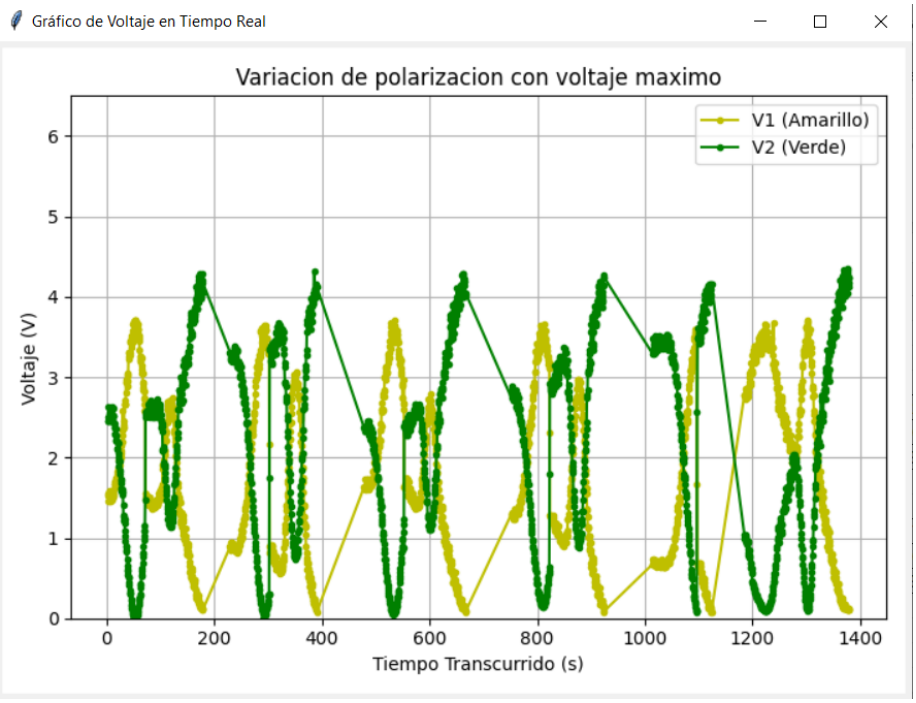


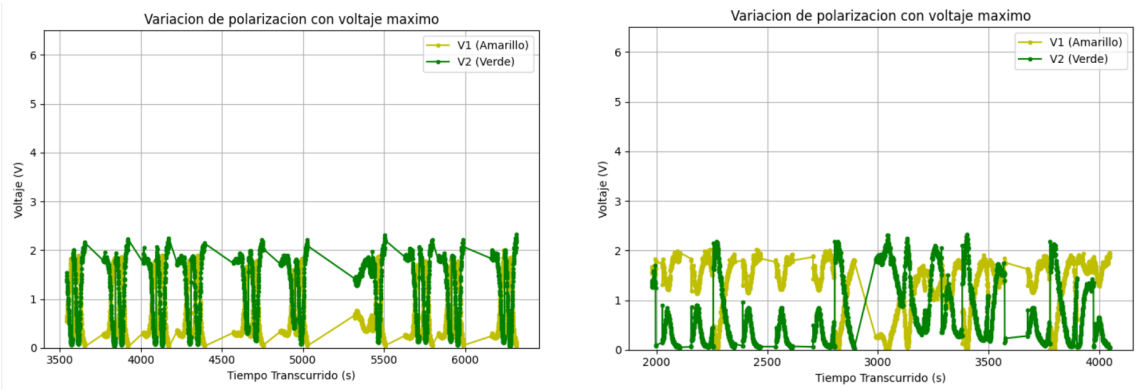
Fig. 4.1: Gráfico en tiempo real con potencia de 3 dBm maximizando la polarización horizontal. Fuente: Elaboración propia.

Se comienza con el análisis en el rango de potencia óptimo. La Figura 4.1 presenta el cambio temporal de los voltajes de ambas salidas de polarización durante la ejecución del algoritmo con una potencia de 3 dBm.

Al presionar el botón de automatización, el sistema inicia el proceso de búsqueda, lo que se observa en las variaciones constantes de las señales de voltaje correspondientes a la polarización horizontal (verde) y la polarización vertical (amarilla). Estas fluctuaciones se deben al barrido angular que realiza cada paleta mientras el algoritmo explora diferentes estados de polarización en busca de la mejor métrica. Se aprecian momentos en los que la señal de la polarización horizontal aumenta, mientras que la de la polarización vertical se atenúa, indicando la identificación de un estado favorable. Las secciones donde las líneas se vuelven estables corresponden a los momentos en que el algoritmo

ha finalizado su búsqueda, deteniendo el movimiento de las paletas al haber encontrado una solución.

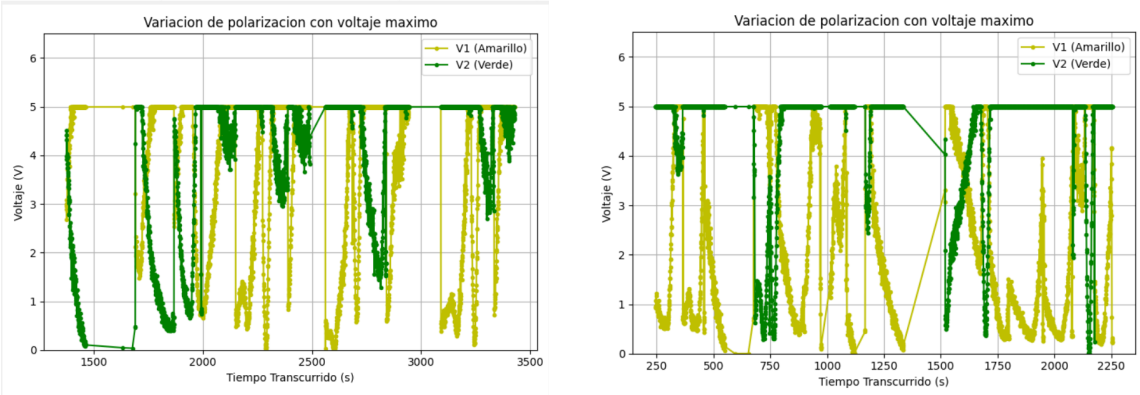
4.2.2. Casos de Baja Potencia y Saturación



(a) Maximizando Pol. horizontal. (b) Maximizando Pol. vertical.

Fig. 4.2: Gráficos en tiempo real con potencia de 1 dBm. Fuente: Elaboración propia.

La eficacia del algoritmo bajo condiciones de señal débil se examinó utilizando una potencia de entrada de 1 dBm. La Figura 4.2 ilustra el comportamiento dinámico del sistema en este escenario. De manera similar al caso de operación óptima, el sistema atraviesa una fase de búsqueda, caracterizada por las oscilaciones de los voltajes, hasta alcanzar un estado final estable. Es evidente en el gráfico que el algoritmo opera de manera efectiva para ambos objetivos de polarización (horizontal y vertical), ya que el voltaje de la polarización seleccionada es consistentemente maximizado, mientras que el de la polarización ortogonal es suprimido de forma sistemática. La disminución general en la amplitud de las señales es una consecuencia previsible de la reducida potencia de entrada y no afecta la lógica de convergencia. Por lo tanto, se confirma la robustez y la correcta implementación del algoritmo.



(a) Maximizando Pol. vertical. (b) Maximizando Pol. horizontal.

Fig. 4.3: Gráficos en tiempo real con potencia de 7.5 dBm, mostrando saturación. Fuente: Elaboración propia.

En contraste, la Figura 4.3 evidencia el fenómeno de saturación al operar con 7.5 dBm. Se observa cómo el voltaje de la polarización objetivo alcanza rápidamente el límite de 5V del conversor del Arduino y se mantiene constante en ese valor (efecto de “clipping”). Esto indica que la potencia de entrada es demasiado alta para ser medida con precisión por el sistema.

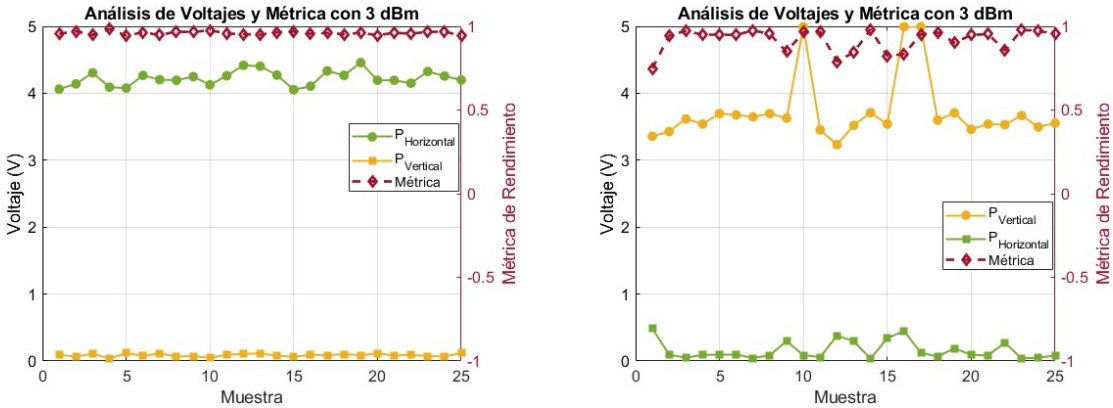
4.3. Evaluación del Rendimiento del algoritmo

Para complementar el análisis dinámico, se evaluó el rendimiento cuantitativo del algoritmo. Se graficaron los valores finales de voltaje y la métrica de rendimiento obtenidos tras detenerse el algoritmo en múltiples ejecuciones. Esta representación permite una evaluación global y comparativa de la eficacia del controlador en las distintas condiciones de potencia.

4.3.1. Rendimiento en Rango Óptimo y Bajo (1 dBm y 3 dBm)

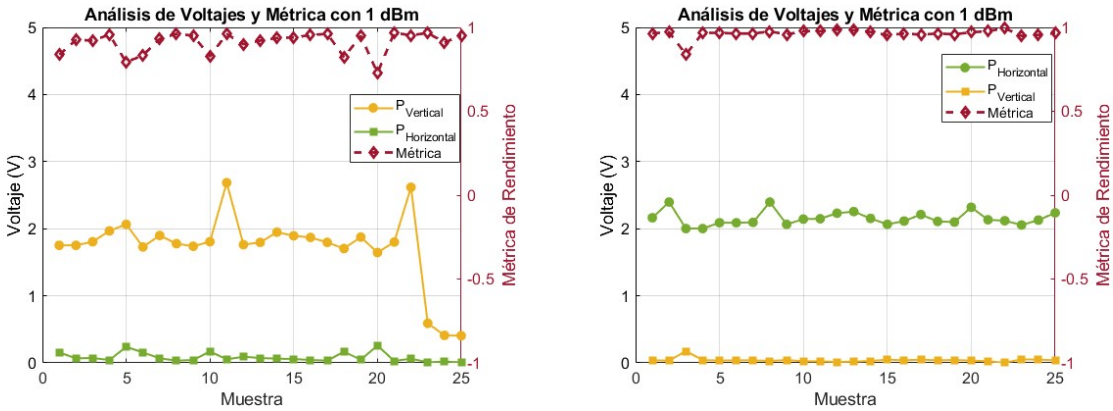
En los escenarios de operación segura, el algoritmo demostró un rendimiento consistente y robusto. Las Figuras 4.4 y 4.5 muestran los resultados para

potencias de 3 dBm y 1 dBm, respectivamente.



(a) Maximizando Pol. Horizontal. (b) Maximizando Pol. Vertical.

Fig. 4.4: Resultados de la automatización con una potencia de 3 dBm.



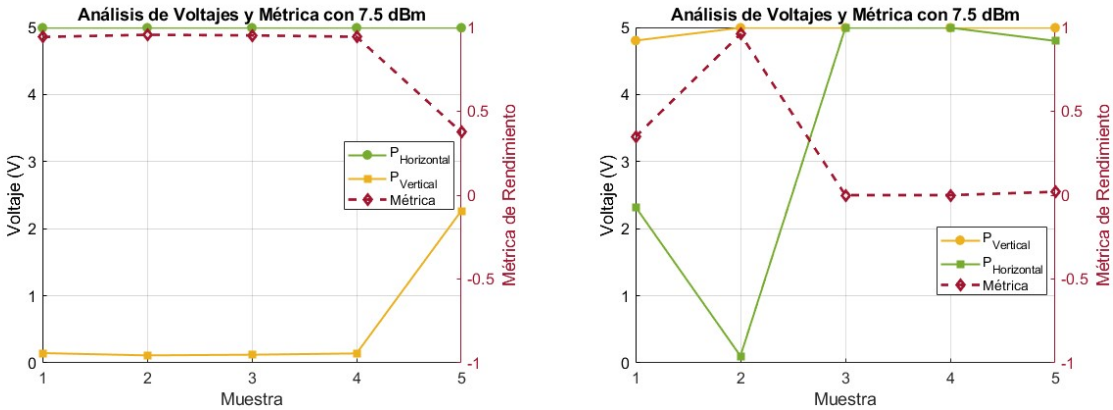
(a) Maximizando Pol. Vertical. (b) Maximizando Pol. Horizontal.

Fig. 4.5: Resultados de la automatización con una potencia de 1 dBm.

En todos estos casos, el sistema converge a un estado estable donde el voltaje objetivo se mantiene en un valor alto y constante (cercano a 4.8V), mientras que el voltaje no objetivo es atenuado a un nivel muy bajo (cercano a 0V). Como consecuencia, la métrica de rendimiento (línea roja discontinua) se estabiliza en un valor cercano al ideal de 1, demostrando la eficacia del sistema. Aunque la amplitud absoluta de los voltajes es menor en el caso de 1 dBm, la capacidad del algoritmo para alcanzar una métrica óptima no se ve comprometida, lo que valida su robustez.

4.3.2. Rendimiento en Condiciones de Saturación (7.5 dBm)

Para evaluar el comportamiento del algoritmo fuera de su rango óptimo, se utilizó una potencia de 7.5 dBm. Los resultados de esta prueba se muestran en la Figura 4.6.



(a) Maximizando Pol. Horizontal. (b) Maximizando Pol. Vertical.

Fig. 4.6: Resultados de la automatización con una potencia de 7.5 dBm.

En estos gráficos se observa claramente el efecto de la saturación en el rendimiento final. El voltaje objetivo alcanza el límite de 5V del conversor. Como se estableció en el diseño, el algoritmo está programado para omitir ángulos que produzcan esta lectura. Si bien este mecanismo de seguridad protege el hardware, impone una consecuencia: es posible que los estados de polarización que producen el mejor contraste (y por tanto la mejor métrica) ocurran precisamente en estos ángulos ahora inaccesibles. Al ser ignorados, el algoritmo converge en la mejor solución posible dentro de los límites seguros, lo que explica por qué la métrica final alcanzada es notablemente inferior a la obtenida en los casos de 1 dBm y 3 dBm. Esta prueba valida el correcto funcionamiento de las medidas de protección y caracteriza las limitaciones operativas del sistema bajo condiciones de alta potencia.

4.4. Validar experimentalmente el sistema diseñado

Tabla 4.1: Promedios y desviacion estandar de las metricas.

Experimento	Promedio	Desviación estandar
3 dBm Horizontal	0.9597	0.0100
3 dBm Vertical	0.9192	0.0674
1 dBm Horizontal	0.9632	0.0281
1 dBm Vertical	0.9105	0.0648
7.5 dBm Horizontal	0.8342	0.2563
7.5 dBm Vertical	0.2655	0.4154

La Tabla 4.1 presenta el rendimiento del algoritmo, mostrando el promedio y la desviación estándar de la métrica para 25 muestras en cada una de las condiciones de potencia evaluadas, menos para las muestras de 7.5 dBm, que por seguridad del Arduino solamente se usaron 10 muestras en total. El análisis de estos datos permite caracterizar la eficacia, repetibilidad y limitaciones del algoritmo de control.

En los escenarios de 1 dBm y 3 dBm, el algoritmo demuestra un rendimiento excelente y altamente consistente. Los valores promedio de la métrica de rendimiento son muy elevados, superando el 0.91 en todos los casos y alcanzando un máximo de 0.9632. Una métrica tan cercana al valor ideal de 1 confirma que el algoritmo es altamente eficaz en su objetivo dual. Esta eficacia se complementa con una alta repetibilidad, evidenciada por la baja desviación estándar (entre 0.0100 y 0.0674). Un valor tan reducido indica que el algoritmo no solo funciona bien, sino que converge a un resultado de alta calidad de manera fiable en la gran mayoría de las ejecuciones. Adicionalmente, se observa una pequeña diferencia de rendimiento al maximizar la polarización horizontal frente a la vertical, lo cual podría atribuirse a ligeras asimetrías en la respuesta de los fotodetectores o en la eficiencia del divisor de haz polarizante (PBS), fenómenos esperables en un sistema experimental real.

Por otro lado, los resultados obtenidos con una potencia de 7.5 dBm son cruciales, ya que validan el comportamiento del sistema bajo condiciones extremas y el funcionamiento de sus mecanismos de seguridad. Como se esperaba, se observa una clara degradación en el rendimiento, la métrica promedio baja a 0.8342 al maximizar la polarización horizontal, y se desploma a 0.2655 en el caso de la vertical. Este comportamiento confirma la hipótesis planteada anteriormente, según la cual el algoritmo, al ignorar los ángulos que saturan el conversor A/D para proteger el dispositivo, se ve forzado a descartar estados de polarización potencialmente óptimos y converger en una solución subóptima pero segura. El dato más revelador es el drástico aumento de la desviación estándar, que alcanza 0.4154. Una dispersión tan alta demuestra que, en el régimen de saturación, el rendimiento no solo disminuye, sino que se vuelve impredecible e inestable, pues el algoritmo tiene dificultades para encontrar una solución consistente sin poder acceder a la información de los picos de potencia reales.

4.5. Tiempo promedio de ejecución.

Finalmente, se realizó una estimación del tiempo de convergencia del algoritmo en los rangos de operación no saturados. El tiempo total de búsqueda depende principalmente del barrido grueso y del ajuste fino realizado por los tres servomotores.

El tiempo del barrido grueso por cada servomotor se calculó de la siguiente manera: el barrido abarca 121 puntos angulares (de 0° a 120°), donde cada punto requiere un tiempo de medición de voltaje de 0.15 segundos y un tiempo de posicionamiento del servomotor de 0.35 segundos. El tiempo del barrido grueso por cada servomotor se calculó en aproximadamente 60.5 seg. Este valor resulta de multiplicar los 121 puntos que abarca el barrido por el tiempo individual requerido para cada punto. A su vez, el tiempo para cada uno de estos puntos es de 0.5 seg, ya que se compone de la suma del tiempo de medición de voltaje 0.15 seg y el de posicionamiento del servomotor 0.35 seg.

A esto se suma el tiempo consumido por la fase de ajuste fino. Basado en la observación en el laboratorio, esta etapa se activa un promedio de 5 a 10 veces por servomotor, realizando en cada activación un barrido de sobre 7 ángulos.

Por lo tanto, una sola activación completa del ajuste fino toma un total de $3{,}5$ seg, ya que se multiplican los 7 ángulos del barrido por los $0{,}5$ seg que requiere cada ángulo. En consecuencia, el rango de tiempo total para el límite inferior es de $17{,}5$ seg, que corresponde al escenario donde el ajuste fino se activa 5 veces, mientras que el límite superior es de 35 seg para un caso con 10 activaciones por servomotor.

De este modo, el tiempo total para que un solo servomotor complete su ciclo se encuentra en un rango definido. El caso más rápido toma 78 seg, resultado de la suma del barrido grueso ($60{,}5$ seg) y el mínimo del ajuste fino ($17{,}5$ seg). En el extremo opuesto, el caso más lento requiere $95{,}5$ seg, ya que al barrido grueso se le suma el tiempo máximo del ajuste fino (35 seg).

Al usar tres servomotores en secuencia, el tiempo máximo teórico alcanzaría los 286.5 seg, que es el triple del tiempo máximo de un solo servomotor. Sin embargo, la evidencia experimental muestra que en más del 90 % de los casos, el sistema converge antes de que el tercer servomotor finalice su ciclo.

Capítulo 5

Conclusiones

5.1. Sumario

A lo largo de este trabajo, se abordó el diseño, la implementación y la validación de un sistema de bajo costo para el control automatizado del estado de polarización (SOP) en fibra óptica. El objetivo fue desarrollar una alternativa asequible y eficiente frente a las soluciones comerciales de alto precio y los ajustes manuales, que son lentos e imprecisos. La solución se materializó en un controlador de polarización mecánico, fabricado mediante impresión 3D, y operado por tres servomotores manejados por un Arduino. La optimización se guió por un lazo de control retroalimentado por dos fotodetectores, que miden la potencia de las salidas de polarización ortogonales del sistema.

El núcleo del sistema es un algoritmo de optimización desarrollado en Python, el cual evolucionó hasta una estrategia robusta de dos etapas: un barrido grueso inicial para explorar todo el rango angular de operación, seguido de un ajuste fino de alta precisión para refinar la posición óptima. Se determinó que el criterio de optimización más eficaz no era la maximización de un único voltaje, sino una métrica de rendimiento normalizada que evalúa el contraste entre ambas salidas de polarización, asegurando simultáneamente la maximización de la potencia deseada y la atenuación de la ortogonal. Los resultados experimentales,

evaluados en condiciones de potencia baja (1 dBm), óptima (3 dBm) y alta (7.5 dBm), demostraron que el sistema funciona con excelente eficacia y repetibilidad en los rangos de operación no saturados, alcanzando valores de métrica consistentemente cercanos al ideal de 1. Sin embargo, también se caracterizó una limitación clave: a potencias elevadas, el sistema se ve afectado por la saturación del conversor analógico-digital del Arduino, lo que degrada el rendimiento de la métrica final.

5.2. Conclusión

Al finalizar esta investigación, se concluye que es totalmente viable desarrollar e implementar un sistema automatizado, de bajo costo y código abierto, para el control efectivo del estado de polarización en fibra óptica. El prototipo no solo demostró ser funcional, sino también preciso y estable dentro de su rango de operación óptimo, posicionándose como una alternativa práctica y accesible para entornos de investigación y desarrollo académico donde los equipos comerciales son prohibitivos. Con esto, se da una respuesta satisfactoria al problema planteado sobre la inestabilidad de la polarización y la falta de herramientas asequibles para su gestión, eliminando la necesidad de ajustes manuales lentos e imprecisos.

Con esta perspectiva, podemos afirmar que los objetivos planteados al inicio de esta memoria han sido cumplidos satisfactoriamente. Se logró construir una base sólida mediante el desarrollo de un código en Python y una interfaz gráfica que permitió, en primera instancia, el control manual del sistema. Sobre esta base se diseñó e implementó un algoritmo de control automático capaz de optimizar el estado de polarización de manera eficiente. Dicho algoritmo se perfeccionó para operar en función de una métrica de rendimiento que asegura el máximo contraste entre las salidas ortogonales. Finalmente, el desarrollo se complementó con una herramienta de visualización en tiempo real, la cual resultó fundamental no solo para observar el comportamiento dinámico del sistema, sino también para validar la correcta convergencia y la eficacia del método de optimización propuesto.

La investigación arrojó un hallazgo importante: para una optimización robusta, no basta con maximizar la potencia en un canal, sino que es fundamental emplear una métrica que considere la relación entre ambas salidas de polarización. La principal contribución de este trabajo es, por tanto, el diseño completo y validado de este sistema, que sirve como un modelo funcional y un punto de partida para futuros desarrollos en el campo de la instrumentación óptica de bajo costo.

5.3. Trabajo a Futuro

Aunque los objetivos se cumplieron, esta investigación abre nuevas e interesantes interrogantes que delinean el camino para futuros trabajos. La progresión más natural sería abordar las limitaciones inherentes al hardware del prototipo. La más crítica es la saturación del conversor del Arduino; su solución, mediante la implementación de una etapa de acondicionamiento de señal, ampliaría el rango dinámico del sistema y permitiría su uso en aplicaciones de mayor potencia. En esta misma línea, para abordar aplicaciones que requieran una compensación más rápida, se podría explorar la sustitución de los servomotores por actuadores de mayor velocidad, como los fiber squeezers piezoeléctricos, mejorando así la velocidad de respuesta del sistema.

Desde la perspectiva del código y la lógica de control, existe un considerable margen para el refinamiento. El algoritmo actual podría evolucionar hacia un método de búsqueda más inteligente, implementando técnicas de optimización avanzadas, como el ascenso de gradiente, o incluso modelos de aprendizaje automático que podrían converger al punto óptimo de forma más rápida y directa. Dicho algoritmo podría, además, ser adaptativo, ajustando dinámicamente sus parámetros de búsqueda en función del estado de la red para mejorar la eficiencia. Finalmente, el desarrollo de un refinamiento del diseño mecánico del controlador no solo mejoraría la precisión del sistema, sino que facilitaría enormemente su replicación y uso por parte de otros investigadores en el campo.

Bibliografía

[1] F. Q. Kareem, S. R. M. Zeebaree, H. I. Dino, M. A. M. Sadeeq, Z. N. Rashid, D. A. Hasan, and K. H. Sharif, “A Survey of Optical Fiber Communications: Challenges and Processing Time Influences,” *Asian Journal of Research in Computer Science*, vol. 7, no. 4, pp. 48–58, April 2021.

[2] C. M. Carrillo-Delgado, J. D. Filoteo-Razo, J. M. Sierra-Hernandez, J. C. Hernandez-Garcia, D. Jauregui-Vazquez, Y. Lopez-Dieguez, J. M. Estudillo-Ayala, L. A. Herrera-Piad, and R. Rojas-Laguna, “Graphic User Interface for Modeling States of Polarization in Fiber Optics,” *Computación y Sistemas*, vol. 23, no. 1, pp. 63–70, 2019.

[3] L. H. Trinh, N. V. Truong, and F. Ferrero, “Low Cost Circularly Polarized Antenna for IoT Space Applications,” *Electronics*, vol. 9, no. 10, p. 1564, September 2020.

[4] Thorlabs, “Fiber Polarization Beam Combiner 1550nm, SM Output, FC/PC,” Product Datasheet, May 2015, item # PBC1550SM-FC, Rev D. Accessed: [2025]. [Online]. Available: https://www.thorlabs.com

[5] Thorlabs Inc., *MPC220 and MPC320 Motorized Fiber Polarization Controllers: User Guide*, Thorlabs, Newton, NJ, USA, 2024, rev. C, May 2024.

[6] Arduino S.r.l., *Arduino® UNO R3: User Manual*, Arduino, Via Andrea Appiani 25, 20900 Monza, Italy, 2025, modified: 19/06/2025, SKU: A000066.

[7] Thorlabs Inc., *PDA05CF2 InGaAs Amplified Fixed Gain Detector: User Guide*, Thorlabs, Newton, NJ, USA, 2018, rev. B, January 3, 2018.

[8] G. Pu, L. Zhang, W. Hu, and L. Yi, “Automatic mode-locking fiber lasers: progress and perspectives,” *SCIENCE CHINA Information Sciences*, vol. 63, no. 6, p. 160404, June 2020.

[9] J. Jeong, “Polarization Control and Measurement for Optical Fibers,” Newport Corporation, Irvine, CA, Application Note 20, 2006.

[10] H. Zhang, L. Liu, R. B. Jander, and A. N. Pilipetskii, “Methods and apparatus for polarization control,” Patent EP 1 574 893 A1, September, 2005, european Patent Application.

[11] T. Nakamura, T. Nomura, M. Endo, A. Sakaguchi, H. Ruofan, T. Kashiwazaki, T. Umeki, K. Takase, W. Asavanant, J.-i. Yoshikawa, and A. Furusawa, “Long-term stability of squeezed light in a fiber-based system using automated alignment,” *Review of Scientific Instruments*, vol. 95, no. 9, p. 093004, September 2024.

[12] A. Zahidi, S. Amrane, N. Azami, and N. Nasser, “Effect of fiber and solenoid variation parameters on the elements of a corrector PID for electromagnetic fiber squeezer based polarization controller,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 3, pp. 2441–2451, June 2020.

[13] D.-J. Wang, X. Zhang, J. Liu, D.-D. Jiao, X. Deng, J. Gao, Q. Zang, D. Wang, T. Liu, R.-F. Dong, and S.-G. Zhang, “Novel Polarization Control Approach to Long-Term Fiber-Optic Frequency Transfer,” *Chinese Physics Letters*, vol. 37, no. 9, p. 094201, September 2020.

[14] S. Kamoldilok, J. Kadtajan, A. Kanajirayupat, and K. Srinuanjan, “Manipulating Polarization via Optical Fiber Twisting for Optical Switch Applications,” *Journal of Physics: Conference Series*, vol. 2934, p. 012024, 2025.

[15] V. Mayboroda, N. Rudavin, P. Kupriyanov, O. Fat'yanov, and R. Shakhovoy, "Analytical solution to the problem of polarization drift compensation in all-fiber QKD system," March 2024.

[16] A. Zahidi, S. Amrane, N. Azami, and N. Nasser, "Identification of solenoid parameters based on fiber squeezer and neural network for Smart control," in *2020 International Conference on Intelligent Systems and Computer Vision (ISCV)*, 2020, pp. 1–6.

[17] T. Baumeister, S. L. Brunton, and J. N. Kutz, "Deep Learning and Model Predictive Control for Self-Tuning Mode-Locked Lasers," November 2017.

[18] N. Braband, A. Mansouri, R. Fazili, S. Czischek, and J. Lundeen, "Fast and Automated Optical Polarization Compensation of Fiber Unitaries," November 2024.

[19] Y. Tan, J. Wang, J. Wu, and Z. He, "Real-time polarization compensation method in quantum communication based on channel Muller parameters detection," *Communications Engineering*, vol. 3, p. 57, March 2024.

[20] P.-H. Chang, M. Fu, P. Yan, A. Samanta, M. B. On, Y. Zhu, S. J. B. Yoo, and S. Palermo, "A Monolithic Polarization Controller and Tracking Loop for Optical Interconnect Demonstrated on a 90 nm Silicon CMOS-Photonic Platform," *Journal of Lightwave Technology*, vol. 41, no. 12, pp. 3832–3841, June 2023.

[21] X. Wang, Y. Jia, X. Guo, J. Liu, S. Wang, W. Liu, F. Sun, J. Zou, and Y. Li, "Silicon photonics integrated dynamic polarization controller," *Chinese Optics Letters*, vol. 20, no. 4, p. 041301, April 2022.

[22] S. Zhao, S. Wang, B. Gu, C. Yan, S. Wang, G. Deng, and S. Zhou, "Controllable mode emission from all-fiber laser based on polarization rotation technology and its application in bending sensing," *Optics Communications*, vol. 528, p. 129007, 2023.

[23] D. G. Winters, M. S. Kirchner, S. J. Backus, and H. C. Kapteyn, "Electronic initiation and optimization of nonlinear polarization evolution mode-locking in

a fiber laser," *Optics Express*, vol. 25, no. 26, pp. 33 216–33 225, December 2017.

[24] T. Nadolny, "Automatic Compensation of Polarization Drifts in an Optical Fiber," München, Germany, July 2018, faculty of Physics.

[25] J. Wang, X. Tan, P. Qi, C. Wu, L. Huang, X. Xu, Z. Huang, L. Zhu, Y. Zhang, X. Lin, J. Zang, and K. Kuroda, "Linear polarization holography," *Opto-Electronic Science*, vol. 1, no. 2, p. 210009, feb 2022.

[26] B. Y. Toh, R. Cahill, and V. F. Fusco, "Understanding and measuring circular polarization," *IEEE Transactions on Education*, vol. 46, no. 3, pp. 313–318, aug 2003.

[27] Y. Lei, G. Shayeganrad, H. Wang, M. Sakakura, Y. Yu, L. Wang, D. Kliukin, L. Skuja, Y. Svirko, and P. G. Kazansky, "Efficient ultrafast laser writing with elliptical polarization," *Light: Science & Applications*, vol. 12, no. 1, p. 74, mar 2023.

[28] D. Tentori, C. Ayala Díaz, and F. Treviño Martínez, "Evaluación de la Birrefringencia de una Fibra Óptica Monomodo Usando el Método de Barrido Espectral, Estudio Comparativo de Dos Metodologías," in *Simposio de Metrología 2008*, Santiago de Querétaro, México, oct 2008, paper SM2008-M222-1115.

[29] *PDA05CF2 InGaAs Amplified Fixed Gain Detector User Guide*, Thorlabs, Newton, New Jersey, jan 2018, revision B.

[30] Thorlabs GmbH, *PRO8000 Series Mainframes Operation Manual*, Thorlabs, jul 2024, version 3.8.

[31] Tektronix, Inc., *TDS2000C and TDS1000C-EDU Series Digital Storage Oscilloscopes User Manual*, Tektronix, Inc., Beaverton, OR, part Number 071-2722-03, Revision A.

[32] ——, *AFG31000 Series Arbitrary Function Generator User's Manual*, Tektronix, Inc., Beaverton, OR, nov 2020, part Number 077-1473-03.

[33] B. H. Billings and H. Hua, “Polarized light,” *AccessScience*, 2008. [Online]. Available: http://www.accessscience.com

[34] IDEAL, “MiniLite-Strip™ Optical Fiber Stripper,” Product Datasheet, catalog No. 45-352.

[35] Grandway, “GW-750 Fiber Cleaver,” Product Datasheet, disponible en: www.grandway.com.cn.

[36] Jilong, “KL-300T Fusion Splicer,” Product Specification Sheet, provided by AAATesters.com.

Appendices

Apéndice A

ANEXO: Figuras relevantes pero no tanto para el cuerpo de la tesis

Apéndice B

ANEXO: Código

```
import pandas as pd
import matplotlib.pyplot as plt
import os

def calc_stats_for_col(serie: pd.Series):

    # Retorna una lista con [Max, Min, Mean, Std, P50, P90].

    col_max   = serie.max()
    col_min   = serie.min()
    col_mean  = serie.mean()
    col_std   = serie.std()
    col_p50   = serie.quantile(0.50)
    col_p90   = serie.quantile(0.90)

    return [col_max, col_min, col_mean, col_std, col_p50, col_p90]
```

Listing B.1: Código