

UNIVERSIDAD MIGUEL HERNÁNDEZ DE ELCHE

ESCUELA POLITÉCNICA SUPERIOR DE ELCHE

GRADO EN INGENIERÍA DE TECNOLOGÍAS DE LA
TELECOMUNICACIÓN



“ANÁLISIS DE CÉLULAS
FOTOVOLTAICAS PARA
COMUNICACIONES LI-FI USANDO
TÉCNICAS DE APRENDIZAJE
AUTOMÁTICO MEDIANTE REGRESIÓN Y
FILTRADO”

TRABAJO FIN DE GRADO

Septiembre - 2024

AUTOR: Luis Carlos Mena Rojas

DIRECTOR: Pablo Corral Gonzalez

RESUMEN

La creciente demanda de capacidad y velocidad en las comunicaciones ha llevado al desarrollo de tecnologías más eficientes, como las comunicaciones mediante luz visible (VLC), que utilizan un espectro que va desde 380 nm hasta 750 nm para transmitir información. Dentro de estas tecnologías, el Li-Fi (Light-Fidelity) se destaca por usar diodos LED para la transmisión de datos, ofreciendo un ancho de banda significativo y altas velocidades en distancias cortas. Este trabajo se centra en evaluar el desempeño de las células fotovoltaicas orgánicas y comerciales en sistemas Li-Fi, utilizando técnicas de aprendizaje automático para optimizar el proceso de transmisión y recepción de datos, además de aplicar técnicas de filtrado para eliminar los picos de señal que podemos observar a altas frecuencias. Los resultados muestran que las células orgánicas, dopadas con disolventes como dimetilformamida (DMF), tetrahidrofurano (THF) y dimetilsulfóxido (DMSO), pueden ser una alternativa interesante para aplicaciones de comunicación Li-Fi.

Palabras clave: Light-Fidelity, células orgánicas, aprendizaje automático



ABSTRACT

The increasing demand for capacity and speed in communications has led to the development of more efficient technologies, such as Visible Light Communications (VLC), which use a spectrum ranging from 380 nm to 750 nm to transmit information. Among these technologies, Li-Fi (Light-Fidelity) stands out by using LED diodes for data transmission, offering significant bandwidth and high speeds over short distances. This work focuses on evaluating the performance of organic photovoltaic cells and commercial LED in Li-Fi systems, using machine learning techniques to optimize the data transmission and reception process in addition to applying filtering techniques to eliminate the frequency peaks that we can observe at high frequencies. The results show that organic cells, doped with solvents such as dimethylformamide (DMF), tetrahydrofuran (THF), and dimethyl sulfoxide (DMSO), can be an interesting alternative for Li-Fi communication applications.

Key words: Light-Fidelity, organic cells, machine learning





ÍNDICE DE FIGURAS

Figura 1. Descripción gráfica de los modelos de clasificación y de regresión.....	4
Figura 2. Compensación entre la complejidad del modelo y la precisión del entrenamiento y las pruebas.	5
Figura 3. Dispositivo Analog Discovery 2.	7
Figura 4. Esquema de conexiones Analog Discovery 2.	8
Figura 5. Esquema de conexionado.	8
Figura 6. Entorno de trabajo Waveforms.	9
Figura 7. LED comercial.	11
Figura 8. Sensibilidad espectral relativa.	12
Figura 9. Características direccionales.	12
Figura 10. Regresión lineal para 40 muestras con datos automatizados.	17
Figura 11. Regresión lineal para 40 muestras con datos introducidos manualmente.	18
Figura 12. Regresión lineal para 1000 muestras.	19
Figura 13. Comparación de grados para una regresión polinomial con 1000 muestras.	20
Figura 14. Regresión lineal.	21
Figura 15. Regresión polinomial.	22
Figura 16. Regresión polinomial.	23
Figura 17. Regresión polinomial.	24
Figura 18. Regresión polinomial.	24
Figura 19. Regresión polinomial.	25
Figura 20. Regresión polinomial.	26
Figura 21. Regresión polinomial.	26
Figura 22. Representación del cálculo del factor de calidad Q.	27
Figura 23. Relación entre el factor de calidad (Q) en lineal y la frecuencia.	31
Figura 24. Relación entre el factor de calidad (Q) en dB y la frecuencia.	32
Figura 25. Relación entre la relación señal a ruido (SNR) en lineal y la frecuencia.	32
Figura 26. Relación entre la relación señal a ruido (SNR) en dB y la frecuencia.	33
Figura 27. Relación entre la tasa de error de bit (BER) y la frecuencia.	33
Figura 28. Relación entre la tasa de error de bit (BER) y la relación señal a ruido (SNR).	34
Figura 29. Portaplacas con célula orgánica fotovoltaica.	35
Figura 30. Ejemplo de célula orgánica fotovoltaica. [8]	35
Figura 31. Células 1 y 2 de plomo.	36
Figura 32. Regresión polinomial.	37
Figura 33. Regresión polinomial.	37
Figura 34. Regresión polinomial.	38
Figura 35. Regresión polinomial.	38
Figura 36. Regresión polinomial.	39
Figura 37. Regresión polinomial.	39
Figura 38. Regresión polinomial a 1 kHz.	40
Figura 39. Regresión polinomial a 1 kHz.	40
Figura 40. Regresión polinomial a 100 kHz.	41
Figura 41. Regresión polinomial a 100 kHz.	41
Figura 42. Regresión polinomial a 1 kHz.	42
Figura 43. Regresión polinomial a 100 kHz.	42
Figura 44. Comparación de regresión polinomial a 100 kHz y 1 kHz Cell 1.	43
Figura 45. Comparación de regresión polinomial a 100 kHz y 1 kHz Cell 2.	43
Figura 46. Células THF.	44

Figura 47. Células DMSO	44
Figura 48. Células DMF	45
Figura 49. Comparación de todas las celdas de las células THF.....	45
Figura 50. Comparación de todas las celdas de las células DMSO.....	46
Figura 51. Comparación de todas las celdas de las células DMF.	46
Figura 52. Comparación de todas las celdas de las células THF.....	47
Figura 53. Comparación de todas las celdas de las células DMSO.....	47
Figura 54. Comparación de todas las celdas de las células DMF.	48
Figura 55. Comparación de las mejores células de cada tipo.....	48
Figura 56. Comparación de las mejores células de cada tipo.....	49
Figura 57. Osciloscopio WaveForms.	50
Figura 58. Señal de 500 kHz sin procesar.	51
Figura 59. Señal procesada con filtro de mediana.....	51
Figura 60. Señal procesada con filtro Gaussiano.	52
Figura 61. Señal de 500 kHz sin procesar.	53
Figura 62. Señal procesada con filtro de mediana.....	53
Figura 63. Señal procesada con filtro Gaussiano.	54
Figura 64. Señal de 500 kHz sin procesar.	54
Figura 65. Señal procesada con filtro de mediana.....	55
Figura 66. Señal procesada con filtro Gaussiano.	55



ÍNDICE DE TABLAS

Tabla 1. Ventajas de las comunicaciones Li-Fi [1].	3
Tabla 2. Archivo .csv con datos de voltaje y porcentaje.	29
Tabla 3. Voltaje Pico-Pico a 1 kHz y 0 cm.	49
Tabla 4. Voltaje Pico-Pico a 1 kHz y 10 cm.	49
Tabla 5. Valores de voltaje pico-pico para células THF, DMSO y DMF con diferentes filtros.....	56
Tabla 6. Reducción porcentual para filtro de mediana y Gaussiano para células THF, DMSO y DMF.....	56





CONTENIDO

CAPÍTULO 1. INTRODUCCIÓN Y OBJETIVOS	1
CAPÍTULO 2. ESTADO DEL ARTE	3
2.1. SISTEMAS DE COMUNICACIONES LI-FI.....	3
2.2. MACHINE LEARNING	3
CAPÍTULO 3. MATERIALES Y MÉTODOS	7
3.1. HARDWARE	7
3.1.1. ANALOG DISCOVERY	7
3.2. SOFTWARE.....	9
3.2.1. WAVEFORMS.....	9
3.2.2. GOOGLE COLABORATORY	10
CAPÍTULO 4. MEDIDAS EXPERIMENTALES	11
4.1. LED COMERCIAL	11
4.2. CÉLULAS COMERCIALES	11
4.2.1. REGRESIONES LINEALES Y POLINÓMICAS EN FUNCIÓN DE LA FRECUENCIA.....	13
4.2.2. REGRESIONES LINEALES Y POLINÓMICAS EN FUNCIÓN DE LA DISTANCIA	20
4.2.3. TIEMPOS DE SUBIDA Y BAJADA.....	22
4.2.4. CICLO DE TRABAJO	25
4.2.5. PARÁMETROS Q-SNR-BER	27
4.3. CÉLULAS ORGÁNICAS	34
4.3.1. REGRESIONES POLINÓMICAS EN FUNCIÓN DE LA FRECUENCIA Y DISTANCIA PARA CÉLULAS DE PLOMO.....	36
4.3.2. COMPARACIÓN CÉLULAS THF, DMSO, DMF	43
4.3.3. FILTRO DE MEDIANA Y GAUSSIANO APLICADO A CÉLULAS THF, DMSO Y DMF	50
CAPÍTULO 5. CONCLUSIÓN Y LINEAS FUTURAS	57
5.1. CONCLUSIÓN	57
5.2. LINEAS FUTURAS.....	57
REFERENCIAS	59

CAPÍTULO 1. INTRODUCCIÓN Y OBJETIVOS

Las comunicaciones a través de la luz visible (VLC) han emergido como una alternativa prometedora a los medios de transmisión tradicionales. Una de las tecnologías más destacadas dentro de las VLC es el Li-Fi (Light-Fidelity), introducida en 2011 por el físico alemán Harald Hass. El Li-Fi utiliza diodos LED para transmitir datos a través de la iluminación, ofreciendo un ancho de banda significativo y altas velocidades en distancias cortas.

El uso de Li-Fi presenta múltiples ventajas, como la seguridad inherente a su operación en espacios cerrados, donde las señales no pueden ser interceptadas fuera de la habitación. Además, el encendido y apagado rápido de los LEDs permite la transmisión de datos a una frecuencia que el ojo humano no puede detectar, haciendo posible la comunicación de datos sin interrupciones perceptibles.

El objetivo principal de este trabajo es explorar y evaluar el desempeño de las células fotovoltaicas orgánicas y comerciales en sistemas de comunicación Li-Fi, empleando técnicas de aprendizaje automático (Machine Learning) para optimizar el proceso de transmisión y recepción de datos mediante regresiones. Para ello se realizarán una serie de objetivos como son realizar mediciones y análisis comparativos de las células fotovoltaicas comerciales y las orgánicas, evaluando su desempeño en términos de frecuencia, distancia, tiempos de subida y bajada, y otros parámetros relevantes. También se desarrollarán modelos de regresión utilizando Python, para predecir y mejorar la eficiencia del sistema Li-Fi en diversas condiciones operativas. Se evaluarán los parámetros de SNR (Signal-to-Noise Ratio), BER (Bit Error Rate) y Q (Factor de Calidad) en diferentes escenarios, para determinar la robustez y fiabilidad del sistema. Y por último hará una comparación de los resultados obtenidos con células fotovoltaicas comerciales y orgánicas, implementando mejoras basadas en los análisis de Machine Learning para optimizar el desempeño del sistema Li-Fi mediante el uso de filtros de mediana y Gaussianos.

Este trabajo busca no solo demostrar la viabilidad de las células fotovoltaicas orgánicas en aplicaciones de comunicación, sino también contribuir al desarrollo de sistemas Li-Fi más eficientes y adaptables mediante el uso de tecnologías avanzadas de análisis de datos.



CAPÍTULO 2. ESTADO DEL ARTE

2.1. SISTEMAS DE COMUNICACIONES LI-FI

La creciente demanda en términos de capacidad y velocidad en el mundo de las comunicaciones ha creado la necesidad de utilizar sistemas de comunicación más eficientes que el limitado espectro de radiofrecuencias, dando lugar a las comunicaciones VLC (Visible light communications), que emplean la luz visible para la comunicación, ocupando un espectro de 380 nm a 750 nm correspondiente en frecuencia de 430 THz a 790 THz. Estos sistemas están diseñados para que tanto el transmisor como el receptor se encuentren en una misma habitación, por lo que están mucho más protegidos ya que los receptores que se encuentren fuera de la habitación no tendrán acceso a la información, haciéndolos mucho más seguros.

Dentro de los diferentes tipos de comunicaciones VLC, encontramos los sistemas de comunicaciones Li-Fi (Light-Fidelity). Li-Fi proporciona transmisión de datos a través de iluminación mediante un diodo LED, obteniendo un gran ancho de banda y una gran velocidad a distancias cortas.

Un LED puede encenderse y apagarse a una frecuencia superior de la que el ojo humano es capaz de captar, por lo que podemos transmitir información sin necesidad de que una persona pueda percibir que se está transmitiendo, simulando que está permanentemente encendida. El proceso de encendido y apagado se puede ver como la transmisión de datos binarios, entendiendo un '1' como un LED encendido y un '0' como un LED apagado, la luz finalmente pasa por un fotodetector, convirtiéndola a los datos originales [1].

Finalmente podemos resumir todas las ventajas que nos aporta el uso de sistemas de comunicaciones Li-Fi en la siguiente tabla:

Luz	Los LEDs producen más luz por Watt que las bombillas incandescentes
Tiempo ON-OFF	Tienen un tiempo de encendido-apagado muy rápido
Toxicidad	En comparación con las lámparas fluorescentes, los LEDs no contienen mercurio
Banda libre	Li-Fi hace uso de una banda libre que no necesita ninguna licencia
Altas velocidades	Ofrece velocidades teóricas del orden de Gigabits por segundo
Aerolíneas	Li-Fi se puede usar de manera segura en aerolíneas sin afectar a ninguna señal
Aplicaciones médicas	Se puede integrar en dispositivos médicos y en hospitales sin tener que involucrar ondas de radio
Utilidad subacuática	Wi-Fi no funciona bajo el agua, pero Li-Fi lo hace y puede ser usado para expediciones subacuáticas
Control de tráfico	Li-Fi puede ser usado en autovías para aplicaciones de control de tráfico. Los coches pueden llevar luces delanteras o traseras de LED que se puedan comunicar con otros coches y prevenir accidentes de tráfico
Postes de luz	Cualquier poste de luz se podría convertir en un punto de acceso gratuito de datos
Alivio del espectro	Los problemas de escasez de radio frecuencias se podrían solventar usando Li-Fi

Tabla 1. Ventajas de las comunicaciones Li-Fi [1].

2.2. MACHINE LEARNING

Dentro de los diferentes campos de conocimiento que encontramos dentro de las matemáticas, cuando se junta con otros campos como la informática, podemos encontrar la ciencia de datos. Su objetivo es tratar de obtener información útil o nuevos conocimientos a través del manejo y procesamiento de grandes cantidades de datos.

Dentro de la ciencia de datos, reside el campo de la inteligencia artificial, con técnicas de aprendizaje automático (Machine Learning) o aprendizaje profundo (Deep Learning) [6].

Los modelos más exitosos de aprendizaje automático son aquellos que a partir de un conjunto de datos conocido, son capaces de automatizar procesos de toma de decisiones, a esta forma de aprender se le llama aprendizaje supervisado, donde se conocen tanto la entrada como la salida de datos, y el algoritmo a través de los datos conocidos, es capaz de aprender y proporcionar una salida dada una entrada. Dentro de los modelos de aprendizaje también podemos encontrar el no supervisado, donde al algoritmo no se le proporcionan datos de salida, si no únicamente de entrada. El aprendizaje no supervisado es un tipo de aprendizaje más complejo y difícil de comprender, y en este trabajo no se utilizará ya que con los métodos que se explicarán a continuación, resultara suficiente para entender los diferentes temas que se van a tratar.

Probablemente la parte más importante dentro del proceso de aprendizaje automático sea entender el conjunto de datos que queremos entrenar y tener claro que queremos hacer con estos, ya que cada algoritmo es diferente a la hora de interpretar y procesar cada tipo de dato.

Dentro del aprendizaje supervisado, podemos encontrar la clasificación y la regresión. En este trabajo haremos únicamente uso de modelos de regresión, el objetivo final es predecir un número, y se diferencia sobre la clasificación en que, una predicción en un modelo de regresión puede ser un número aproximado, por ejemplo, si se intenta predecir cual será la cifra de ingresos que obtendrá una empresa en un determinado año, basándose en varios factores como puede ser los ingresos de otros años, la cifra resultante podría ser 40001, cuando el valor esperado podría ser 40000, por lo que es un valor con cierta desviación. En los modelos de clasificación esto no es posible, un ejemplo podría ser una base de datos con audios de latidos de corazones con determinadas enfermedades, un modelo con la suficiente cantidad de muestras debería ser capaz de predecir que patología puede estar sufriendo un determinado paciente escuchando los latidos de su corazón, y decir si tiene alguna anomalía en su ritmo cardiaco o no, no existe un término medio entre sufrir algún tipo de patología cardiaca o no. Simplificando, gráficamente la diferencia entre ambos métodos se puede observar en la figura 1:

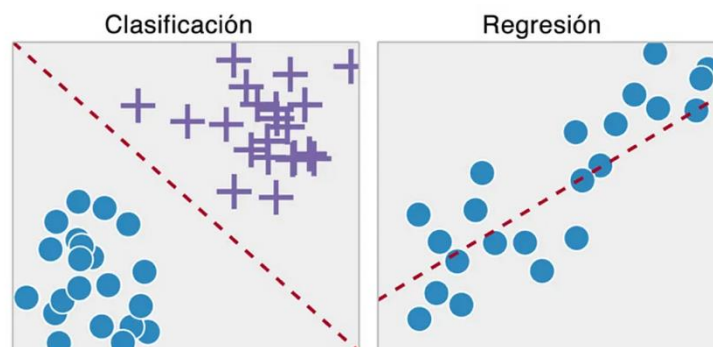


Figura 1. Descripción gráfica de los modelos de clasificación y de regresión.

En el aprendizaje supervisado, el objetivo final es crear un modelo a partir del entrenamiento de datos para después poder hacer predicciones lo más precisas posibles. Si los conjuntos de entrenamiento y prueba tienen suficientes puntos en común, se espera que el modelo también sea preciso en el modelo de prueba, pero en algunos casos se puede querer construir un modelo demasiado complejo para la cantidad de datos que disponemos, a este suceso se le llama “*Overfitting*” o sobreajuste, esto ocurre cuando se ajusta demasiado un modelo a las particularidades del conjunto de entrenamiento, pero

no es capaz de generalizarse a los nuevos datos. En contraparte, si el modelo es demasiado simple, es posible que no se pueda capturar todos los aspectos y variabilidad de los datos, produciéndose lo que se llama “*Underfitting*” o desajuste. Es importante tener un buen punto medio que producirá el mejor rendimiento de generalización, como se puede observar en la figura 2:

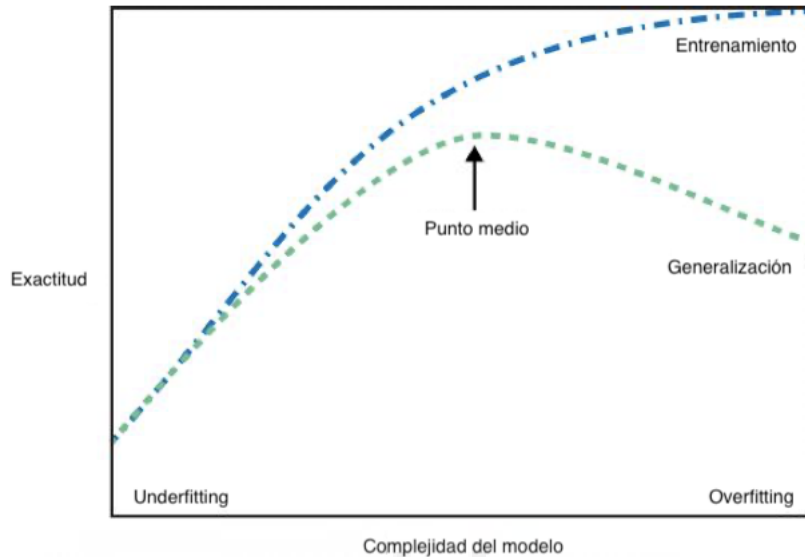


Figura 2. Compensación entre la complejidad del modelo y la precisión del entrenamiento y las pruebas.

Por último, cabe destacar que para la implementación de modelos de predicción a través de Machine Learning, vamos a utilizar lenguaje de programación Python. Python es el lenguaje más utilizado dentro de las aplicaciones de ciencia de datos, combina el poder de los lenguajes de programación de propósito general con la gran manejabilidad de algunos lenguajes de programación de dominio específico, como puede ser MATLAB. Con Python podemos cargar grandes cantidades de datos de cualquier archivo, tiene una amplia gama de funciones con las que se puede procesar datos o imágenes, se puede interactuar directamente con el código, además de permitir la creación de interfaces gráficas de usuario (GUI) [5].



CAPÍTULO 3. MATERIALES Y MÉTODOS

3.1. HARDWARE

En lo referente al hardware, hemos utilizado el dispositivo Analog Discovery 2, donde podremos visualizar y exportar las diferentes medidas que haremos sobre el sistema de comunicaciones.

3.1.1. ANALOG DISCOVERY

Analog Discovery, desarrollado por Digilent, es un instrumento multipropósito diseñado para poder medir, visualizar, generar, grabar y controlar circuitos de señal mixta de todo tipo. Tiene las ventajas de tener un tamaño pequeño y un coste bajo, proporcionando un sistema suficientemente potente para ocupar el lugar de algunos instrumentos de laboratorio [2].



Figura 3. Dispositivo Analog Discovery 2.

Como podemos observar en la figura 3, en la parte frontal del dispositivo encontramos una serie de entradas y salidas que podemos ver esquematizadas en la figura 4:

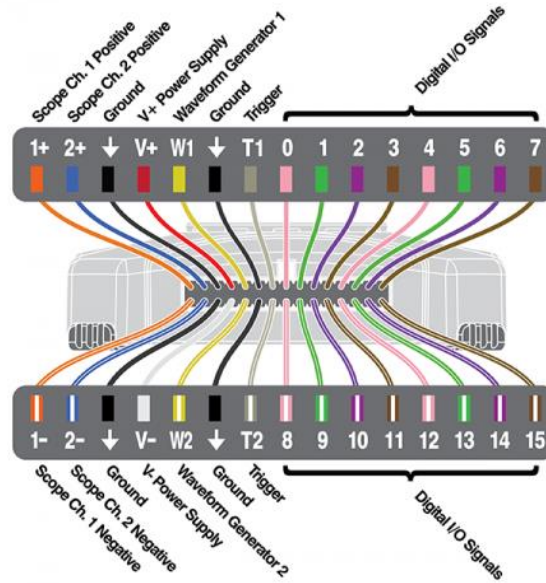


Figura 4. Esquema de conexiones Analog Discovery 2.

Para nuestro proyecto únicamente usaremos los pines '1+', '1-', correspondientes al canal 1, '2+', '2-', al canal 2, '↓', siendo la toma de tierra, y por último 'W1', el generador de señal, quedando finalmente el esquema de conexionado que se ve en la figura 5:

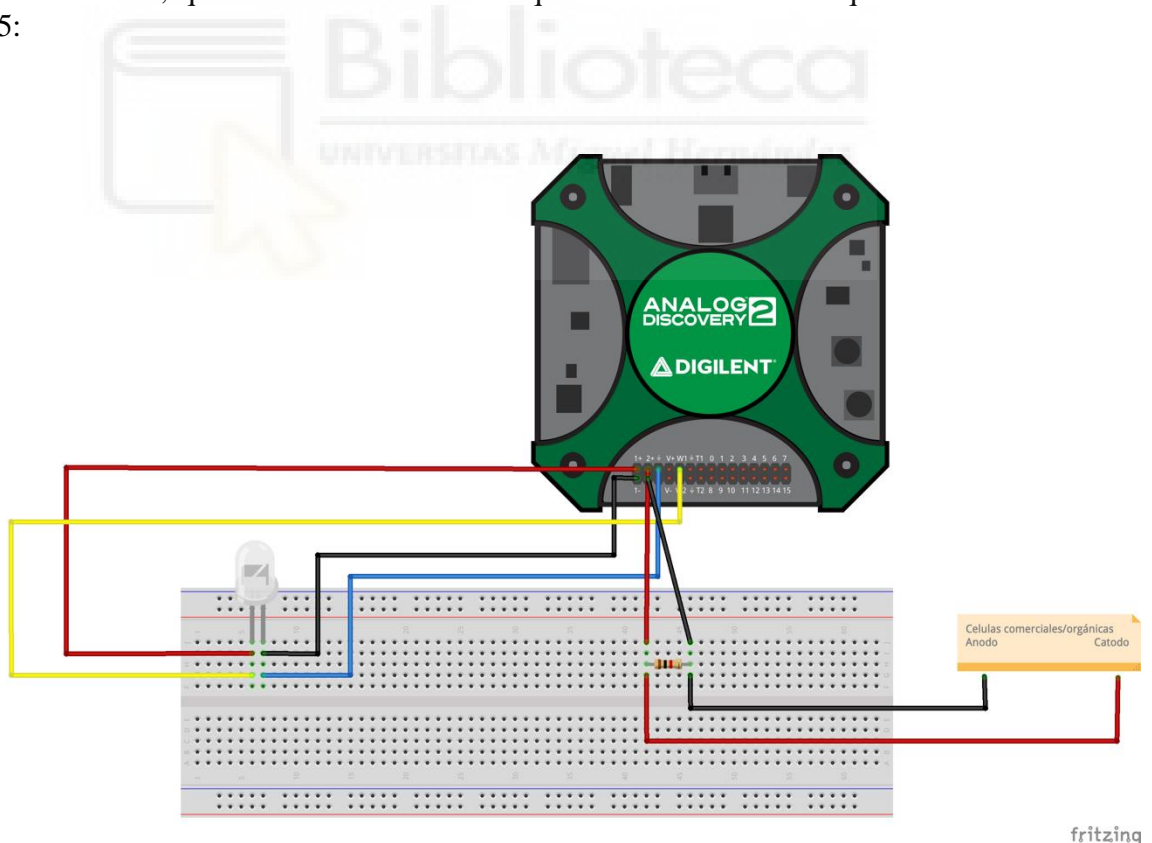


Figura 5. Esquema de conexionado.

3.2. SOFTWARE

Para poder recoger la información, procesarla y simularla en diferentes escenarios es necesario el uso de diferentes softwares que se van a detallar a continuación.

3.2.1. WAVEFORMS

Waveforms es un software que nos proporciona una interfaz para poder interactuar con diferentes dispositivos como Analog Discovery, además incluye una interfaz de programación de aplicaciones publicas (API), que permite al usuario crear aplicaciones personalizadas [3]. Esta interfaz nos resultara muy útil, ya que es necesario programar el comportamiento que queremos que siga nuestro generador de luz, cambiando diferentes parámetros como la frecuencia, offset o distancia del emisor al receptor, además de exportar los diferentes ficheros que posteriormente utilizaremos en otros programas.

Dentro del software podemos diferenciar 3 ventanas que se usarán principalmente, el generador de onda, donde podremos introducir algunos datos como la frecuencia o el voltaje que queremos introducir a nuestro sistema. También tenemos el osciloscopio, donde podremos ver visualmente el comportamiento de los canales 1 y 2, emisor y receptor respectivamente. Finalmente tenemos un script, siendo la parte más importante ya que es donde definiremos el comportamiento de todo el sistema, desde donde podemos controlar las dos anteriores ventanas, y exportar los archivos necesarios.



Figura 6. Entorno de trabajo Waveforms.

En la figura 6 podemos observar la ventana del osciloscopio, en amarillo (Canal 1) aparece la señal correspondiente a la fuente emisora de luz, mientras que en azul (Canal 2), vemos la señal que capta la célula orgánica fotovoltaica. El principal parámetro que nos interesa es el voltaje Pico-Pico, que nos servirá de gran utilidad a lo largo del proyecto, además del histograma que podemos observar representado en la parte inferior de la figura. También es interesante variar los parámetros de “Offset” y “Range” que encontramos en el cuadro de la derecha, ya que conforme variamos la frecuencia y la distancia, debemos cambiar algunos de estos parámetros para poder visualizar correctamente la señal.

El software Waveforms ha sido fundamental para la recogida y posterior exportación de datos.

3.2.2. GOOGLE COLABORATORY

Google Colaboratory, más comúnmente llamado Google Colab, es un producto de Google Research que permite escribir y ejecutar código arbitrario de Python en el navegador, está especialmente diseñado para trabajar en ámbitos de aprendizaje automático, análisis de datos e investigación [4]. Para este proyecto, nos centraremos en la parte de aprendizaje automático, utilizando técnicas de Machine Learning que serán posteriormente desarrolladas.

Para entrenar modelos de Machine Learning, utilizaremos la librería “Scikit-learn”, ya que es una de las librerías más utilizadas hoy en día para ello.



CAPÍTULO 4. MEDIDAS EXPERIMENTALES

Para este trabajo principalmente usamos dos tipos de fotodetectores, uno de carácter comercial, y varias células orgánicas fotovoltaicas que serán desarrolladas en los siguientes apartados.

4.1. LED COMERCIAL

Como transmisor se utilizó un LED comercial de 5 mm blanco, que proporciona un voltaje entre 3 y 3,2 V y una corriente de 20 mA. El uso de un LED comercial como transmisor es debido a que es un componente realmente barato y fácil de conseguir, otorgando buenos resultados ya que la limitación principal a la hora de medir los parámetros vendrá dada por el fotodetector.

También se barajó utilizar un LED comercial de alta potencia de 10 mm, pero al realizar las medidas transmitiendo una señal sinusoidal, se pudo observar un rizado en determinados puntos que podían afectar a la calidad de la señal recibida, y puesto que la diferencia de la potencia medida no era tan grande, se decidió seguir el trabajo utilizando el LED de 5 mm.



Figura 7. LED comercial.

4.2. CÉLULAS COMERCIALES

Las primeras medidas fueron tomadas con un fotodetector comercial ya que la respuesta es significativamente mejor comparada con las células orgánicas, por lo que vamos a poder ver mejores resultados. En este caso se usó el fotodetector comercial BPW 21, un fotodiodo de silicio usado para el rango espectral visible, especialmente adecuado para el rango de 350 nm a 820 nm. A continuación, podemos observar su sensibilidad espectral y características direccionales en la Fig. 8 y Fig. 9:

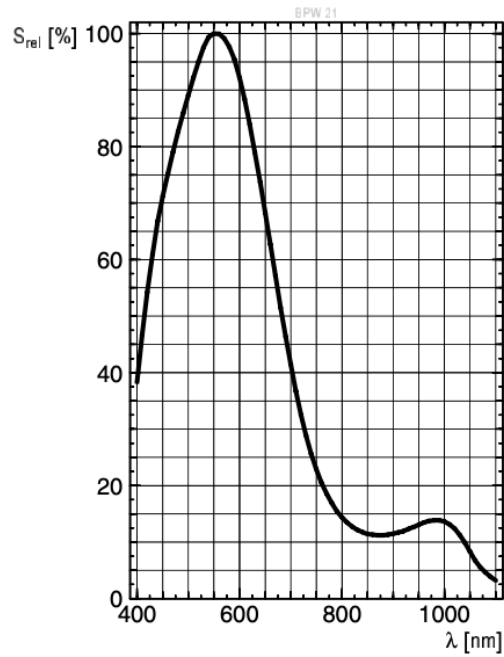


Figura 8. Sensibilidad espectral relativa.

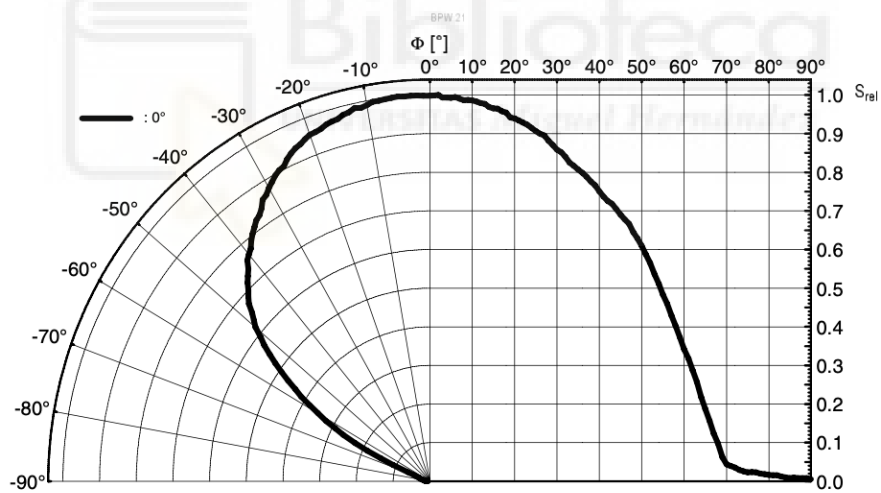


Figura 9. Características direccionales.

Podemos ver que en la Figura 8 el dispositivo alcanza su máxima sensibilidad en frecuencia a 550 nm, mientras que en la Figura 9 la máxima sensibilidad direccional se alcanza a 0°.

4.2.1. REGRESIONES LINEALES Y POLINÓMICAS EN FUNCIÓN DE LA FRECUENCIA

El fundamento básico de la regresión lineal parte de la ecuación de la recta:

$$y = mx + b \quad (1)$$

Donde 'y' es la variable dependiente, representa un valor en el eje vertical para un valor dado de 'x', que es la variable independiente. 'b' es la ordenada en el origen, representa el punto donde la recta cruza el eje 'y'. Y 'm' es el valor de la pendiente, indica la inclinación de la línea, si su valor es positivo, la recta subirá de izquierda a derecha, si es negativa, la recta bajará de izquierda a derecha. Viene definida por:

$$m = \frac{y_2 - y_1}{x_2 - x_1} \quad (2)$$

Para la primera medida, usando las células comerciales anteriormente mencionadas, y exportando sus datos a un archivo .csv para poder manejarlos con Google Colaboratory, pasamos a elaborar un código en el script donde usando la librería "Scikit-learn", podemos crear modelos de regresiones para predecir el comportamiento de nuestro sistema a diferentes frecuencias. Para ello, pasamos a explicar los diferentes códigos elaborados, mediante el siguiente pseudocódigo:

```
# Importar las bibliotecas necesarias
```

```
Importar data_split, linear_regression, mean_squared_error, numpy, matplotlib, pandas
```

```
# Cargar archivo CSV
```

```
datos ← ReadCSV("/ruta/al/archivo/40Muestras.csv", delimiter=";")
```

```
# Reformatear los arrays para que sean bidimensionales
```

```
frecuencias ← Reshape(datos["Frecuencias"], (-1, 1))
```

```
voltajes ← Reshape(datos["Voltajes"], (-1, 1))
```

```
# Convertir voltajes a milivoltios
```

```
for i ← 1, len(voltajes) do
```

```
    voltajes[i] ← voltajes[i] * 1000
```


end for

```
# Crear y entrenar el modelo de regresión lineal, junto con algunos parámetros de calidad
```

```
modelo ← TrainModel(LinearRegression(), frecuencias, voltajes)
```

```
Imprimir "Offset:", modelo.intercept_
```

```
Imprimir "Pendiente:", modelo.coef_
```

```
Imprimir "Precisión del modelo:", modelo.score(frecuencias, voltajes)
```

```
# Predecir voltajes de salida para frecuencias dadas
```

```
frecuencias_prueba ← [[1], [20], [50], [60]]
```

```
Imprimir "Resultado:", modelo.predict(frecuencias_prueba)
```

```
# Graficar resultados
```

```
PlotScatter(frecuencias_prueba, modelo.predict(frecuencias_prueba), color="red")
```

```
PlotLine(frecuencias_prueba, modelo.predict(frecuencias_prueba), color="black")
```

```
PlotPoints(frecuencias, voltajes, color="green")
```

```
SetLabels('Frecuencia (kHz)', 'Voltaje Pico-Pico (mV)', 'Regresión Lineal')
```

```
ShowPlot()
```

Viendo el pseudocódigo, podemos resumir todos los códigos en un esquema bastante parecido, donde primero se importan las librerías, después subimos el archivo .csv, redimensionamos el array, creamos el modelo de regresión que se estime oportuno y por último lo representamos gráficamente con todos los detalles y ajustes que necesite el modelo. A continuación, detallamos el primer código para tener una referencia de como son, pero no se volverán a explicar los siguientes códigos a excepción de algunas partes destacables, puesto el objetivo de este trabajo no es explicar los apartados de programación, además de que quedaría un trabajo innecesariamente largo debido a la cantidad de diferentes códigos que se han creado.

```
from sklearn.model_selection import train_test_split #para dividir
los datos en conjuntos de entrenamiento y prueba.
from sklearn.linear_model import LinearRegression #para el modelo de
regresión lineal.
```

```

from sklearn.metrics import mean_squared_error #para evaluar el
rendimiento del modelo.
import numpy as np #para manipulación de matrices y operaciones
numéricas.
import matplotlib.pyplot as plt
import pandas as pd

```

En el anterior código podemos ver como primero se implementan diferentes librerías, *train_test_split*, para dividir los datos en conjuntos de entrenamiento y prueba, *LinearRegression*, para crear y entrenar un modelo de regresión lineal, *mean_squared_error*, para calcular el error cuadrático medio del modelo (se utilizará más adelante para otros códigos), *numpy*, para manipulación de matrices y operaciones numéricas, *matplotlib.pyplot*, para crear gráficos y finalmente *pandas*, para manipulación de datos, especialmente para leer el archivo .csv.

```

#Subimos el archivo .csv
datos = pd.read_csv("/content/40Muestras.csv", delimiter=";")

#Ajustamos el array para que sea bidimensional
frecuencias = datos["Frecuencias"].values.reshape(-1,1)
voltajes = datos["Voltajes"].values.reshape(-1,1)

for i in np.arange(0, len(voltajes)):
    voltajes[i] = voltajes[i]*1000

```

A continuación, pasamos a cargar los datos del archivo “40Muestras.csv” donde especificamos que “;” será un delimitador de nuestras columnas, esto es debido a la forma en que interpreta Google Colaboratory los archivos. Más adelante pasamos a ajustar las columnas “Frecuencias” y “Voltajes” para que sean matrices bidimensionales, necesarias para poder aplicar el modelo de regresión.

```

#Creamos nuestro modelo de regresion lineal
regresion = LinearRegression()
modelo = regresion.fit(frecuencias,voltajes)
print("Offset",modelo.intercept_)#Donde empieza el primer punto de
la recta
print("Pendiente",modelo.coef_)
print("Precision del modelo")
print(modelo.score(frecuencias,voltajes))

#Definimos las diferentes frecuencias para las cuales queremos
calcular el voltaje a la salida
frecuencia = [[1],[20],[50],[60]]
print("resultado", modelo.predict(frecuencia))

```

Creamos una instancia del modelo de regresión lineal mediante “LinearRegression” y lo entrenamos con los datos de frecuencia y voltaje de nuestro archivo, después mostramos por pantalla algunos datos como el Offset y la pendiente de

nuestro modelo. En las dos últimas líneas simplemente creamos un array de frecuencias e imprimimos para cada frecuencia el valor de voltaje que el modelo debería predecir según la regresión creada.

```
plt.scatter(frecuencia, modelo.predict(frecuencia), color="red")
plt.plot(frecuencia, modelo.predict(frecuencia), color="black")
plt.plot(frecuencias, voltajes, 'go', label = 'future')
plt.xlabel('Frecuencias(kHz)', fontsize=15)
plt.ylabel('Voltaje Pico-Pico(mV)', fontsize=15)
plt.title('Regresión lineal', fontsize=15)
plt.xticks(fontsize=14)
plt.yticks(fontsize=14)
plt.grid()
plt.show()
```

Por último, añadimos las líneas necesarias para representar adecuadamente nuestro modelo de regresión lineal.

Para el primer código, se ha utilizado un archivo de 40 muestras, que van incrementándose 1 kHz desde 1 hasta 40, este proceso está automatizado, por lo que las muestras se van sacando automáticamente mediante el software de Waveforms. En la Fig. 10 podemos ver el conjunto de muestras, junto con una línea que representa el modelo de regresión lineal entrenado a partir del número de muestras que se le han pasado. En adición, se puede ver que la línea está representando datos hasta 60 kHz, esto se debe a que el modelo es capaz de predecir que datos esperamos obtener siguiendo la tendencia del modelo entrenado, el cual arroja los siguientes resultados:

```
Offset [460.54970904]
Pendiente [[-0.45640785]]
Precision del modelo
0.49513149088805664
resultado [[460.09330119]
[451.42155197]
[437.72931637]
[433.16523783]]
```

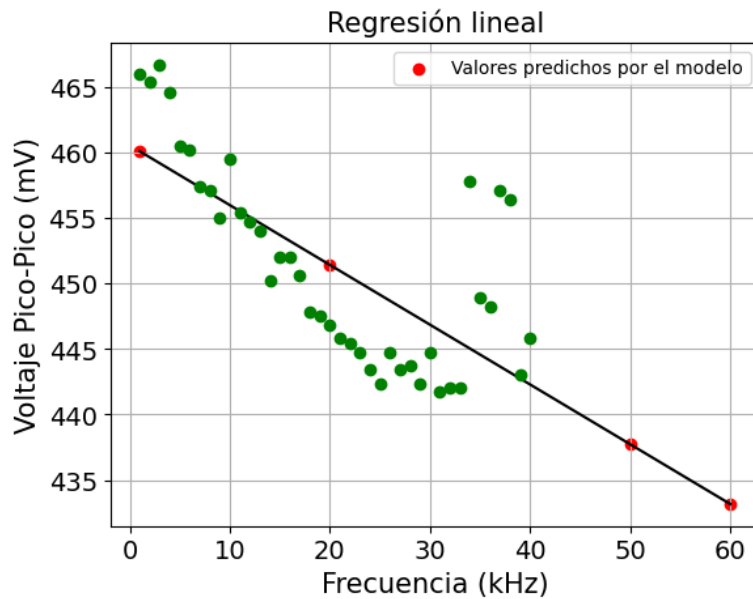


Figura 10. Regresión lineal para 40 muestras con datos automatizados.

Para la Fig. 11, a diferencia de la Fig. 10, los datos se han ido recogiendo a mano, es decir, cambiando manualmente los valores de la frecuencia y apuntando cada valor de voltaje recibido. Como se puede ver, la tendencia es mucho menos dispersiva, lo cual reduce significativamente el error, por lo que el método sería más adecuado a la hora de obtener mejores resultados, además, la precisión del modelo aumenta considerablemente respecto a la Figura 10, de 0,495 a 0,874, cabe destacar que la precisión que se mide en los ejemplos esta normalizada respecto a 1, por lo que no son valores porcentuales. Sin embargo, no es una forma viable de hacer las representaciones ya que como veremos más adelante, el número de muestras se puede incrementar de forma considerable.

También se puede ver un valor de Offset que indica el punto en el cual la recta cruza el eje y, que como se ha explicado antes, se corresponde con el termino 'b' de la ecuación (1).

```
Offset [456.97680488]
Pendiente [[-0.75331051]]
Precision del modelo
0.8744178642344459
resultado [[456.22349437]
[441.91059475]
[419.31127955]
[411.77817448]]
```

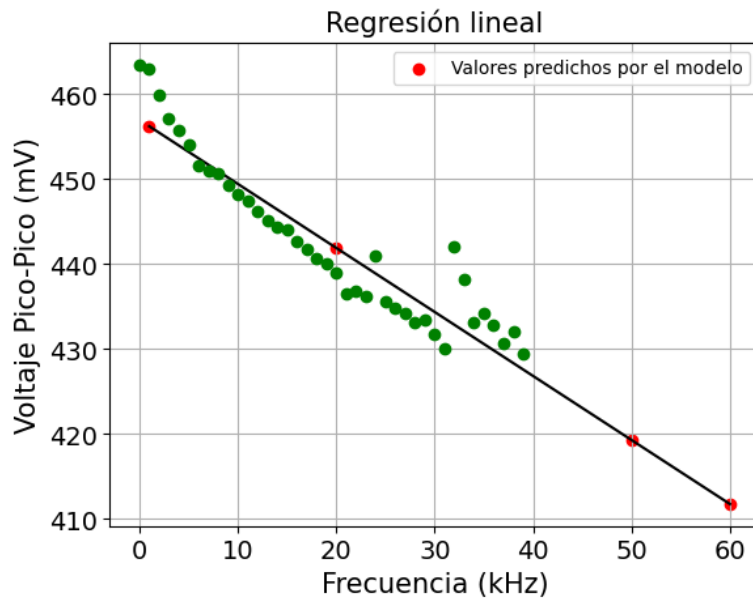


Figura 11. Regresión lineal para 40 muestras con datos introducidos manualmente.

Para la Fig. 12, se ha utilizado un conjunto de muestras más amplio, llegando hasta los 1000 kHz, tomando una muestra por kHz. La tendencia de los datos ya no es tan lineal, se puede observar una ligera curvatura, lo cual hace que el modelo de regresión ya no se ajuste adecuadamente a los datos que obtenemos.

```

Offset [370.90121938]
Pendiente [[-0.44682571]]
resultado [[ 370.45439366]
 [ 361.96470514]
 [ 348.55993378]
 [ 344.09167666]
 [ 281.53607698]
 [ 125.14707779]
 [ -8.9006358 ]
 [-120.60706379]
 [-178.69440634]]
Precision del modelo
0.8962218084967649

```

Por lo que se puede observar en los parámetros que miden la calidad del modelo, la precisión ha aumentado, siendo incluso superior a las Figuras 7 y 8, pero al ser una regresión lineal, se puede ver que, al intentar predecir valores por encima de unos 800 kHz, los resultados que arroja no tienen sentido, ya que son valores de voltaje negativos, por lo que se debe utilizar otro modelo que se ajuste de forma correcta a los datos que se introducen.

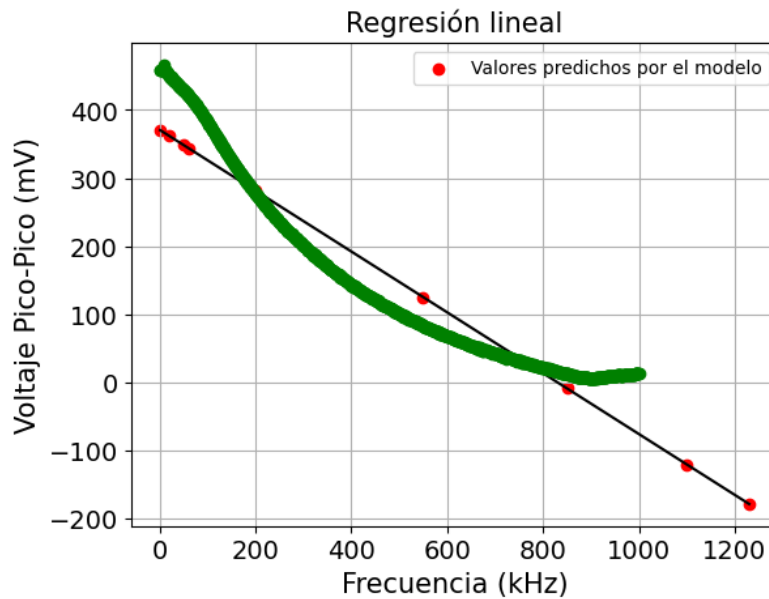


Figura 12. Regresión lineal para 1000 muestras.

Como se ha visto anteriormente, el modelo de regresión lineal no siempre se ajusta de manera adecuada a un conjunto de datos introducido, por lo que de aquí en adelante se utilizará un modelo de regresión polinomial que se ajustará mejor a los siguientes datos. En la Fig. 13 tenemos el mismo número de muestras que en la Fig. 12, pero en este caso hemos creado un modelo de regresión polinomial, aumentando el grado polinómico para ver en que punto se ajusta mejor a nuestros datos. Al ver la figura. 12, se podría pensar que la regresión es mucho menos dispersiva ya que visualmente se ve mucho más uniforme, sin embargo, esto se debe a que la ventana que estamos visualizando es mucho mayor que la figura 10 y 11, ya que pasamos de ver 60 kHz a ver 1200 kHz, y los efectos de dispersión producidos por la toma de medidas automática con el software Waveforms ya no son apreciables. Finalmente obtenemos los siguientes resultados:

```
Precision del modelo de Grado 1
0.8956238102161604
Precision del modelo de Grado 2
0.9968710194426849
Precision del modelo de Grado 3
0.9989489652543221
Precision del modelo de Grado 4
0.9989695433114432
Precision del modelo de Grado 5
0.999682778195126
Precision del modelo de Grado 6
0.9992650318331924
```

A partir de este punto ya no se utilizarán algunos parámetros mencionados anteriormente como el Offset o la pendiente a no ser que se considere relevante para la explicación de las medidas.

Como se puede ver, el modelo de regresión polinómica de grado 5 es el que da una mejor precisión (0.9996), sin embargo, se puede concluir que a partir del grado 2, los resultados ya son lo suficientemente buenos como para poder usar cualquiera de los grados excepto el grado 1, ya que este sería semejante a una regresión lineal.

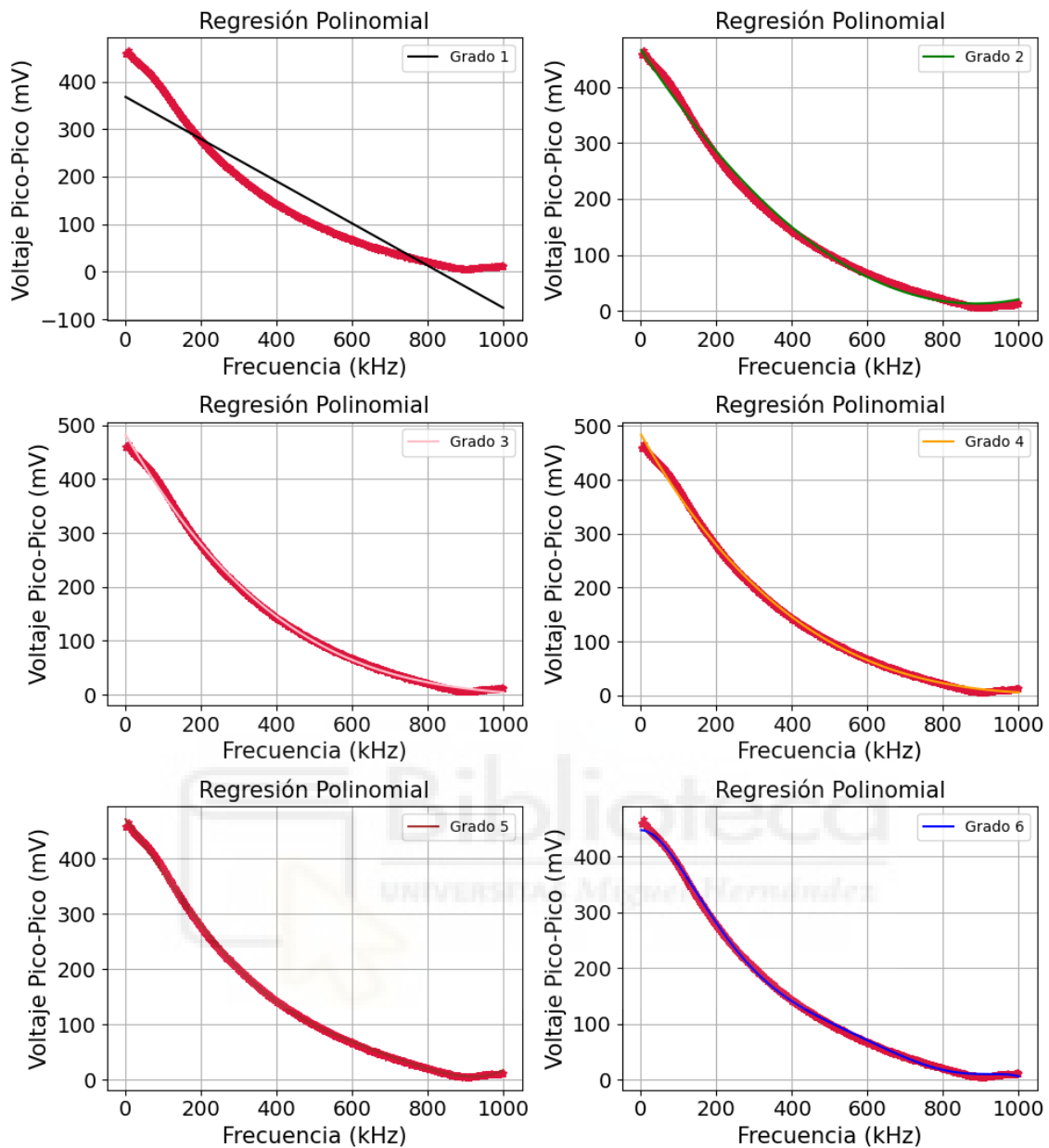


Figura 13. Comparación de grados para una regresión polinomial con 1000 muestras.

Con todo lo expuesto anteriormente, podemos concluir que las regresiones lineales pueden ser útiles para modelos más dispersivos, pero cuando tenemos un conjunto de muestras más definido donde además el comportamiento no es muy lineal, es más interesante el uso de regresiones polinomiales, pudiendo ajustar mejor o peor el modelo según el grado que le indiquemos.

4.2.2. REGRESIONES LINEALES Y POLINÓMICAS EN FUNCIÓN DE LA DISTANCIA

Para este punto se han tomado 60 muestras de voltaje en función de la distancia, una muestra cada 2,5 mm, desde 0 hasta 15 cm. Las medidas se toman manualmente, ya que, al tomarse en función de la distancia, automatizar el proceso sería complicado y requeriría más tiempo que el que se tarda en tomar las medidas. En la Fig. 14 se plantea

un modelo de regresión lineal, que como se puede observar, no es un modelo fiable a la hora de predecir los resultados, por lo que en la Fig. 15 es utilizada una regresión polinomial, que, a pesar de no ajustarse perfectamente al conjunto de datos, si que da unos resultados mucho más fiables.

Los valores tomados de voltaje tienen una forma de filtro paso bajo, y se puede observar un aumento del voltaje en el primer centímetro, esto se puede deber a que el fotodetector comercial capta un mayor rango de la luz procedente del LED a 1 cm de distancia. También se puede concluir que entre 2 y 6 cm hay una caída muy abrupta del voltaje, y a partir de los 6 cm se suaviza significativamente. Para la Fig. 15 se utilizó una relación polinómica de grado 6, se ajustó manualmente hasta ver cual daba los mejores resultados en función de su precisión, como vemos, no se ajusta perfectamente, esto se debe a que el modelo no siempre se puede ajustar con exactitud al conjunto de datos, especialmente cuando hay una caída de valores muy abrupta, sin embargo, la precisión del modelo es lo suficientemente buena, por lo que los valores no se desvían mucho.

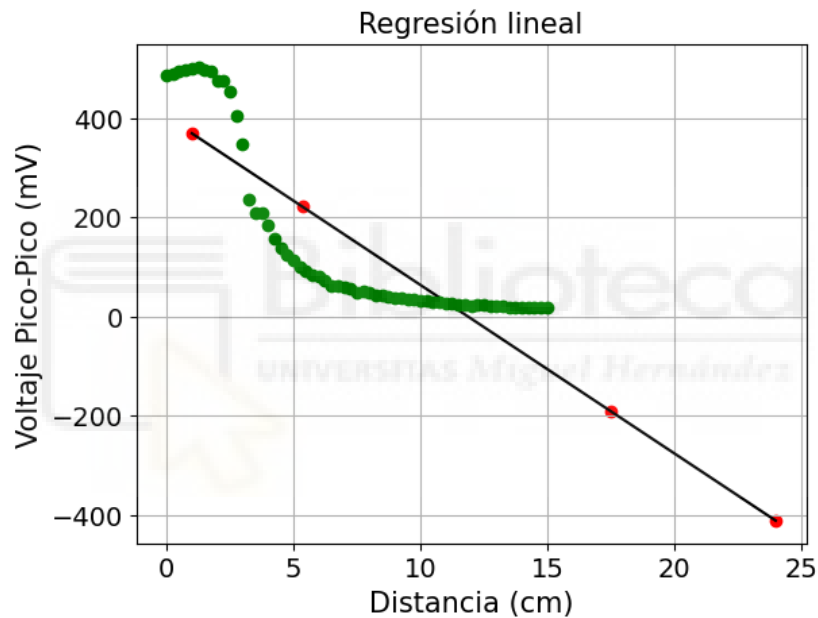


Figura 14. Regresión lineal.

Precision del modelo
0.7205476367245858

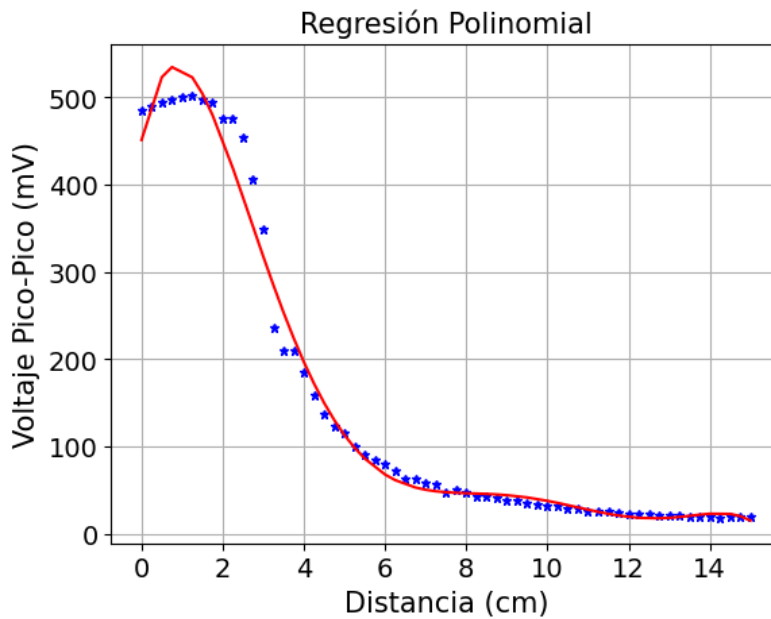


Figura 15. Regresión polinomial.

Precision del modelo
0.9860300138910542

4.2.3. TIEMPOS DE SUBIDA Y BAJADA

En este apartado se analizarán los tiempos de subida y bajada que se miden dentro del programa de Waveforms. En la Figura 16 se observa que el tiempo de bajada es mayoritariamente superior al tiempo de subida, sin embargo, el tener una bajada tan marcada en los primeros 100 kHz, con una forma de exponencial negativa, las muestras se separan demasiado por lo que no permite ver de forma adecuada el inicio del modelo.

El modelo de regresión que hemos aplicado en este caso es de grado 3, pero como vemos no se ajusta de manera correcta a los datos, esto se debe a que los modelos de regresión polinomial no se ajustan bien a fuertes cambios de tendencia, creando las ondulaciones se observan en la figura 16 debido a que no le da tiempo a ajustarse adecuadamente. Este desajuste se produce de la misma forma aunque sigamos subiendo el grado del polinomio.

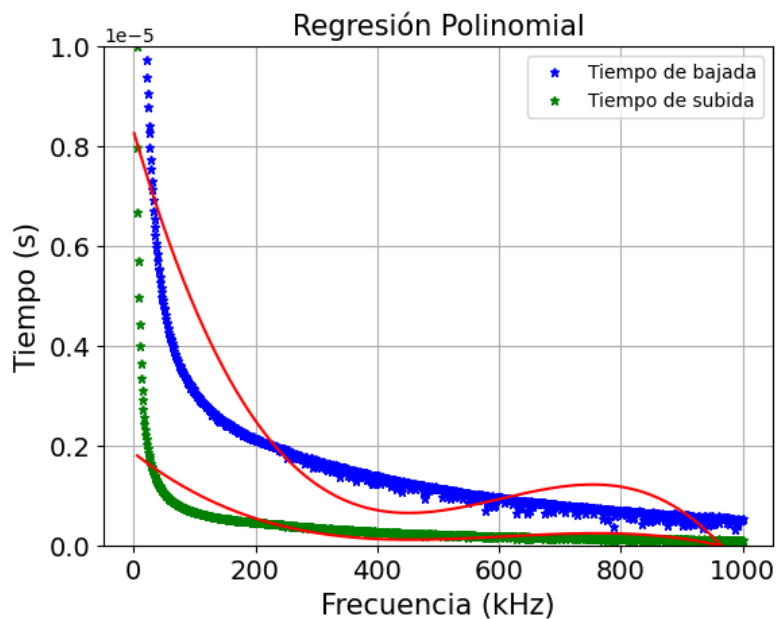


Figura 16. Regresión polinomial.

En la Figura 17 se ha normalizado el tiempo de bajada respecto al de subida, y vemos que aproximadamente entre 1 y 2 kHz los valores del tiempo normalizado son menores a 1, lo cual nos indica que, entre estas dos frecuencias, el tiempo de subida es mayor al de bajada. La frecuencia exacta a la que se produce el cambio en el cual predomina el tiempo de subida frente al de bajada, se produce aproximadamente en 2,1955 kHz, esto lo podemos saber haciendo uso de la función “predict”, utilizada anteriormente para predecir el valor del modelo de regresión en un determinado punto:

```
tiempolkHz = modelo.predict(grado.transform([[2.1955]]))
print(tiempolkHz)
```

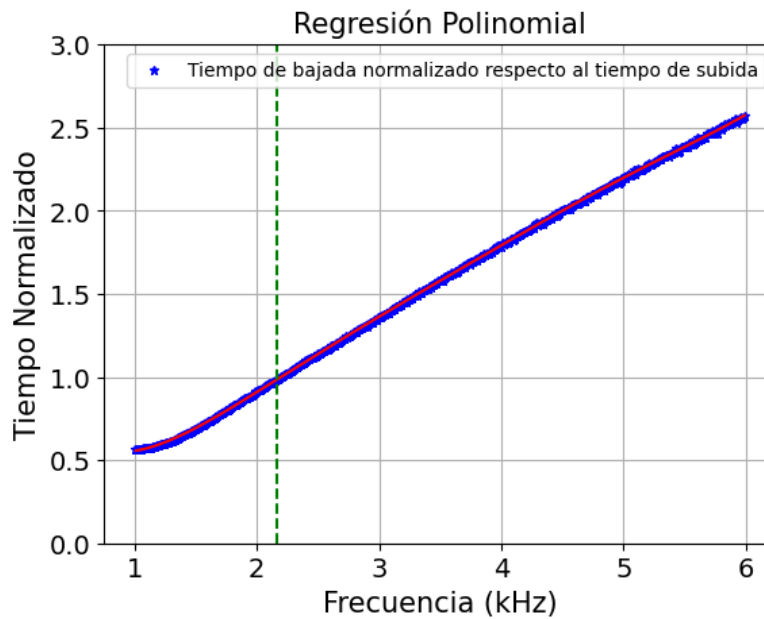


Figura 17. Regresión polinomial.

Para corroborar que el cálculo realizado anteriormente es correcto, vamos a ajustar la ventana para visualizar de una mejor forma el inicio de la regresión, centrándola entre 1 y 6 kHz. De nuevo se puede observar que en la Figura 18, el cambio de tendencia entre los tiempos de subida y bajada se produce entorno a los 2 kHz, como habíamos podido comprobar por el razonamiento que se ha seguido anteriormente.

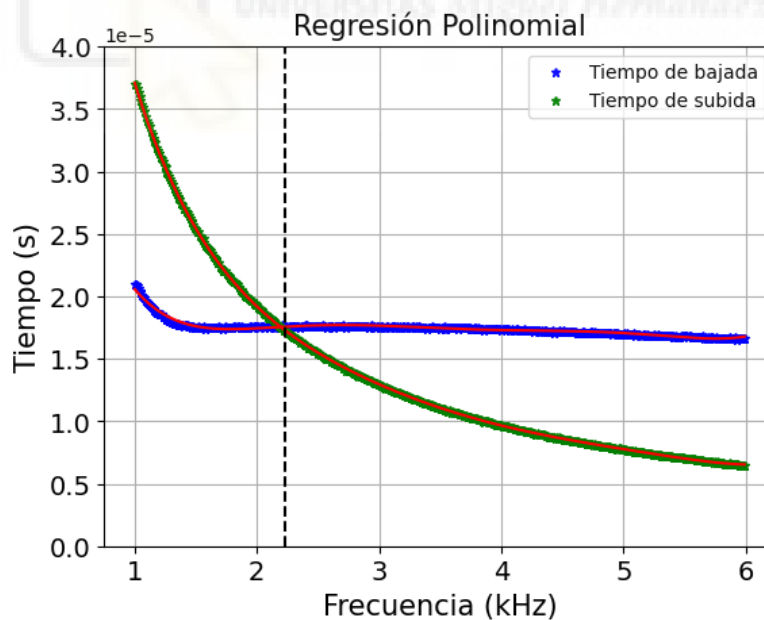


Figura 18. Regresión polinomial.

4.2.4. CICLO DE TRABAJO

El ciclo de trabajo es la relación entre el tiempo en que la señal se encuentra es estado activo, es decir, que el led este encendido, y el periodo de la señal, y viene definida por la siguiente formula:

$$D = \frac{\tau}{T} \quad (3)$$

Podemos diferenciar entre ciclo positivo y ciclo negativo, siendo el primero el tiempo total en el cual el LED permanece encendido, y el segundo el tiempo total en el cual permanece apagado. Los ciclos de trabajo se miden porcentualmente y en la Figura 19 se puede visualizar una representación de ambos ciclos junto con un modelo de regresión de grado 4, donde cabe destacar que aproximadamente un poco antes de los 800 kHz, se produce un cambio en el cual el ciclo positivo comienza a tener un valor porcentual superior al ciclo negativo. Además, a partir de los 400 kHz se observa una clara dispersión que disminuye la precisión del modelo de regresión.

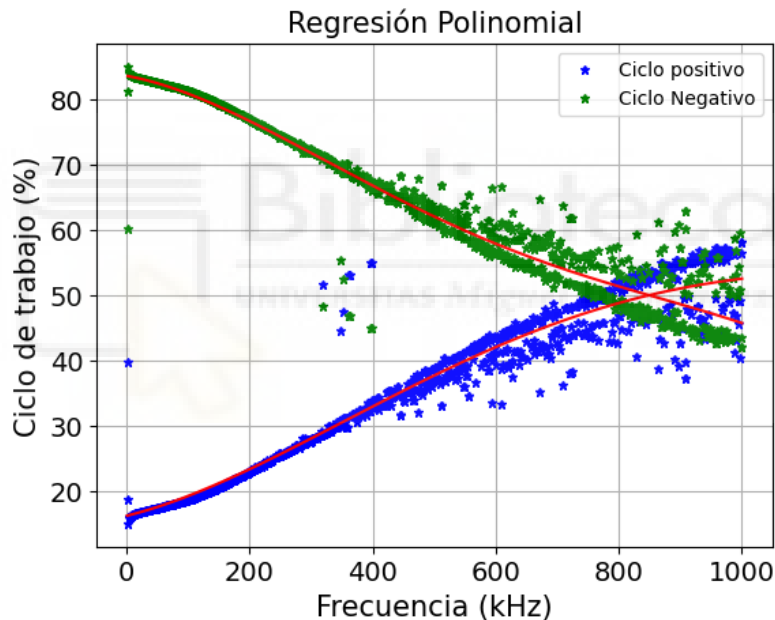


Figura 19. Regresión polinomial.

Para poder observar como sigue la tendencia del ciclo de trabajo, se ha aumentado el numero de muestras hasta 2000 kHz y el grado del polinomio a 4 para ajustarlo mejor, donde podemos concluir que a medida que aumenta la frecuencia por encima de 800 kHz, el ciclo positivo aumenta respecto al negativo.

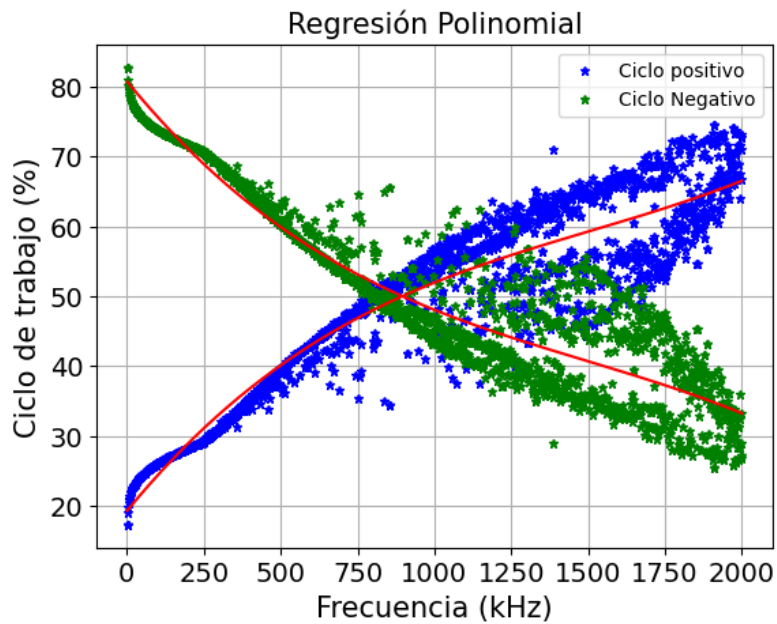


Figura 20. Regresión polinomial.

En la figura 21, hemos normalizado el ciclo positivo respecto al negativo sobre las 1000 muestras, en este caso, debido a la dispersión tan grande que se produce en la figura 19, se decidió coger los valores que genera el modelo de predicción. Observamos el cambio de tendencia entre los dos ciclos cuando la recta supera el valor normalizado de 1, este se produce a los 856 kHz, que visualmente coincide con lo que podemos ver en la figura 19.

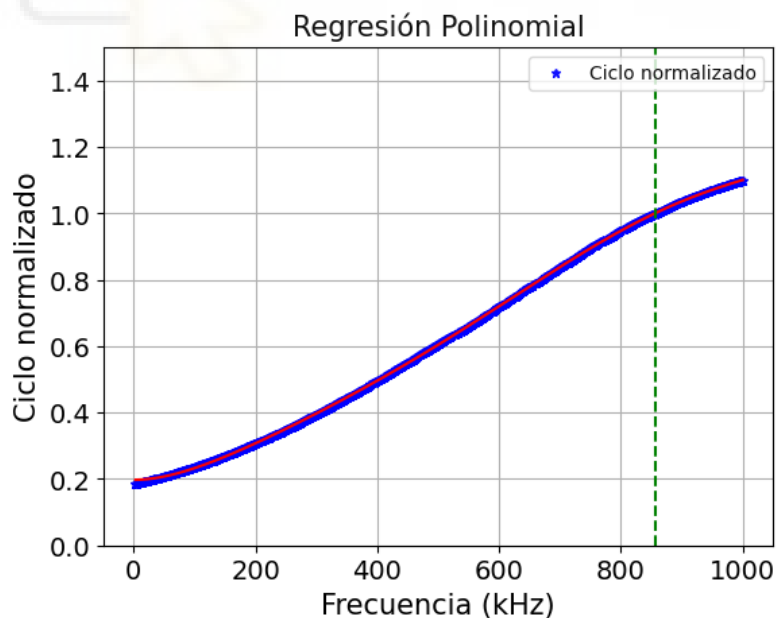


Figura 21. Regresión polinomial

4.2.5. PARÁMETROS Q-SNR-BER

Para el siguiente apartado se analizarán algunos parámetros típicos de los sistemas de telecomunicaciones como son la Q, SNR y BER.

El factor de calidad se define como Q y sirve para medir de forma general la calidad de nuestro sistema. La SNR tiene dos valores asociados según el estado del sistema sea alto o bajo, mientras que el parámetro Q nos permite combinar ambas SNR, obteniendo un valor más certero en la calidad del sistema. Su valor se puede calcular como el cociente entre la diferencia de los valores de tensión V1 y V0, y la diferencia entre la potencia cuadrática media σ_1 y σ_0 , que representan la mitad del ancho de banda que se puede observar sobre el 60% del valor total.

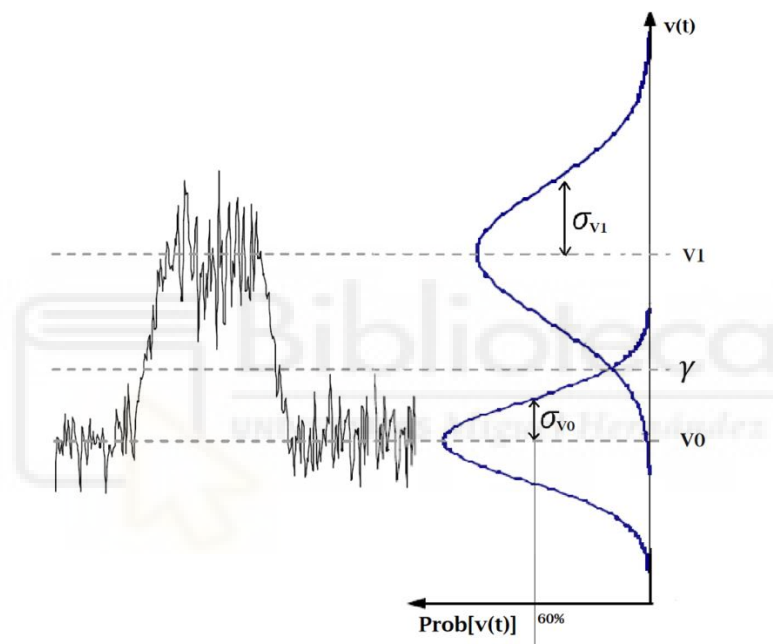


Figura 22. Representación del cálculo del factor de calidad Q.

Todo lo que se ha expuesto anteriormente lo podemos resumir en la siguiente ecuación: [9]

$$Q = \frac{V_1 - V_0}{\sigma_1 - \sigma_0} \quad (4)$$

La SNR hace referencia a la relación señal a ruido, se utiliza para cuantificar el nivel de calidad de una señal en presencia de ruido y se define como la relación entre la potencia de la señal y el nivel de potencia de ruido que acompaña a la señal. Una SNR alta es esencial para una comunicación clara y eficiente, mientras que una SNR baja puede resultar en una comunicación deficiente y errores en la transmisión de datos.

Para obtener la expresión que define la SNR de la señal, vamos a calcular el nivel de señal (S) por un lado, y el nivel de ruido (N) por otro. El nivel de señal se puede definir como el producto de la tensión por la corriente:

$$S = v * i \quad (5)$$

Sustituyendo a través de la ley de Ohm podemos llegar a la siguiente expresión:

$$S = \frac{v^2}{R} = i^2 * R \quad (6)$$

Ahora, el nivel de ruido se define como:

$$N = \frac{1}{T} \int_0^T i^2 * R dt = \sigma_i^2 * R \quad (7)$$

$$N = \frac{1}{T} \int_0^T v^2 * \frac{1}{R} dt = \sigma_v^2 * \frac{1}{R} \quad (8)$$

Donde T es el periodo de integración y σ es la potencia cuadrática media.

Una vez hemos obtenido estos dos valores por separado, podemos relacionarlos de la siguiente forma: [9]

$$SNR = \frac{S}{N} = \frac{v^2/R}{\sigma_v^2/R} = \frac{v^2}{\sigma_v^2} \quad (9)$$

$$SNR = \frac{S}{N} = \frac{i^2 * R}{\sigma_i^2 * R} = \frac{i^2}{\sigma_i^2} \quad (10)$$

A partir de este punto, se utilizará la expresión (7) en el cálculo de la SNR.

En tercer lugar, está la tasa de error de bit (BER), se define como la relación entre el número de bits erróneos y el número total de bits transmitidos durante un intervalo de tiempo específico y sirve para evaluar la calidad y fiabilidad del enlace de comunicación. Una baja BER es indicativa de un sistema eficiente y robusto, mientras que una alta BER sugiere problemas de calidad y posibles necesidades de mejora en el sistema de comunicación, se define mediante la siguiente ecuación: [10]

$$BER = \frac{1}{2} * Erfc\left(\frac{Q}{\sqrt{2}}\right) \quad (11)$$

Donde Erfc es la función de error complementaria, que se calcula automáticamente mediante la función “erfc” incluida en Python.

Todo lo anteriormente explicado lo podemos resumir en los siguientes códigos:

```
from sklearn.model_selection import train_test_split #para dividir
los datos en conjuntos de entrenamiento y prueba.
from sklearn.linear_model import LinearRegression #para el modelo
de regresión lineal.
from sklearn.metrics import mean_squared_error #para evaluar el
rendimiento del modelo.
import numpy as np #para manipulación de matrices y operaciones
numéricas.
```

```

import matplotlib.pyplot as plt
import pandas as pd
from sklearn.preprocessing import PolynomialFeatures
from scipy.special import erfc

#Subimos el archivo .csv
datos = pd.read_csv("/content/Histograma 6 A6 led alta luminosidad
blanco 5 mm 3.3 V 1 KHz 0 cm.csv",delimiter=";")
Voltaje = datos["Voltaje"]
Porcentaje = datos["Porcentaje"]

```

Si miramos dentro de los archivos .csv que importamos a Google Colaboratory, y centramos la ventana en valor de voltaje cercanos a 0, podemos ver lo representado en la siguiente tabla:

8185	-0.0034004	0
8186	-0.003058	0
8187	-0.0027156	0
8188	-0.0023732	0.025
8189	-0.0020308	1.275
8190	-0.0016884	12.125
8191	-0.001346	33.075
8192	-0.0010036	22.8375
8193	-0.0006612	6.4125
8194	-0.0003188	0.575
8195	0.0000236	0.225
8196	0.00036598	0.225
8197	0.00070838	0.1125

Tabla 2. Archivo .csv con datos de voltaje y porcentaje.

Se puede ver que un poco antes de llegar al valor de voltaje de 0, el valor porcentual llega hasta el 33.075%, lo cual representa el pico de valor en V0, por lo que en el código vamos a hacer un bucle que recorra todos los valores de porcentaje y busque el valor mas alto, una vez lo encuentre, lo multiplicará por 0.6 para buscar un valor que sea un 40% inferior al máximo.

```

MaxPorc = max(Porcentaje);#Calculamos el máximo porcentual para
averiguar donde está V0
#Bucle para calcular el valor de V0
for i in range(0,len(Porcentaje)):
    if Porcentaje[i] == MaxPorc:
        V0 = Voltaje[i];
        aux = i;

V060Porc = MaxPorc*0.6

```



```

for i in range(0, len(Porcentaje)-1):
    V0ValorActP = Porcentaje[i]
    V0ValorSigP = Porcentaje[i+1]
    if (V0ValorActP >= V060Porc >= V0ValorSigP):
        V0ValorActV = Voltaje[i]
        V0ValorSigV = Voltaje[i+1]
        m = (V0ValorSigP-V0ValorActP)/(V0ValorSigV-V0ValorActV)
        V060Volt = ((V060Porc-V0ValorActP)/m)+V0ValorActV

sigma0 = abs(V0)-abs(V060Volt)

aux = aux+100;
#Bucle para calcular el valor de V1
IntervaloV1 = Porcentaje[aux::]
MaxPorcV1 = max(IntervaloV1)
V160Porc = MaxPorcV1*0.6

for i in range(aux, len(Porcentaje)-1):
    V1ValorActP = Porcentaje[i]
    V1ValorSigP = Porcentaje[i+1]
    if Porcentaje[i] == MaxPorcV1:
        V1 = Voltaje[i];
    if (V1ValorActP >= V160Porc >= V1ValorSigP):
        V1ValorActV = Voltaje[i]
        V1ValorSigV = Voltaje[i+1]
        m = (V1ValorSigP-V1ValorActP)/(V1ValorSigV-V1ValorActV)
        V160Volt = ((V160Porc-V1ValorActP)/m)+V1ValorActV

sigma1 = abs(V160Volt)-abs(V1)

print('Valor Porcentual V0: ', MaxPorc);
print('Valor Porcentual V1: ', MaxPorcV1);
print('Valor de V0: ', V0);
print('Valor de V1: ', V1);

print('Valor de sigma0: ', sigma0);
print('Valor de sigma1: ', sigma1);

SNR_1kHz = (V1**2)/(sigma1**2);
SNR_1kHz_dB = 10*np.log10(SNR_1kHz)
Q_1kHz = (V1-V0)/(sigma1+sigma0);
Q_1kHz_dB = 10*np.log10(Q_1kHz)
BER_1kHz = (1/2)*erfc(Q_1kHz_dB/(np.sqrt(2)));

print('SNR: ', SNR_1kHz)
print('SNR(dB): ', SNR_1kHz_dB)
print('Q: ', Q_1kHz)

```

```
print('Q (dB): ', Q_1kHz_dB)
print('BER: ', BER_1kHz)
```

Todas las gráficas están representando los valores en lineal y en dB μ V, para realizar esta conversión se ha seguido la siguiente ecuación:

$$dB\mu V = 20 * \log\left(\frac{V}{1\mu V}\right) \quad (12)$$

Siendo V el valor en lineal medido en microvoltios y multiplicándose por 20 debido a las relaciones de voltaje de estos parámetros.

En la Fig. 23 se observa como varia el valor del factor de calidad en función de la frecuencia, se puede ver que una vez se pasan los 100 kHz, el factor de calidad no varia tanto en comparación con la fuerte caída al inicio:

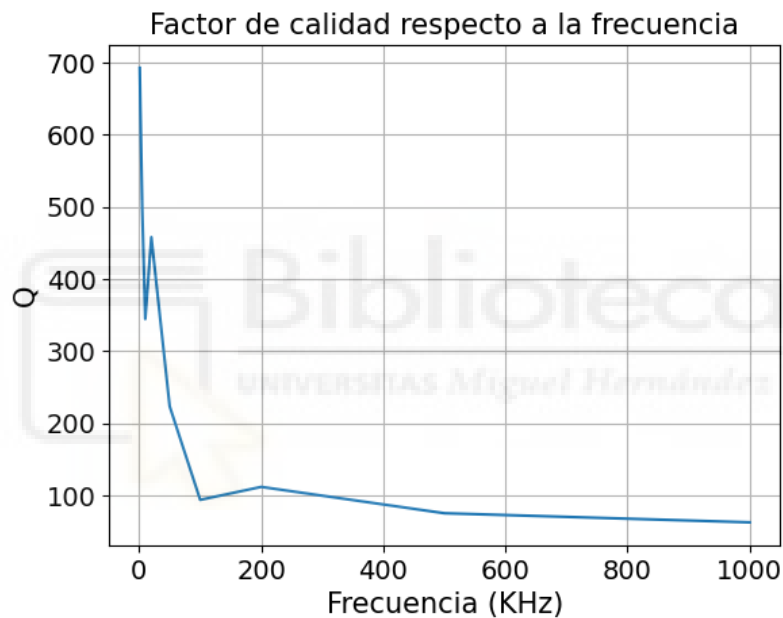


Figura 23. Relación entre el factor de calidad (Q) en lineal y la frecuencia.

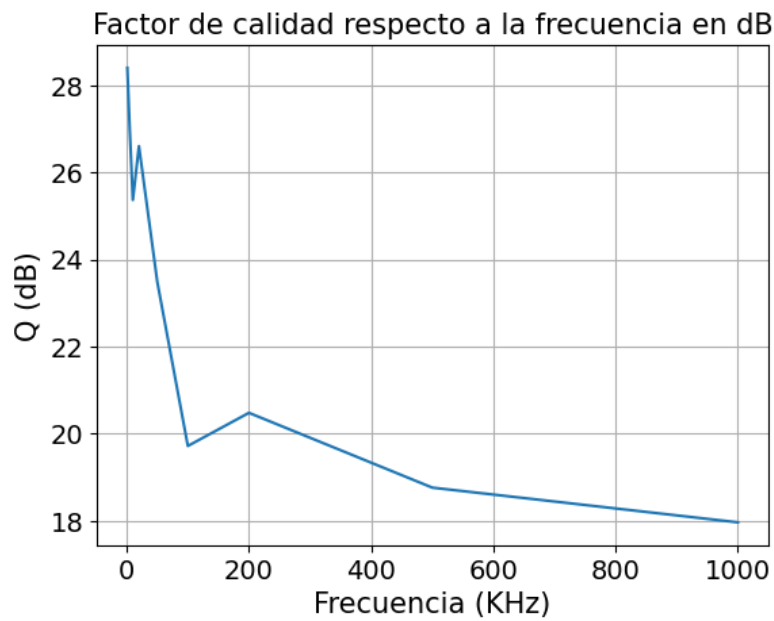


Figura 24. Relación entre el factor de calidad (Q) en dB y la frecuencia.

En cuanto a la SNR, en la Fig. 25 se aprecia un comportamiento muy similar al observado en la Fig. 23, donde se observa una fuerte caída al inicio seguida por una caída más moderada a partir de lo 100 kHz.

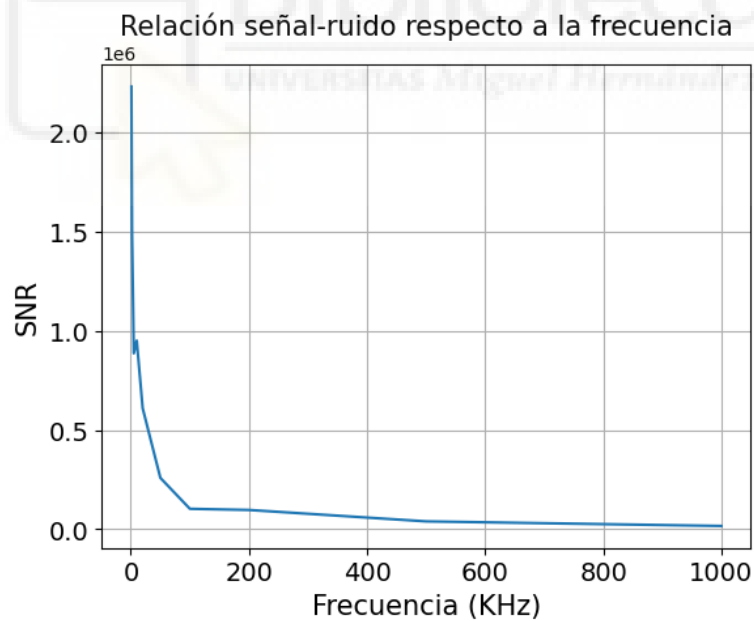


Figura 25. Relación entre la relación señal a ruido (SNR) en lineal y la frecuencia.

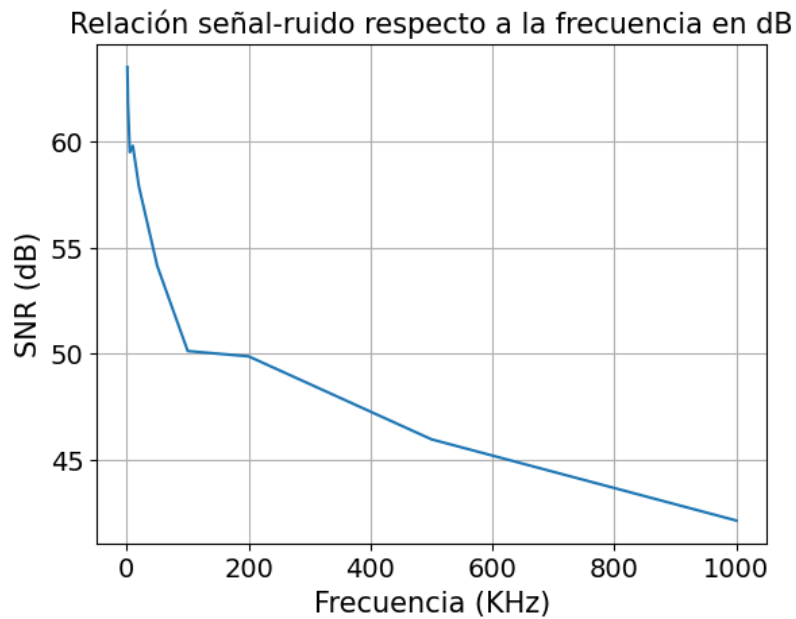


Figura 26. Relación entre la relación señal a ruido (SNR) en dB y la frecuencia.

Para la BER vemos un comportamiento inverso a las Fig. 23 y 25, en la Fig. 27 se puede apreciar como la BER aumenta considerablemente en los primeros 100 kHz, y a partir de ese punto tiene un aumento mucho más progresivo, esto tiene sentido puesto que a medida que aumentamos la frecuencia, la señal se degrada y por tanto el número de bits que detectará correctamente el sistema disminuirá, haciendo que aumente la tasa de error de bits.

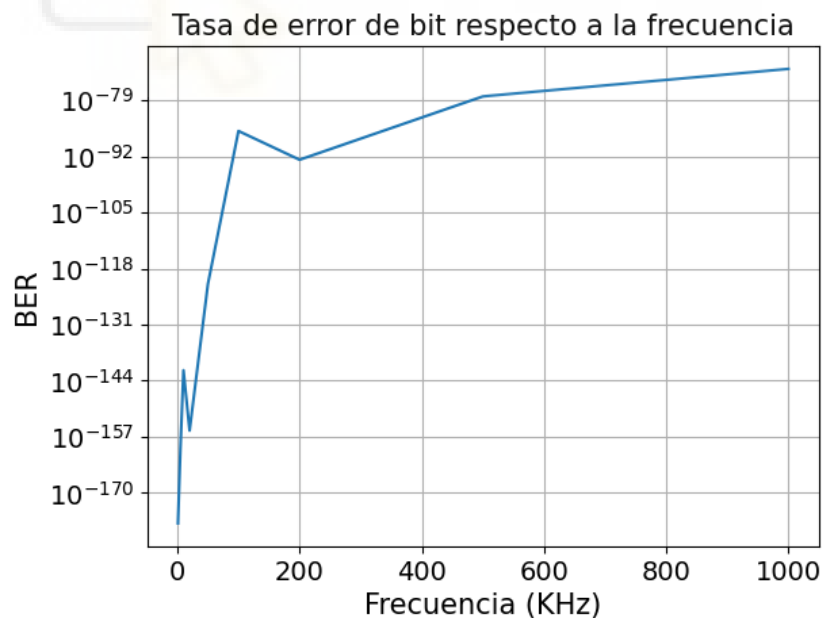


Figura 27. Relación entre la tasa de error de bit (BER) y la frecuencia.

En la figura 28 se hace una comparativa de la BER en función de la SNR, como se puede apreciar, a medida que aumenta la SNR de la señal, la BER disminuye, puesto que como se ha mencionado anteriormente, una SNR alta es indicativo de una buena

calidad en un sistema de comunicaciones, reduciendo así el número de bits erróneos que se detectarían.

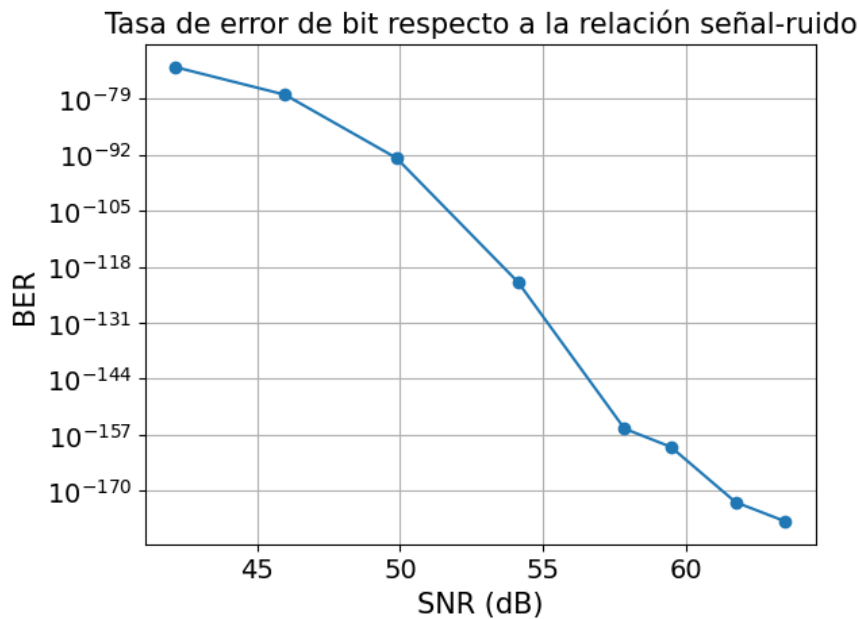


Figura 28. Relación entre la tasa de error de bit (BER) y la relación señal a ruido (SNR).

Como se puede ver, todas las figuras de este apartado dan valores muy buenos, especialmente a frecuencias bajas, esto se debe a que están calculadas en condiciones ideales, es decir, a 0 cm, por lo que no representan un ejemplo real de los valores que obtendríamos en aplicaciones prácticas de sistemas de comunicación Li-Fi.

4.3. CÉLULAS ORGÁNICAS

Para la realización de las siguientes medidas, utilizaremos unas células orgánicas fotovoltaicas fabricadas en la Universidad Miguel Hernández de Elche.

Estas células han sido creadas con el objetivo de seguir promoviendo el uso de tecnologías aplicadas al ámbito de las energías renovables, fomentando el cambio gradual hacia fuentes más sostenibles y limpias. Además, las células ofrecen diversas ventajas frente a los combustibles fósiles (principal fuente de generación eléctrica en la actualidad), como puede ser la obtención de energía a través de la radiación solar con la consecuente contribución a la reducción de gases de efecto invernadero que pueden ser provocados por los combustibles fósiles. También tiene un carácter descentralizado, ya que pueden ser instaladas en una amplia variedad de lugares, evitando así el uso de instalaciones centralizadas, donde un fallo en una zona tan focalizada puede suponer un corte en el suministro de una gran parte de la población, lo que en resumen se traduce en una mayor seguridad energética.

Actualmente, la industria fotovoltaica se basa principalmente en la tecnología de silicio, que pese a sus excelentes propiedades que ayudan a aprovechar una gran cantidad de radiación solar, su uso conlleva una serie de efectos negativos. La demanda del silicio es cada vez mayor, lo que puede suponer un problema de escasez futuro, elevando los costes debido a la alta demanda y la baja oferta, además, el proceso de extracción del silicio implica un alto coste de energía y recursos.

Es por esto por lo que las células solares orgánicas se ofrecen como una alternativa sostenible a los problemas expuestos.

La característica principal de las células solares orgánicas es que están dopadas con diferentes disolventes orgánicos que ayudan a mejorar la conductividad de la capa de transporte de huecos. En este trabajo nos centraremos en tres tipos de células dopadas con disolventes como dimetilformamida (DMF), tetrahidrofurano (THF) y dimetilsulfóxido (DMSO).

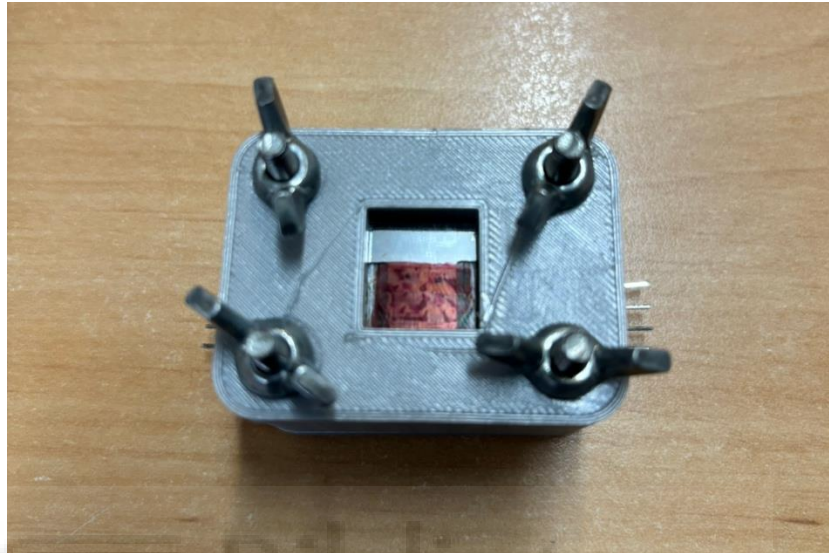


Figura 29. Portaplacas con célula orgánica fotovoltaica.

En la figura 29 podemos observar el porta placas construido mediante impresión 3D que se utilizó para la toma de medidas, que mediante un sistema métrico también construido mediante impresión 3D, se va variando la distancia en función de lo que queramos medir. Dentro del porta placas irá la célula orgánica, que está compuesta por 6 celdas de 6 mm^2 ($1,5 \text{ mm} \times 4 \text{ mm}$) enumeradas de arriba abajo y de derecha a izquierda.

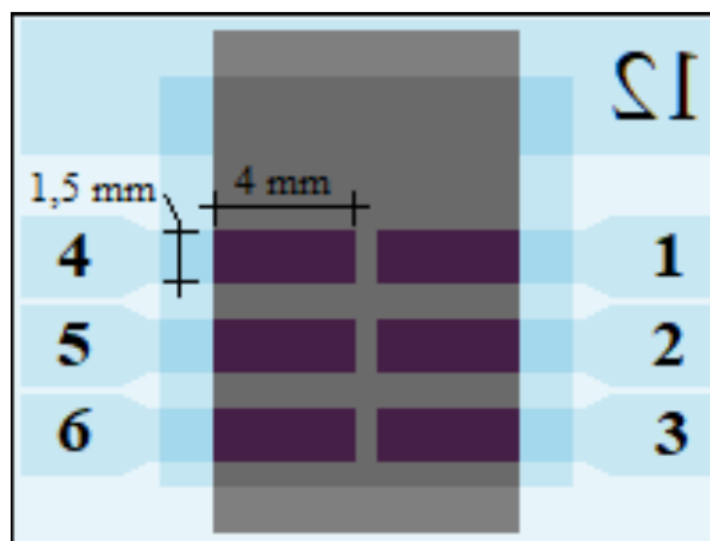


Figura 30. Ejemplo de célula orgánica fotovoltaica. [8]

4.3.1. REGRESIONES POLINÓMICAS EN FUNCIÓN DE LA FRECUENCIA Y DISTANCIA PARA CÉLULAS DE PLOMO

Inicialmente se utilizan dos células de plomo, este tipo de células son más recientes que las células que se verán en el apartado 4.3.2, pero están dopadas con compuestos diferentes por lo que después se podrá observar como varían los resultados de unas respecto de otras.

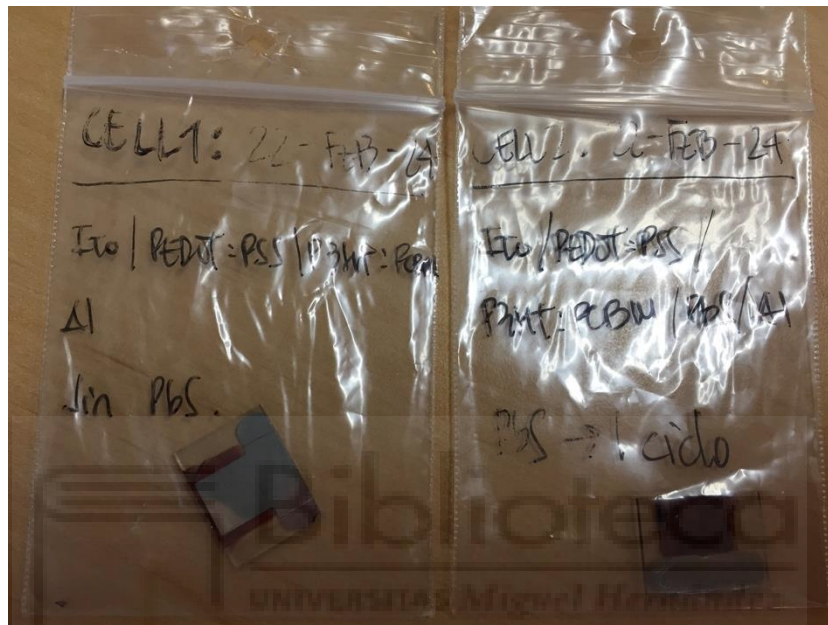


Figura 31. Células 1 y 2 de plomo.

Se van haciendo mediciones en frecuencia siguiendo la serie 1, 2, 5, 10, 20, 50, 100, 200, 500 y 1000 kHz, esta serie viene definida por códigos ya realizados en WaveForms para otros trabajos similares [7], y lo vemos representado en las figuras 32, 33, 34 y 35. Como se puede ver, a partir de 200 kHz la tensión recibida es inferior a 25 mV, siendo ya una señal bastante degradada.

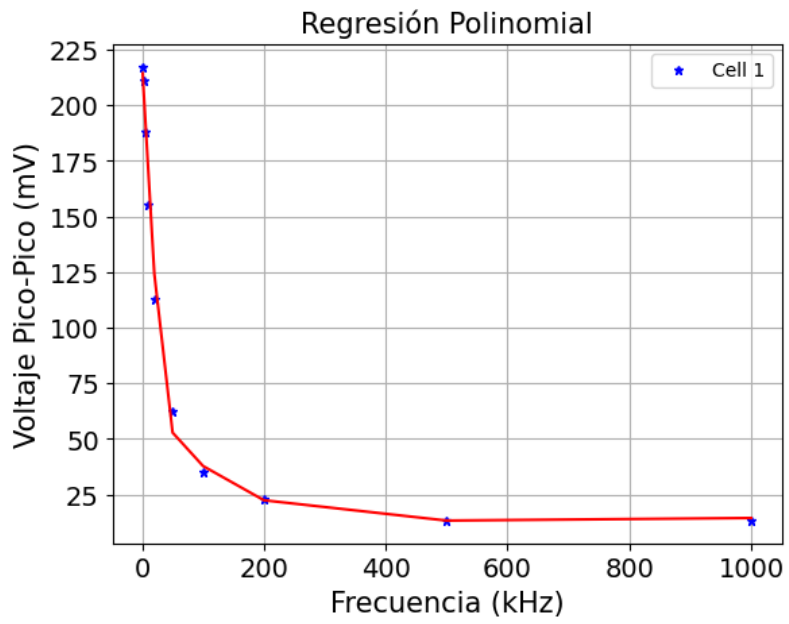


Figura 32. Regresión polinomial.

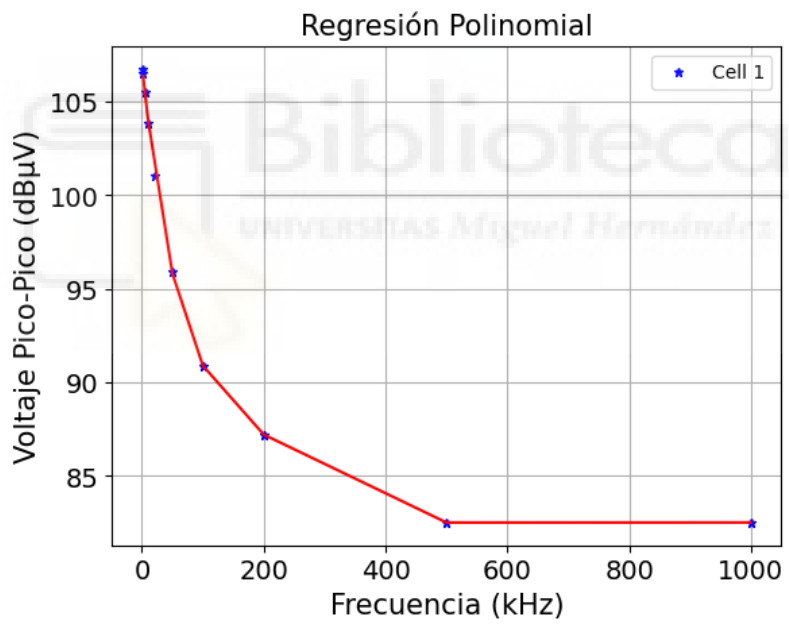


Figura 33. Regresión polinomial.

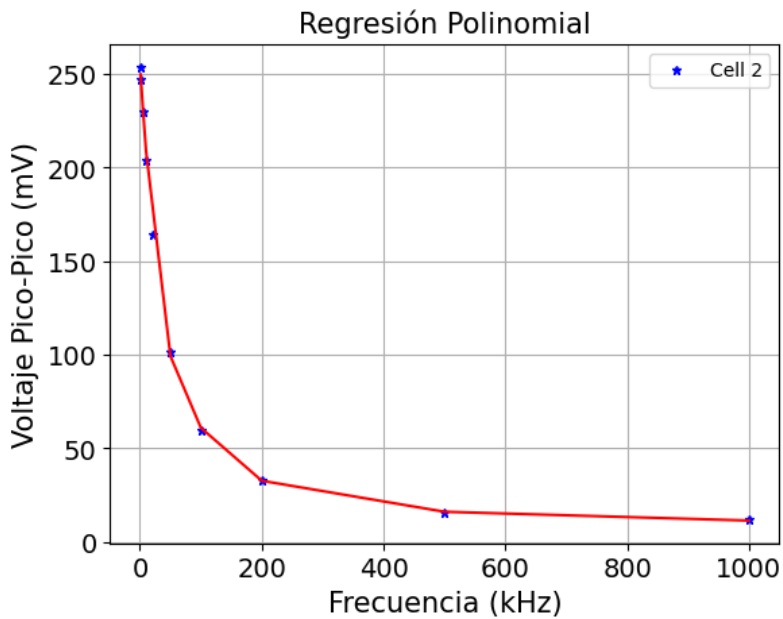


Figura 34. Regresión polinomial.

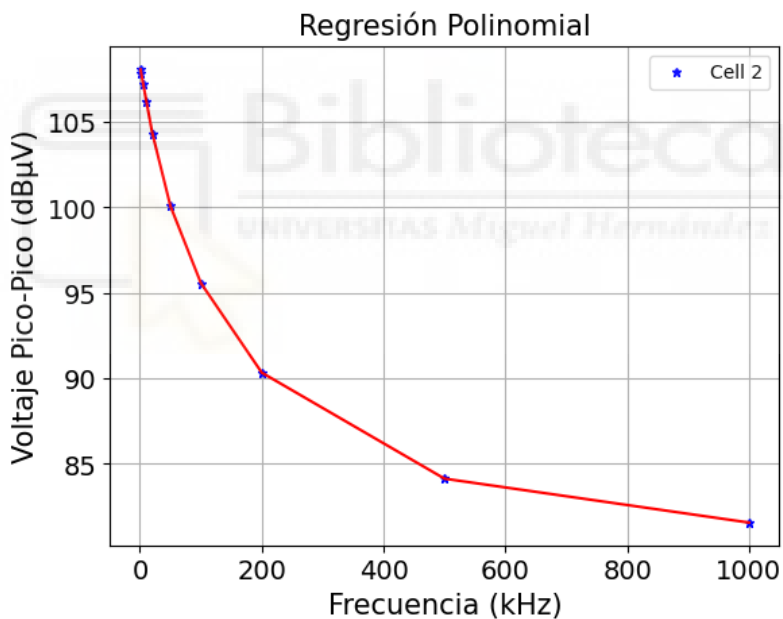


Figura 35. Regresión polinomial.

Para las figuras 36 y 37 se ha aumentado el número de muestras a 1000 para poder obtener un resultado más real, en la figura 35 se puede apreciar que, a partir de los 200 kHz, el sistema aumenta considerablemente la desviación de los resultados, esto se puede apreciar de mejor forma en la figura 37, donde el ruido se hace muy evidente al estar representado en dB. Esto representa un problema a la hora de crear los modelos de regresión, ya que como vemos en las figuras 36 y 37, cuando comenzamos a experimentar ruido, el modelo de regresión sufre pequeñas fluctuaciones debido a que no es capaz de mantener una posición fija. Es por esto que se hace necesario reducir el problema del ruido para obtener mejores precisiones en los modelos, lo cual se desarrollará más adelante, en el apartado 4.3.3.

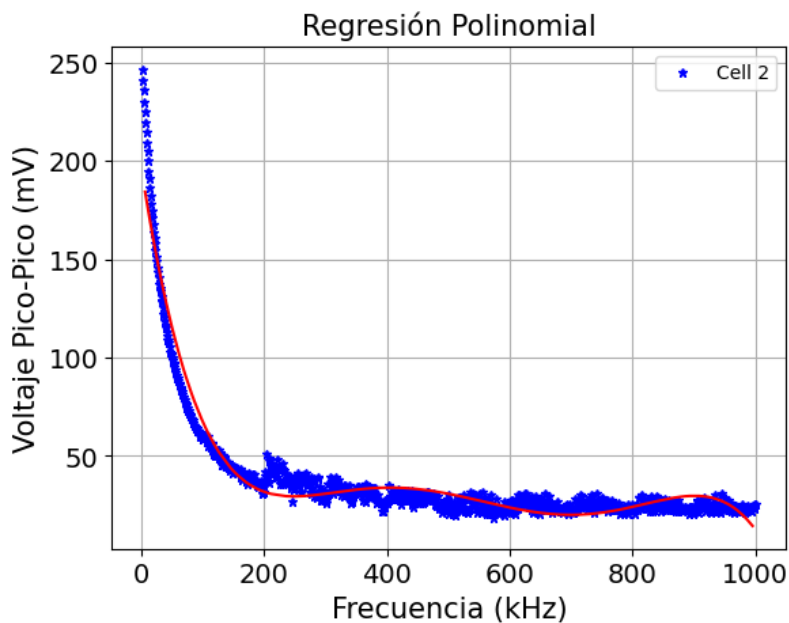


Figura 36. Regresión polinomial.

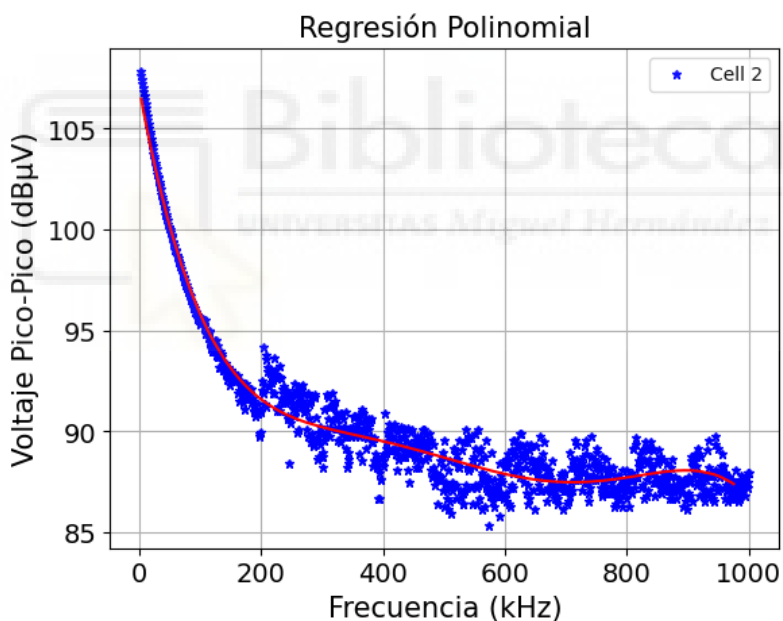


Figura 37. Regresión polinomial.

Para las medidas en distancia, nuevamente se toman 40 muestras, 4 por cada centímetro de distancia. Se puede ver una clara diferencia entre la figura 38 y la 40, teniendo la primera una forma de decaimiento más lineal, mientras que para 100 kHz decae de forma exponencial hasta llegar a los 3 cm, donde parece que tiende incluso a incrementar el voltaje que se recibe, esto no tiene sentido por lo que se puede deber a imprecisiones en las mediciones o interferencias con otras fuentes de luz. Finalmente, en la figura 42 vemos una comparación de las dos células a 1 kHz, pudiendo concluir que la célula 2 es más efectiva a la hora de captar luz.

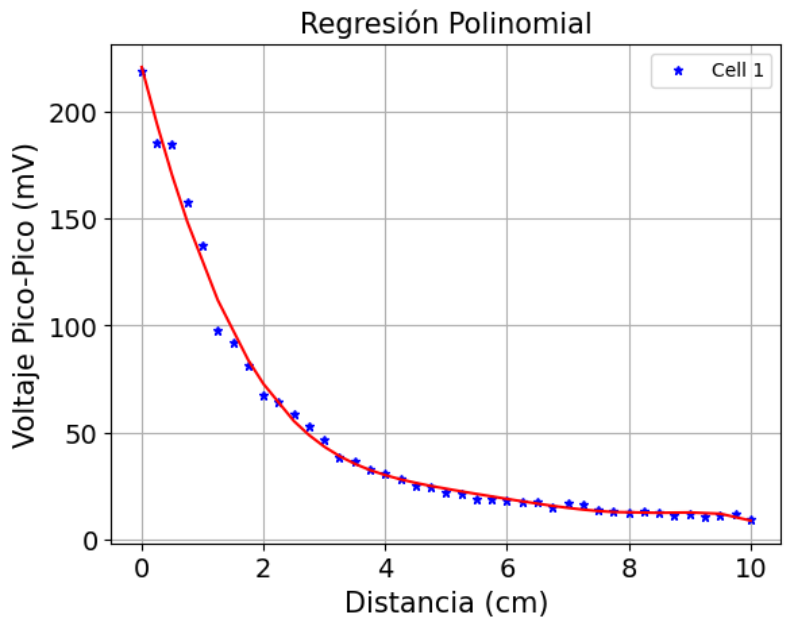


Figura 38. Regresión polinomial a 1 kHz.

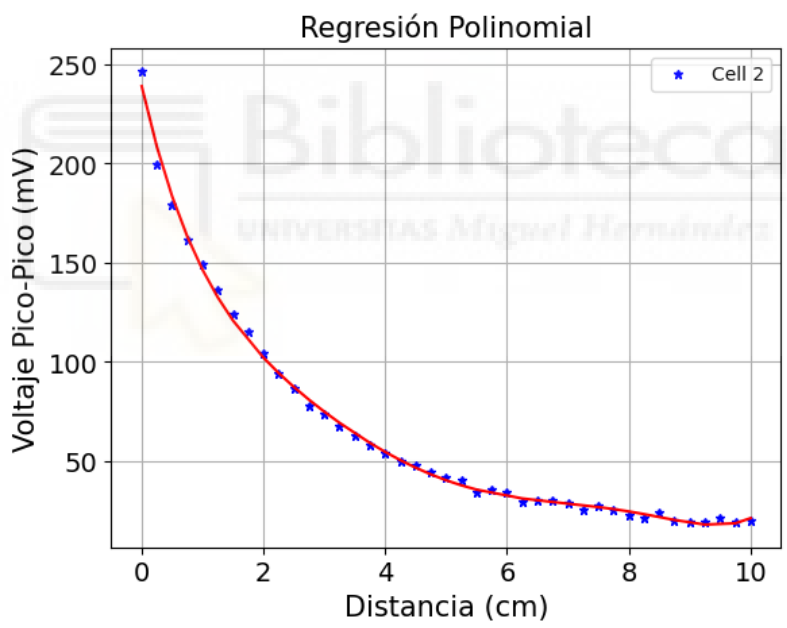


Figura 39. Regresión polinomial a 1 kHz.

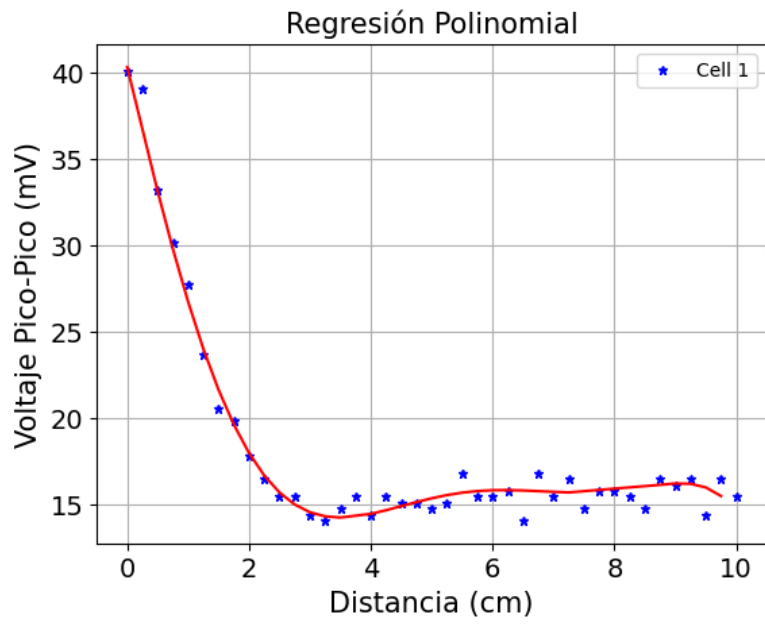


Figura 40. Regresión polinomial a 100 kHz.

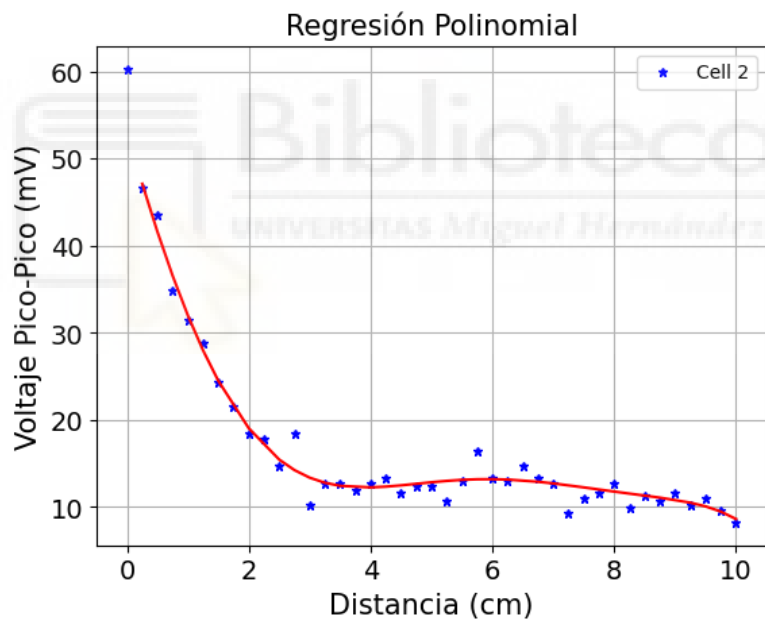


Figura 41. Regresión polinomial a 100 kHz.

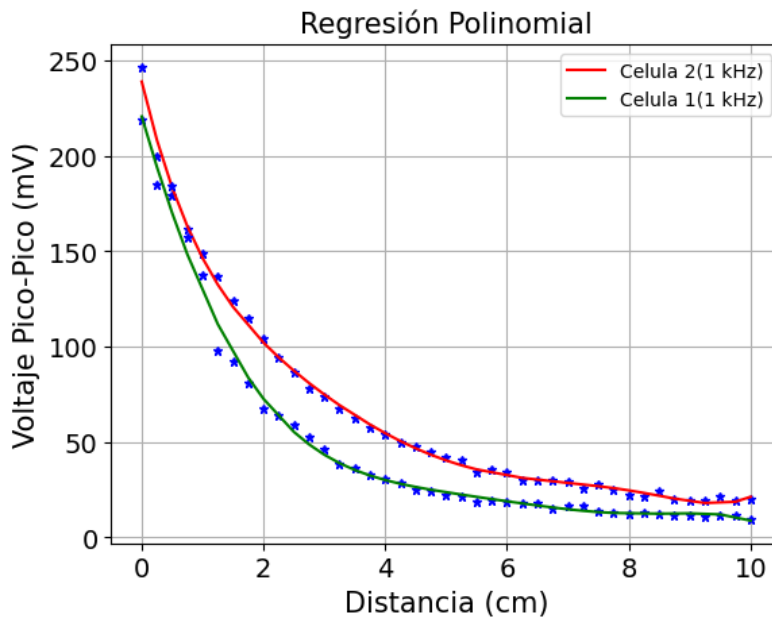


Figura 42. Regresión polinomial a 1 kHz.

Pese a que la célula 2 se comporta mejor para 1 kHz, en la figura 43 se ha realizado la comparación para 100 kHz, en los primeros 2 cm de medición, la célula 2 ofrece un mejor comportamiento, pero se puede ver que después, el comportamiento de la célula 1 es más eficiente.

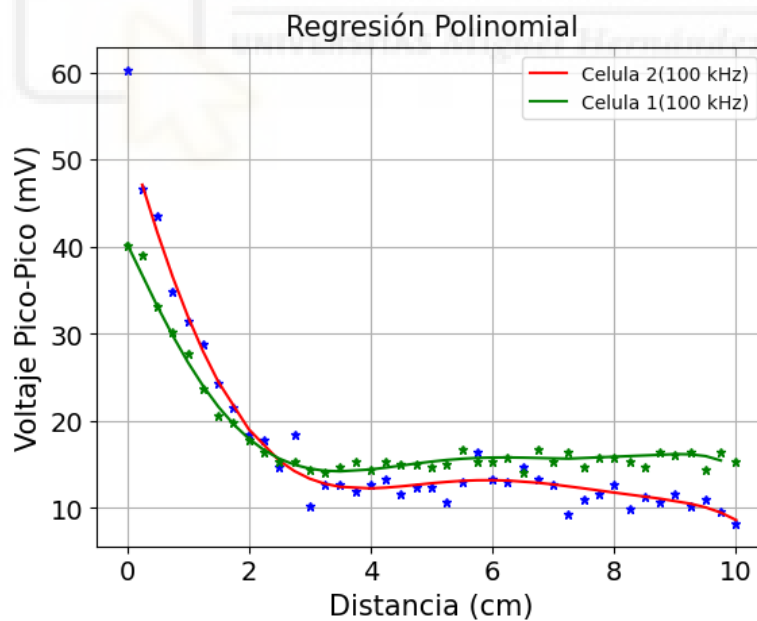


Figura 43. Regresión polinomial a 100 kHz.

Para terminar, se ha comparado los valores de voltaje en función de la distancia de las células 1 y 2, para valores de 1 y 100 kHz, obteniendo como resultado las figuras 44 y 45, donde se puede observar que, a partir de los 6 cm, los valores de voltaje son muy similares, por lo que solo es interesante trabajar a 1 kHz en distancias menores a 6 cm, que es donde la diferencia de voltaje es más notable.

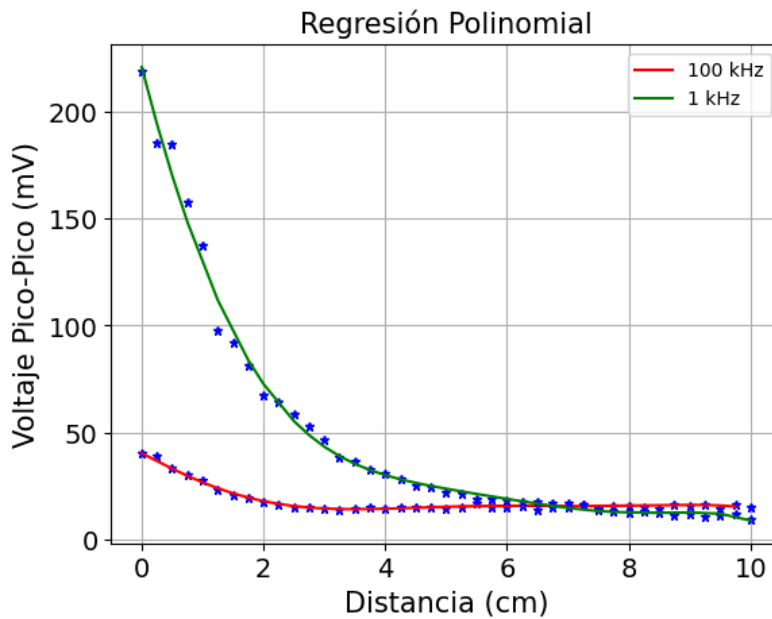


Figura 44. Comparación de regresión polinomial a 100 kHz y 1 kHz Cell 1.

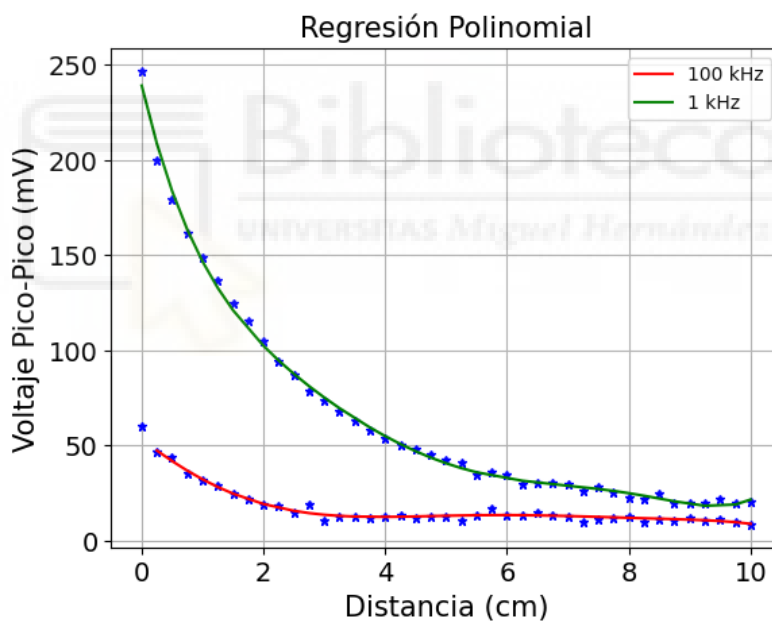


Figura 45. Comparación de regresión polinomial a 100 kHz y 1 kHz Cell 2.

4.3.2. COMPARACIÓN CÉLULAS THF, DMSO, DMF

A continuación, trabajaremos con las células THF, DMSO y DMF, que podemos ver en las figuras 46, 47 y 48 respectivamente, con su correspondiente fecha de producción. Para los tres tipos de fotodetectores, disponemos de 6 células, siendo la célula 1 la de referencia, que servirá para comparar el comportamiento del resto de células.

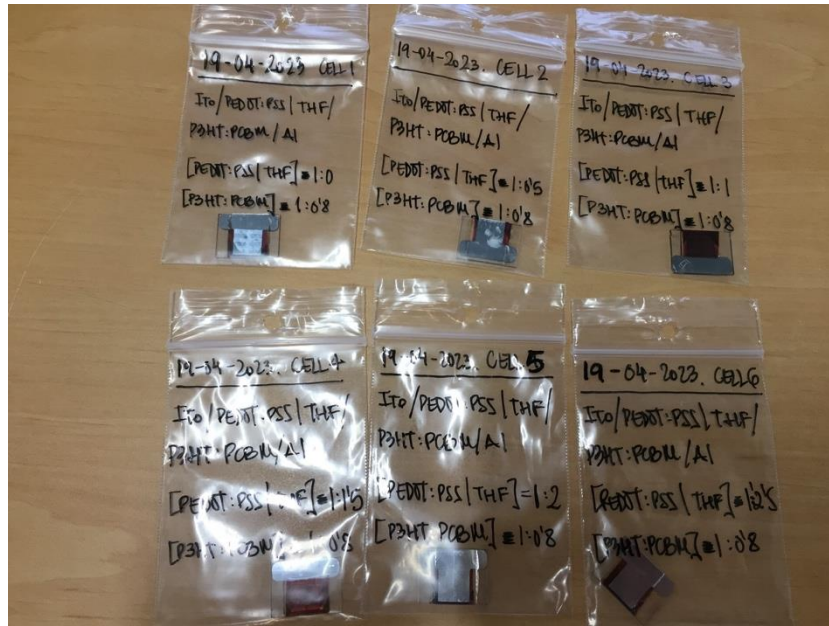


Figura 46. Células THF.

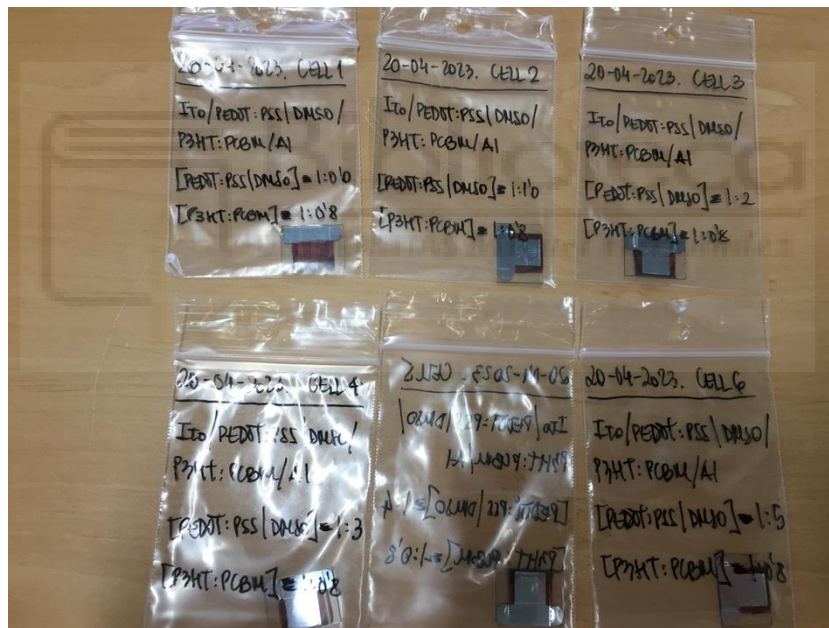


Figura 47. Células DMSO

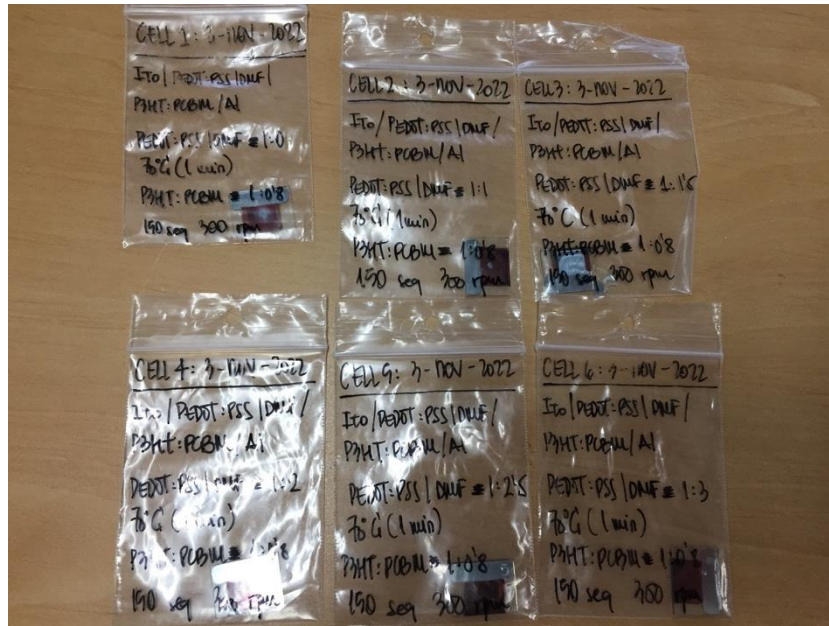


Figura 48. Células DMF

Vamos a realizar una comparativa de 6 células de un mismo tipo, por ejemplo, para las 6 células disponibles de THF, medimos manualmente cual es la celda en la que se mide una mayor potencia, y las comparamos entre si. Estas medidas se hacen variando tanto la frecuencia como la distancia.

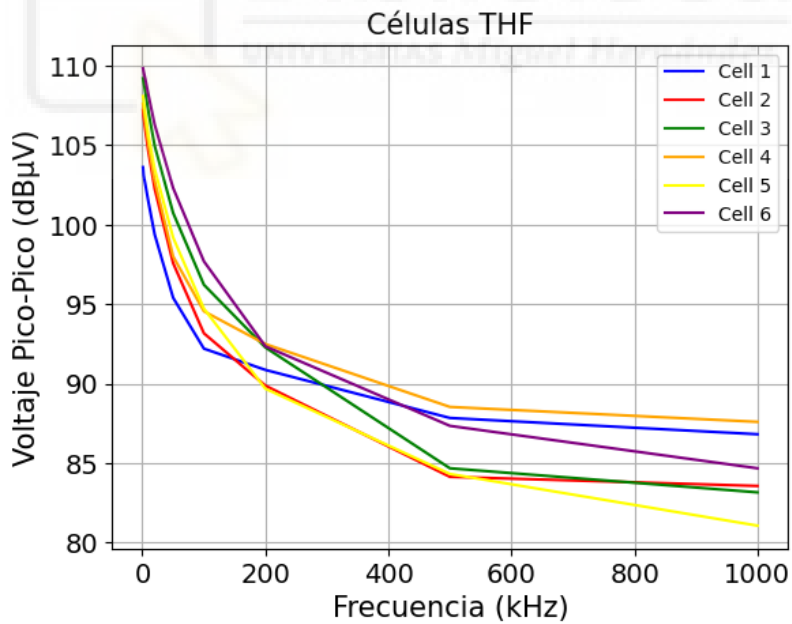


Figura 49. Comparación de todas las celdas de las células THF.

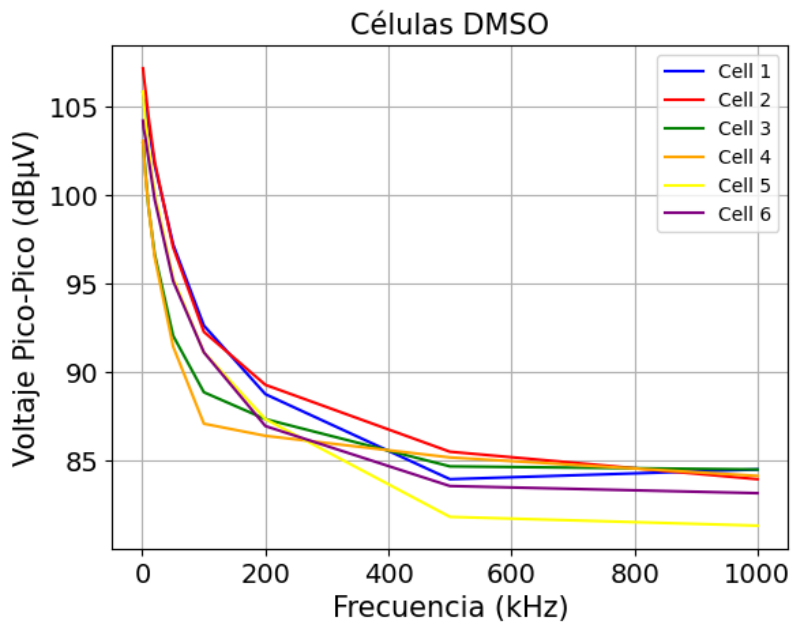


Figura 50. Comparación de todas las celdas de las células DMSO.

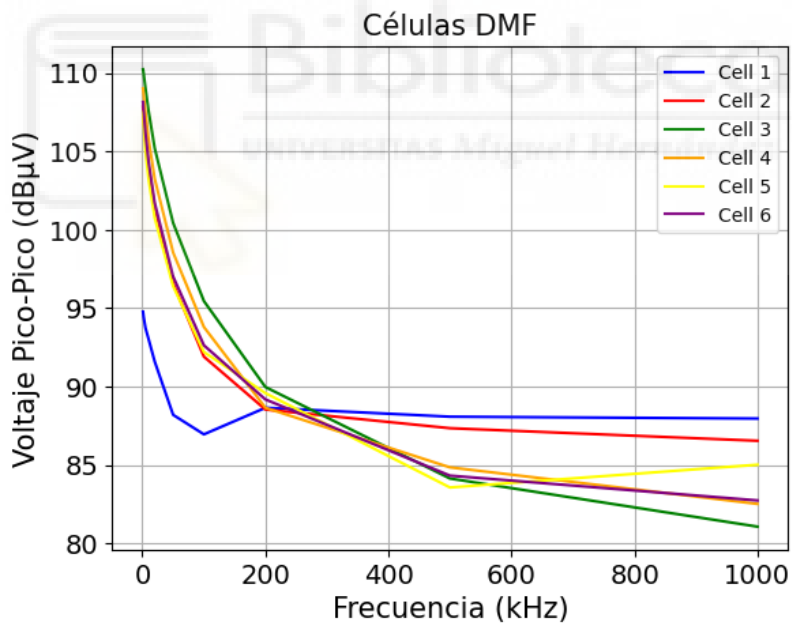


Figura 51. Comparación de todas las celdas de las células DMF.

Para las medidas en frecuencia, en THF vemos como la primera célula efectivamente empieza con peores valores que las demás, sin embargo, a medida que aumenta la frecuencia, su comportamiento mejora, llegando a situarse entre las que mejor respuesta obtienen a altas frecuencias, este patrón es similar para las DMSO, y para DMF se ve mucho más acentuado, puesto que los valores a baja frecuencia son muy bajos, y a partir de los 300 kHz, pese a que el comportamiento es muy lineal, la célula 1 es significativamente mejor que las demás.

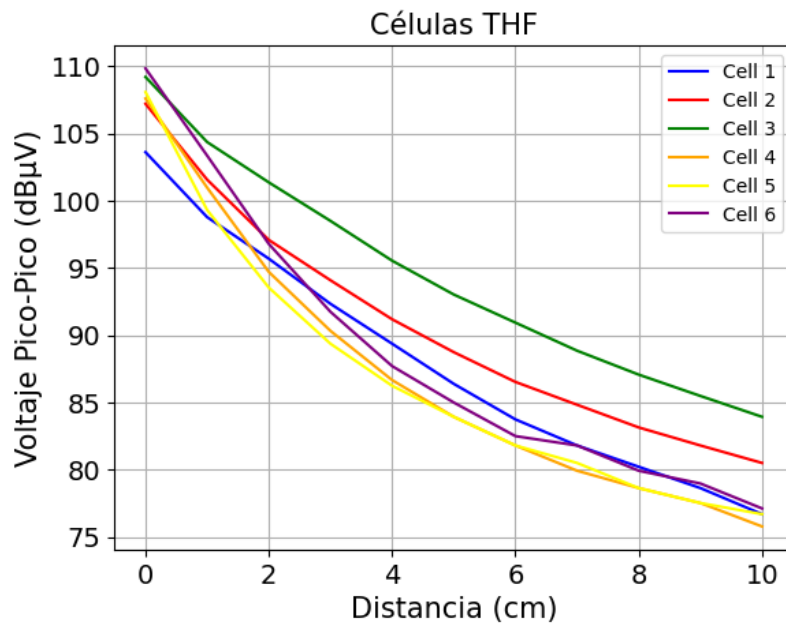


Figura 52. Comparación de todas las celdas de las células THF.

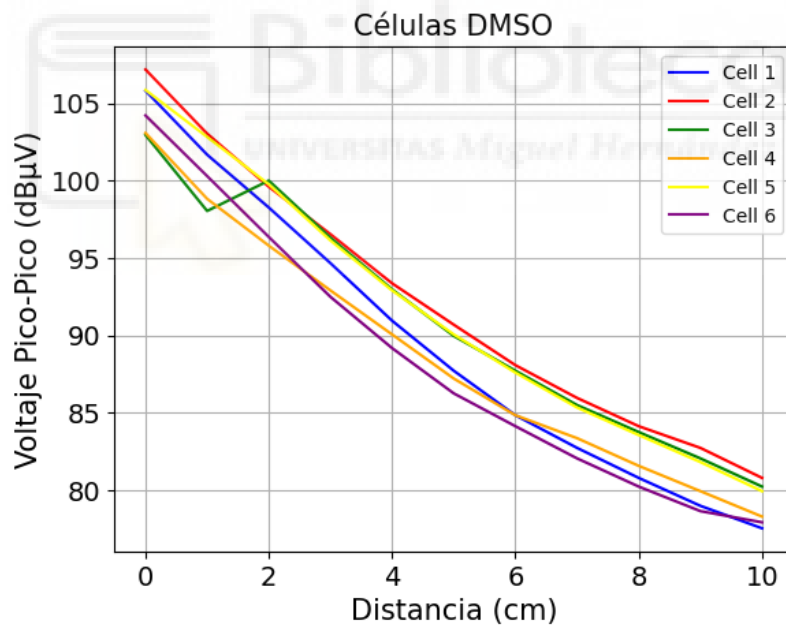


Figura 53. Comparación de todas las celdas de las células DMSO.

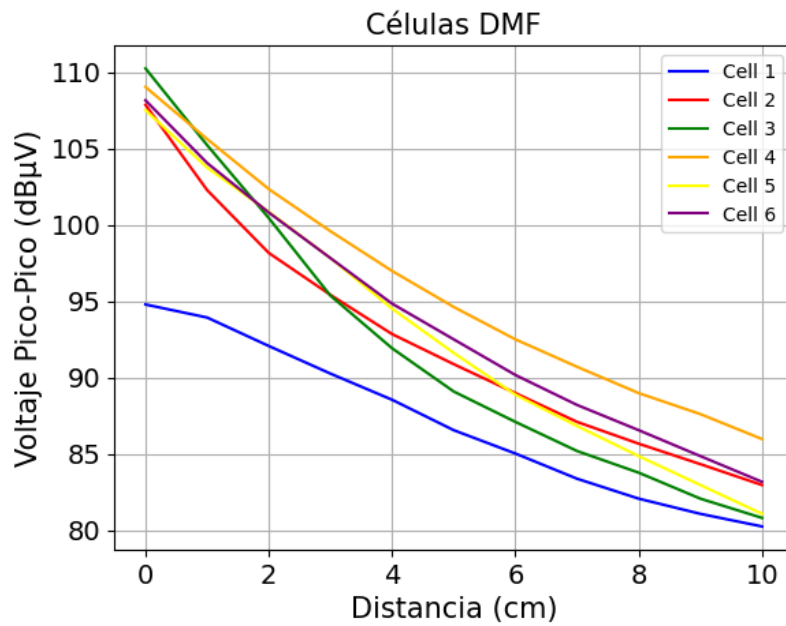


Figura 54. Comparación de todas las celdas de las células DMF.

Los valores en distancia difieren respecto a los de frecuencia, la única que tiene un comportamiento similar a las anteriores, es la figura 52, mientras que en las figuras 53 y 54, la célula 1 no mejora respecto a las demás, o incluso empeora.

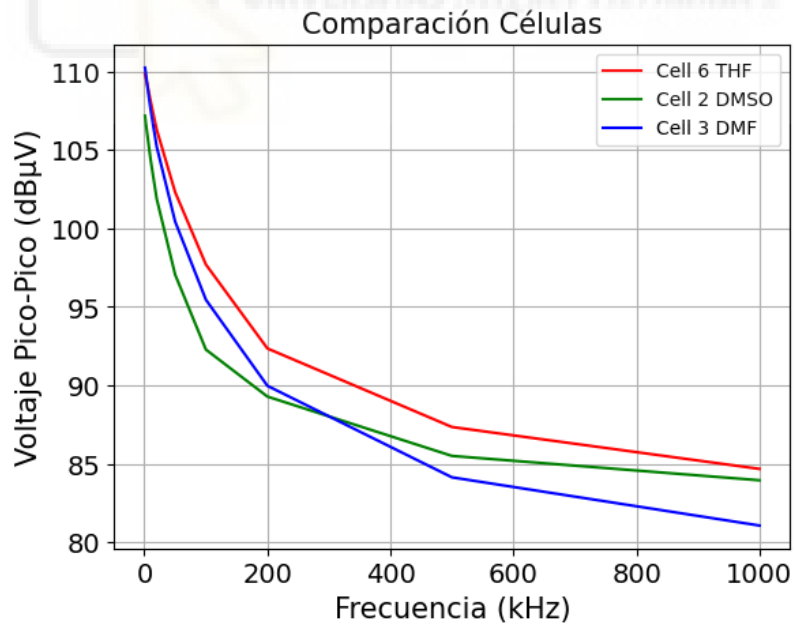


Figura 55. Comparación de las mejores células de cada tipo.

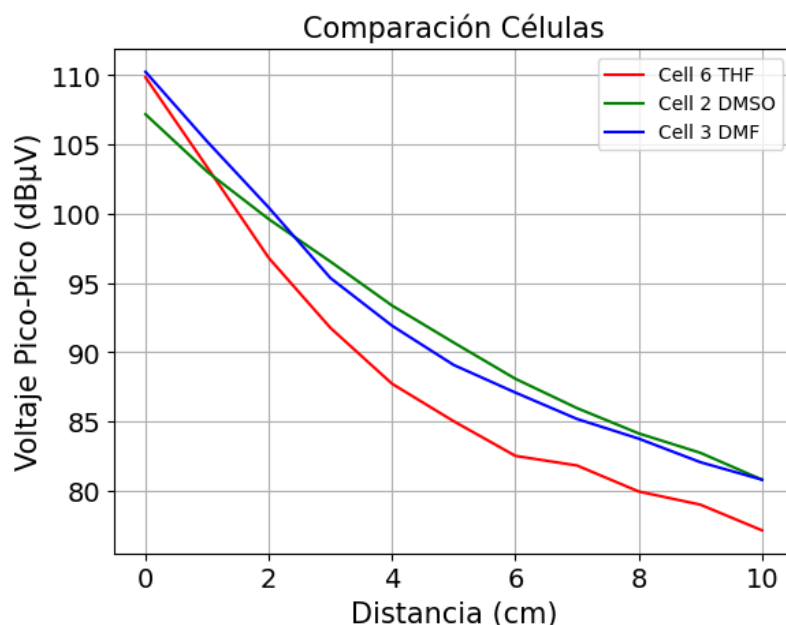


Figura 56. Comparación de las mejores células de cada tipo.

Por último, hacemos una comparación de las células que obtienen mejores valores a bajas frecuencias, vemos que en frecuencia la THF es claramente superior, mientras que en distancia es la peor, estando DMSO y DMF muy parejas. Esto lo podemos ver de manera más clara representado en las tablas 3 y 4.

	THF	DMSO	DMF
Célula 1	103.61 dBµV	105.80 dBµV	94.77 dBµV
Célula 2	107.21 dBµV	107.17 dBµV	107.84 dBµV
Célula 3	109.20 dBµV	102.94 dBµV	110.23 dBµV
Célula 4	107.60 dBµV	103.07 dBµV	109.01 dBµV
Célula 5	108.08 dBµV	105.85 dBµV	107.50 dBµV
Célula 6	109.84 dBµV	104.20 dBµV	108.14 dBµV

Tabla 3. Voltaje Pico-Pico a 1 kHz y 0 cm.

	THF	DMSO	DMF
Célula 1	76.71 dBµV	77.53 dBµV	80.23 dBµV
Célula 2	80.51 dBµV	80.79 dBµV	82.94 dBµV
Célula 3	83.94 dBµV	80.23 dBµV	80.79 dBµV
Célula 4	75.79 dBµV	78.29 dBµV	85.95 dBµV
Célula 5	76.71 dBµV	79.93 dBµV	81.06 dBµV
Célula 6	77.13 dBµV	77.92 dBµV	83.15 dBµV

Tabla 4. Voltaje Pico-Pico a 1 kHz y 10 cm.

Observando la tabla 4, vemos que los valores de voltaje son bastante variables según la célula, por ejemplo, la célula 3 para DMF empieza siendo la segunda con mejores valores y termina siendo la peor. Esto no debería ser así cuando medimos en función de la distancia, y se tiene que deber a errores a la hora de la medición de las células, como puede ser no haber enfocado correctamente el LED en determinadas distancias.

4.3.3. FILTRO DE MEDIANA Y GAUSSIANO APLICADO A CÉLULAS THF, DMSO Y DMF

Google Colaboratory tiene integradas muchas funciones que podemos usar para filtrar una determinada señal, con los filtrados buscamos obtener una señal con una mejor calidad, obteniendo valores de SNR y Q más altos y bajando la BER. Vamos a usar como ejemplo las células THF.

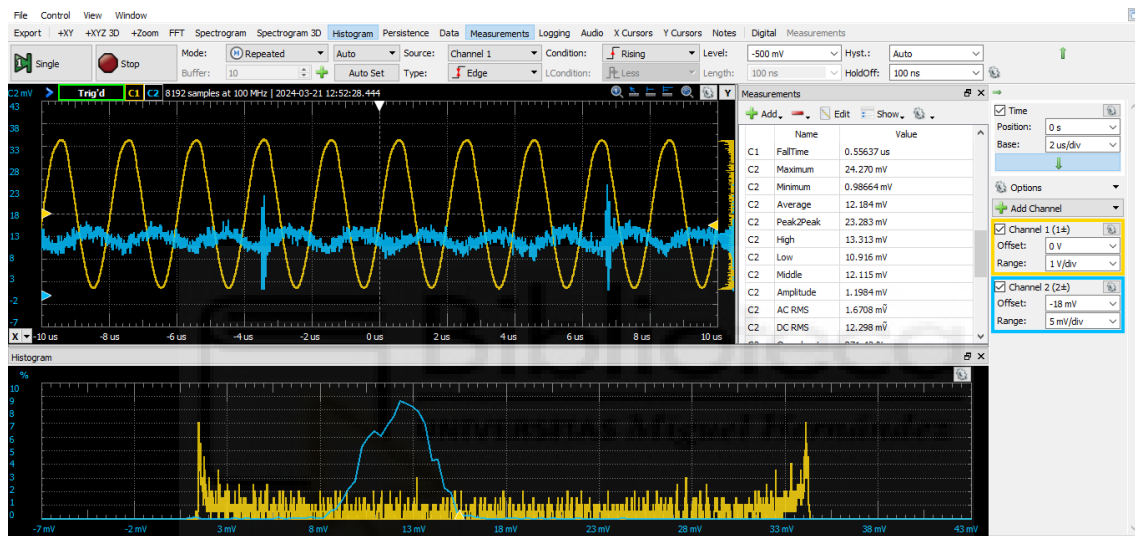


Figura 57. Osciloscopio WaveForms.

Encontramos una gran utilidad en el uso de este tipo de filtros sobre todo para las situaciones límite, donde el transmisor se encuentra a una frecuencia muy alta, entorno a los 500 kHz, o a grandes distancias, superiores a unos 7 cm. En la figura 57, podemos ver como la señal está especialmente degradada debido al ruido y los rizados que aparecen en determinados intervalos de tiempo, pero cuando usamos el filtro de mediana, vemos un aumento considerable en la calidad de la señal que recibimos. Además de ver una señal mucho más suavizada, donde el ruido ha disminuido considerablemente, vemos que los picos son menores, esto lo podemos corroborar utilizando la función “max” integrada en Google Colaboratory, obteniendo los siguientes resultados:

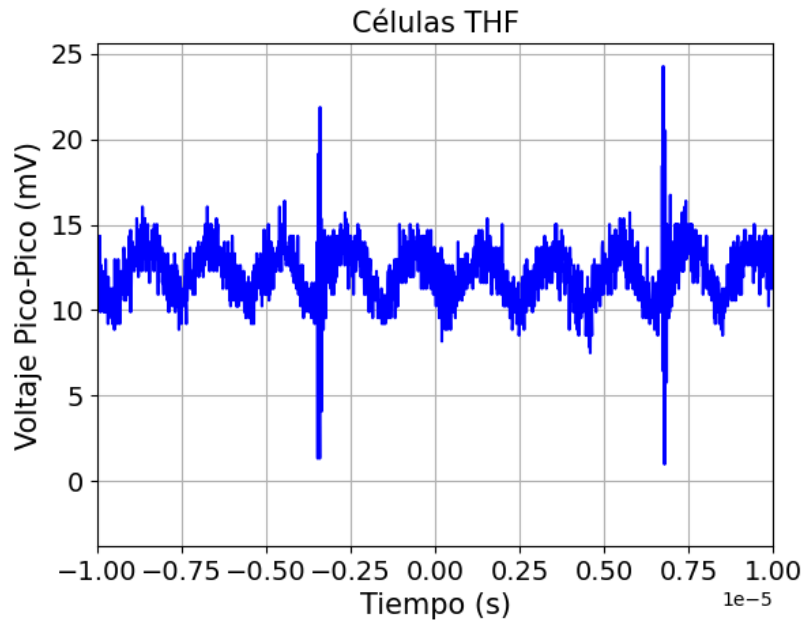


Figura 58. Señal de 500 kHz sin procesar.

Voltaje pico-pico estimado figura 59: 26.706931000000004

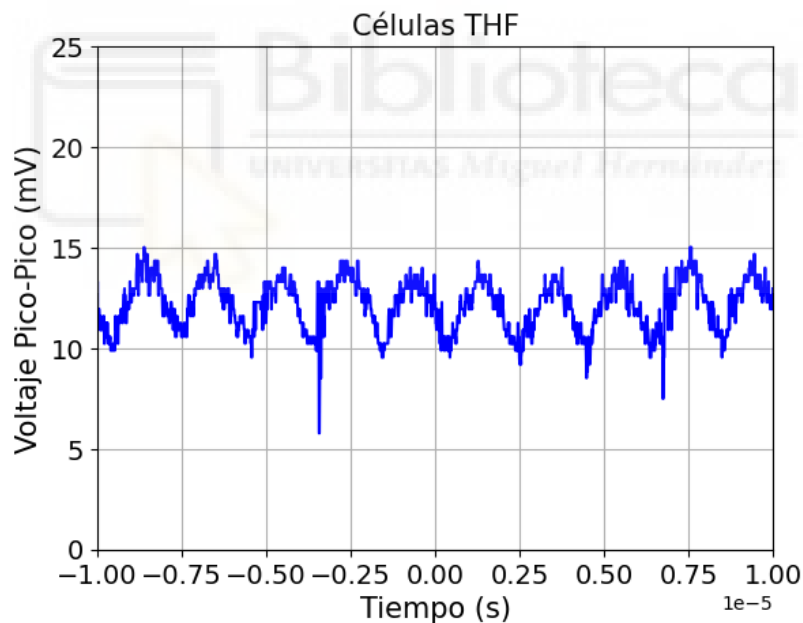


Figura 59. Señal procesada con filtro de mediana.

Voltaje pico-pico estimado figura 58: 12.668673000000002

Como vemos, el voltaje pico-pico que antes se media en la señal, no era un valor real, y venía determinado por los picos de voltaje que se ven en la figura 58, con el filtrado hemos conseguido obtener un valor 14 mV más bajo, no obstante, el valor de voltaje sigue sin ser un reflejo real de lo que se debería medir, esto lo podemos comprobar visualmente en la figura 59. Es por esto último que se hace necesario el uso de un mejor filtro, que mejore aún más la calidad de nuestra señal, para ello se utiliza un filtro Gaussiano, los resultados se reflejan en la figura 60:

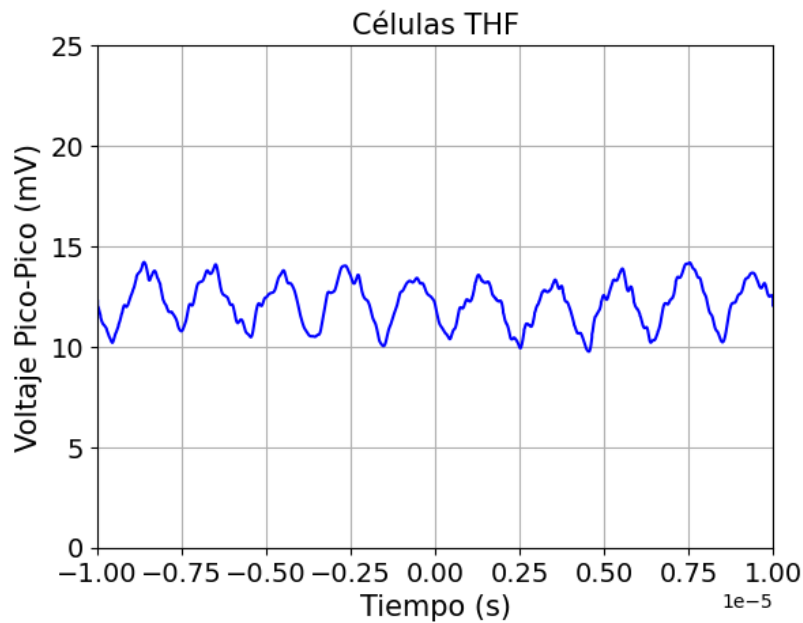


Figura 60. Señal procesada con filtro Gaussiano.

Voltaje pico-pico estimado figura 59: 4.796478826743835

Como se puede ver, el filtrado es mucho mejor, obteniendo valores de voltaje muy ajustados a lo que visualmente cabría esperar de la señal, los picos han desaparecido por completo y el ruido a disminuido mucho más en comparación con el filtro de mediana. Si nos fijamos en la figura 57, el valor de voltaje pico a pico es de 23,28 mV, valor que no encaja con lo medido una vez hemos procesado la señal en la figura 58, donde el valor de voltaje es de 26,70 mV, es decir, 3,42 mV superior. Esta diferencia se debe a que el software de Waveforms calcula todos los parámetros basándose en lo que muestra la ventana del osciloscopio, para nuestro caso, de -10 us a 10 us, por lo que únicamente cogerá los valores de voltaje de los dos rizados que observamos en la figura 57, mientras que para la figura 58, a pesar de que está representado la misma ventana, el valor de voltaje se está calculando mediante el array de valores que le pasamos al programa para representarlo. Adicionalmente, aplicamos los filtros para las células DMSO y DMF:

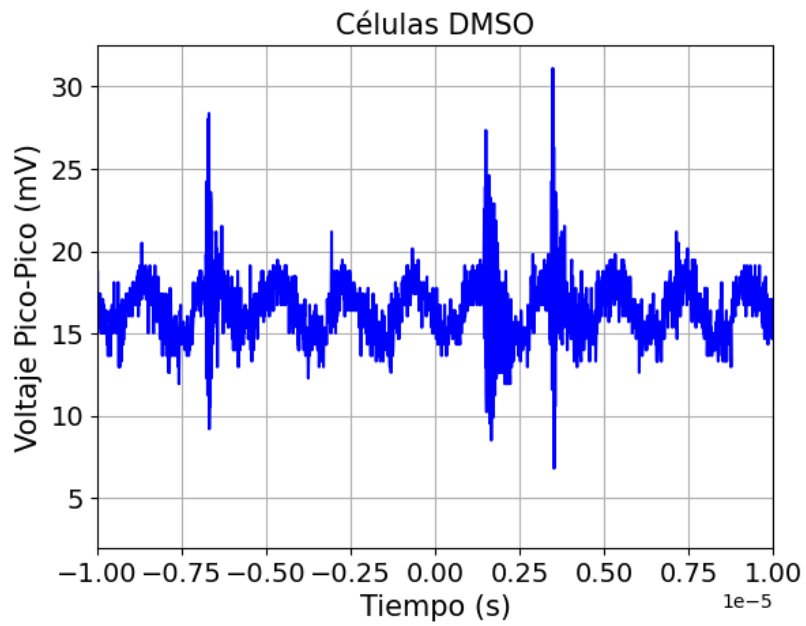


Figura 61. Señal de 500 kHz sin procesar.

Voltaje pico-pico 27.734121

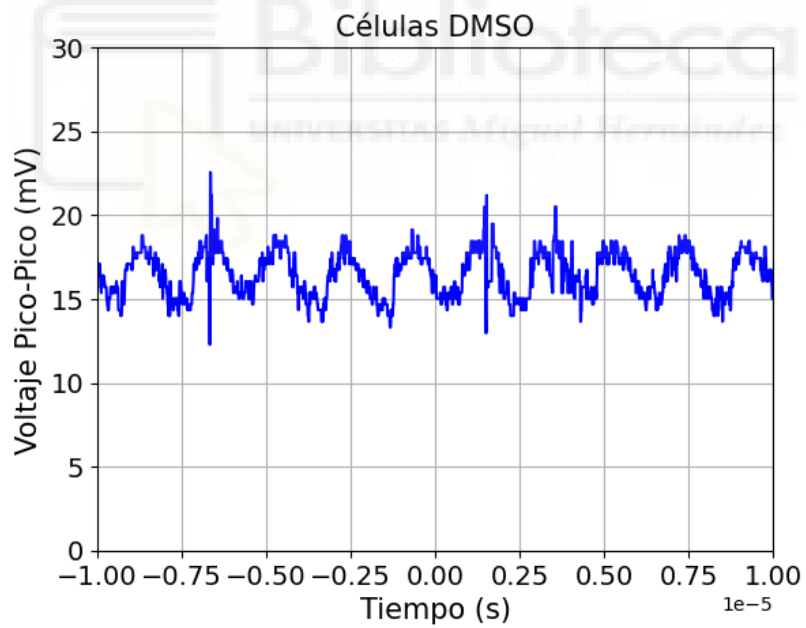


Figura 62. Señal procesada con filtro de mediana.

Voltaje pico-pico estimado: 11.983879000000002

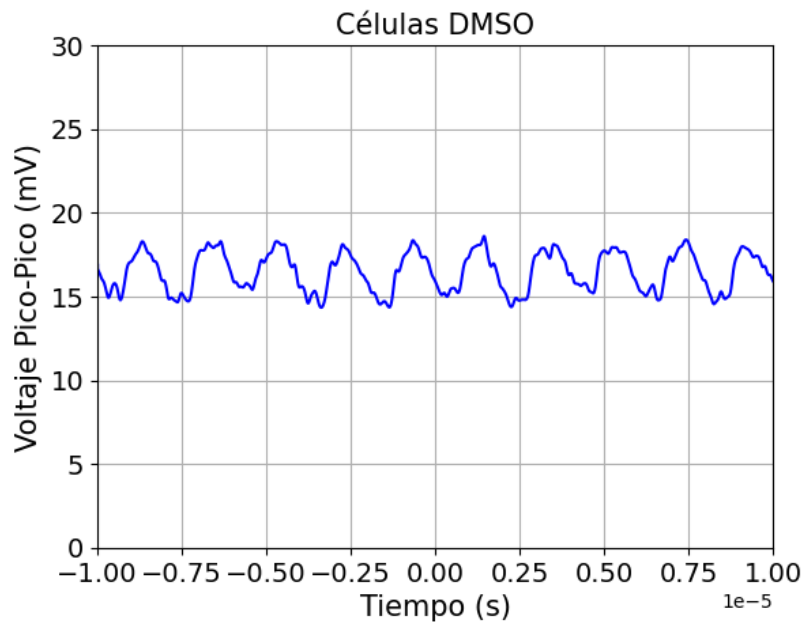


Figura 63. Señal procesada con filtro Gaussiano.

Voltaje pico-pico estimado: 5.368706494478223

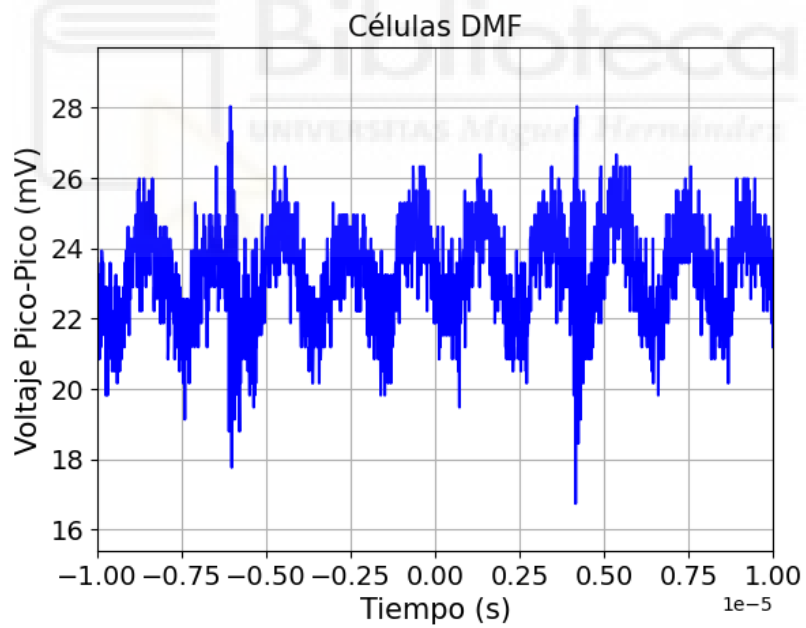


Figura 64. Señal de 500 kHz sin procesar.

Voltaje pico-pico 13.011068999999999

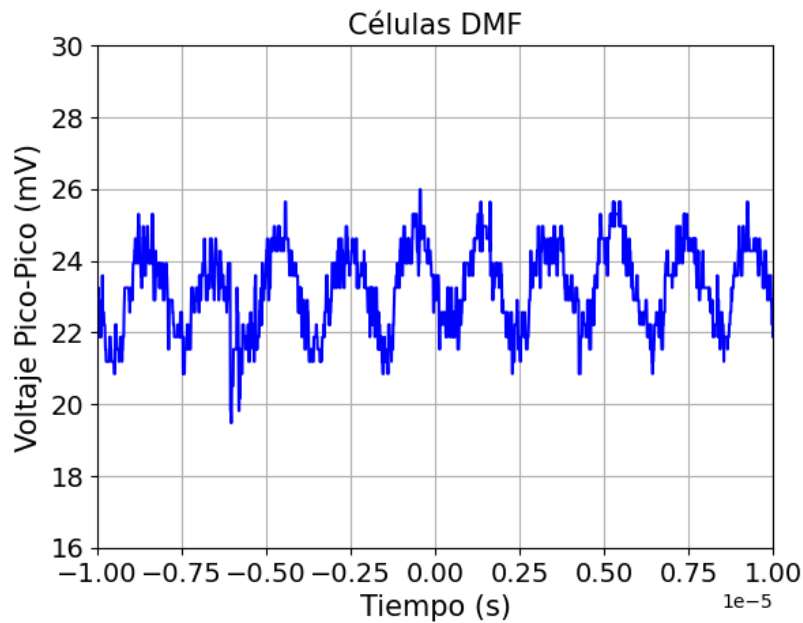


Figura 65. Señal procesada con filtro de mediana.

Voltaje pico-pico estimado: 6.505535000000002

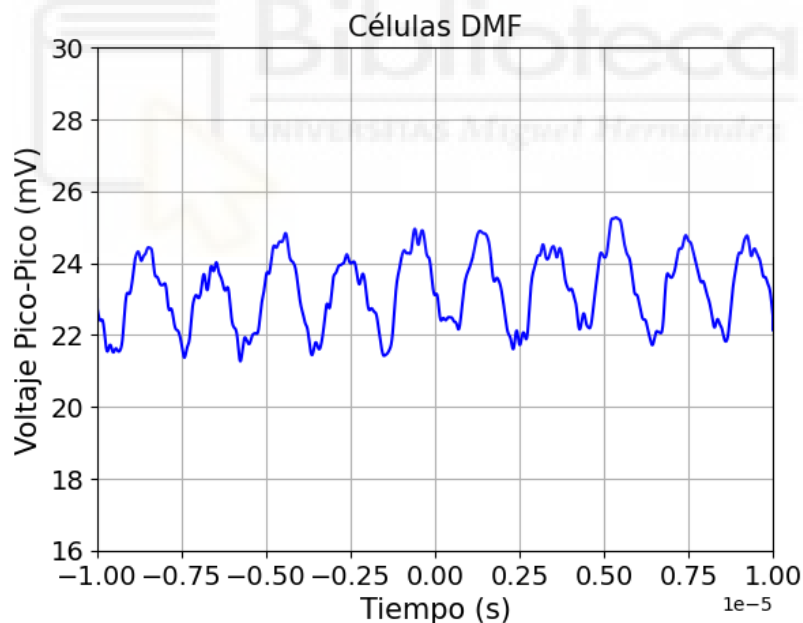


Figura 66. Señal procesada con filtro Gaussiano.

Voltaje pico-pico estimado: 4.527748080771353

Finalmente, podemos resumir todos los valores obtenidos en la tabla 5, donde vemos todos los valores de voltaje pico-pico para los tres tipos de células, con la señal sin filtrar y con los dos tipos de filtros que hemos visto. Además, se vuelve a observar el efecto que mencionamos anteriormente, ya que en las figuras 63 y 66, vemos valores que no parecen coincidir con el voltaje medido justo debajo de las figuras, nuevamente se debe a que los valores se calculan sobre el array total por lo que podemos suponer que

fuera de nuestra ventana de visualización habrá valores más altos y más bajos de lo que podemos ver.

	THF	DMSO	DMF
Señal sin filtrar	26,70 mV	27,73 mV	13,01 mV
Filtro de mediana	12,66 mV	11,98 mV	6,50 mV
Filtro Gaussiano	4,79 mV	5,36 mV	4,52 mV

Tabla 5. Valores de voltaje pico-pico para células THF, DMSO y DMF con diferentes filtros.

Para poder analizar los resultados de una mejor forma, se ha calculado la reducción del voltaje de forma porcentual, es decir, en las células DMF hemos obtenido un valor de 13,01 mV para la señal sin filtrar, y 4,52 mV para el filtro gaussiano, esto implica una reducción del 65,25% del voltaje total medido en la señal sin filtrar.

	THF	DMSO	DMF
Reducción filtro de mediana	52,58%	56,79%	50,03%
Reducción filtro Gaussiano	82,05%	80,67%	65,25%

Tabla 6. Reducción porcentual para filtro de mediana y Gaussiano para células THF, DMSO y DMF.

Como podemos ver en la tabla 6, para THF se produce una mayor reducción, llegando a ser hasta del 82,05%, seguida por DMSO y DMF. Las reducciones entre THF y DMSO son muy parecidas puesto que presentan valores de voltaje muy parecidos, mientras que en DMF la reducción es mucho menor debido a que el voltaje inicial es más bajo debido a que las células DMF no presentan rizados tan marcados como en las otras dos células, y el voltaje para las tres células en su mejor filtro no baja de unos 4,5 mV, siendo en los tres muy parecido.

CAPÍTULO 5. CONCLUSIÓN Y LINEAS FUTURAS

5.1. CONCLUSIÓN

En este trabajo, se ha explorado el uso de células fotovoltaicas orgánicas dopadas con diversos disolventes (DMF, THF y DMSO) en sistemas de comunicación Li-Fi, en comparación con células comerciales. A través de un enfoque basado en técnicas de aprendizaje automático mediante regresiones, se ha optimizado el proceso de transmisión y recepción de datos, lo que ha permitido una mejor comprensión de los parámetros críticos que afectan a la eficiencia y la calidad de la comunicación en estos sistemas.

En un principio, hemos utilizado regresiones lineales con 40 muestras, que se han aumentado posteriormente a 1000 para poder ver el comportamiento del sistema a frecuencias mayores, donde hemos necesitado utilizar regresiones polinomiales debido a la baja precisión que aportaban las lineales cuando la tendencia no es tan lineal. Además, hemos visto que el desempeño de las regresiones polinomiales es bastante bueno a partir del segundo grado, aunque para algunos casos necesitaremos que sean de grado 4 o incluso 5 dependiendo de que la tendencia sea más o menos lineal.

Primero hemos trabajado con fotodetectores comerciales, donde hemos estudiado sus tiempos de subida y bajada, así como los ciclos de trabajado, que nos han servido para observar que su comportamiento no es uniforme, si no que varía, habiendo un cambio de tendencia entre los tiempos de subida y bajada a los 2,19 kHz y entre el ciclo positivo y negativo a 856 kHz. Respecto a los parámetros secundarios SNR, Q y VER, hemos obtenido unos resultados muy buenos debido a que hemos tomado las medidas en condiciones ideales (0 cm y 1 kHz).

Después hemos pasado a trabajar con las células orgánicas, dopadas con diferentes sustratos, donde hemos visto que para todos los tipos de sustratos aparecen rizados en nuestra señal, y se ha propuesto diferentes maneras de reducirlo, como puede ser el ajuste manual del sistema, el aislamiento eléctrico o, la que resulta ser más eficiente, el uso de filtros mediante el procesamiento de la señal. Hemos conseguido mejoras respecto a la célula de referencia, que no se mantienen al aumentar la distancia y la frecuencia debido al rizado y a posibles imprecisiones en la toma de medidas, por lo que el filtro de mediana puede representar una solución a estos rizados. Al aplicar dicho filtro se ha observado que, si bien minimiza estos picos de señal, no los elimina. Estos rizados residuales, se han podido solucionar mediante el uso de filtros Gaussianos, ya que eliminan completamente los picos que surgen en la señal.

Los resultados obtenidos demuestran que las células orgánicas, especialmente aquellas dopadas con THF, podrían ser una alternativa interesante y más sostenible a los fotodetectores comerciales de silicio que ocupan una mayor cuota de mercado.

Asimismo, la implementación de modelos de regresión ha permitido predecir el comportamiento de los sistemas Li-Fi bajo diferentes condiciones operativas, lo que contribuye a una optimización más precisa de los mismos y un ahorro de tiempo a la hora de caracterizar dichos dispositivos. Las mediciones realizadas sobre los parámetros SNR, BER y Q han sido fundamentales para evaluar el desempeño global del sistema y proponer mejoras que podrían ser implementadas en futuros trabajos.

5.2. LINEAS FUTURAS

El presente trabajo abre varias oportunidades para investigaciones futuras en el ámbito de las comunicaciones mediante luz visible y Li-Fi. A continuación, se estudiarán cuales son las posibles líneas a estudiar en un futuro.

Continuar con la investigación de otros disolventes y materiales para dopar las células fotovoltaicas orgánicas debería ser la línea principal, ya que es la mayor limitante que encontramos en este tipo de sistemas, con el fin de mejorar su eficiencia y adaptabilidad a diferentes condiciones ambientales.

Realizar pruebas en escenarios más complejos, incluyendo ambientes con mayor interferencia lumínica y distancias de transmisión más largas, para evaluar la robustez de las células orgánicas en condiciones más exigentes.

Investigar el uso de algoritmos de aprendizaje automático más avanzados, como puede ser el uso de Deep Learning, para mejorar aún más la predicción del comportamiento del sistema y la optimización de los parámetros operativos.

Explorar la integración de sistemas Li-Fi con otras tecnologías de comunicación, como Wi-Fi, para desarrollar redes híbridas que puedan aprovechar las ventajas de ambas tecnologías.

Estas líneas futuras no solo ayudarán a mejorar la eficiencia y viabilidad de las comunicaciones Li-Fi, sino que también contribuirán al desarrollo de tecnologías de comunicación más sostenibles y adaptables a las necesidades futuras.



REFERENCIAS

- [1] Anurag Sarkar, Shalabh Agarwal, Asoke Nath, “Li-Fi Technology: Data Transmission through Visible Light”, *International Journal of Advance Research in Computer Science and Management*, Julio 2015.
- [2] Digilent. (s.f.). “Analog Discovery 2: 100MS/s USB Oscilloscope, Logic Analyzer and Variable Power Supply”. Disponible en: <https://digilent.com/>. [Último acceso: 12-03-2024].
- [3] Digilent. “WaveForms SDK Reference Manual”, Disponible en: <https://digilent.com/>. [Último acceso: 12-03-2024]
- [4] David Paper, Apress, “State-of-the-Art Deep Learning Models in TensorFlow”, Agosto 2021
- [5] Andreas C. Müller & Sarah Guido, O’Reilly Media, “Introduction to Machine Learning with Python, A GUIDE FOR DATA SCIENTISTS”, Noviembre 2016.
- [6] Yolanda Orenes Casanova, Tesis doctoral, Universidad Miguel Hernández, “Contribuciones al problema de clasificación en machine learning”, Octubre 2022.
- [7] Jorge García Brea, Trabajo de final de grado, Universidad Miguel Hernández, “Automatización de medidas para sistemas de comunicación por luz visible mediante células orgánicas fotovoltaicas”, Junio 2022.
- [8] Fernando Rodríguez Mas, Pablo Corral González, “Protocolo para las Mediciones de los Experimentos de sistemas VLC / Li-Fi”, Universidad Miguel Hernández
- [9] Maxim Integrated. (2008). “Optical Signal-to-Noise Ratio and the Q-Factor in Fiber-Optic Communication Systems”. Disponible en: <https://www.maximintegrated.com/>.
- [10] RF Café-Tech News & Resources. (s.f.). “Bit Error Rate”. Disponible en: <https://www.rfcafe.com>.