



CONFIGURACIÓN DEL PATRÓN DE FOCALIZACIÓN EN LENTES DE ULTRASONIDOS

Rubén Pérez Bernabeu

Tutora: Pilar Candelas Valiente
Cotutora: Constanza Rubio Michavila

Trabajo Fin de Grado presentado en la Escuela
Técnica Superior de Ingenieros de
Telecomunicación de la Universitat
Politècnica de València, para la obtención del
Título de Graduado en Ingeniería de
Tecnologías y Servicios de Telecomunicación

Curso 2017-18
València, 26 de junio de 2018



Resumen

Este proyecto se enmarca dentro de las tareas propias de una Beca Colaboración del Ministerio de Educación. En él, se ha realizado un estudio acerca de las lentes Fresnel y su posible aplicación en el entorno de la medicina mediante el uso de ondas de ultrasonido. Este trabajo empieza con un desarrollo teórico de las ecuaciones que definen el comportamiento de lentes Fresnel ante ondas esféricas y el posterior diseño de las características de la lente utilizada a lo largo del proyecto. Seguidamente, también se han realizado simulaciones para demostrar el correcto funcionamiento del experimento y desarrollar un sistema de configuración del patrón de focalización en lentes de ultrasonidos. Finalmente, con la ayuda de un sistema acuático robotizado diseñado en el laboratorio de Física de la ETSIT y diversos transductores, se han realizado distintas medidas para tratar de demostrar experimentalmente la eficacia de nuestra idea. A lo largo de los tres bloques claramente diferenciados: modelo teórico, simulaciones y experimentos, ha sido necesario aplicar diversos conceptos relacionados con el procesado de señal y diseño de aplicaciones mediante Matlab, control remoto de instrumentos electrónicos, y uso de Comsol Multiphysics.

Resum

Aquest projecte s'emmarca dins de les tasques pròpies d'una Beca Col·laboració del Ministeri d'Educació. En ell, s'ha realitzat un estudi sobre les lents Fresnel i la seua possible aplicació en l'entorn de la medicina mitjançant l'ús d'ones d'ultrasò. Aquest treball comença amb un desenvolupament teòric de les equacions que defineixen el comportament de lents Fresnel davant ones esfèriques i el posterior disseny de les característiques de la lent utilitzada al llarg del projecte. Seguidament, també s'han realitzat simulacions per demostrar el correcte funcionament de l'experiment i dissenyar un sistema de configuració del patró de focalització en lents d'ultrasò. Finalment, amb l'ajuda d'un sistema aquàtic robotitzat dissenyat al laboratori de Física de l'ETSIT i diversos transductors, s'han realitzat diferents mesures per intentar demostrar experimentalment l'eficàcia de la nostra idea. Al llarg dels tres blocs clarament diferenciats: model teòric, simulacions i experiments, ha estat necessari aplicar diversos conceptes relacionats amb el processat de senyal i disseny d'aplicacions mitjançant Matlab, control remot d'instruments electrònics, i ús de Comsol Multiphysics.

Abstract

This project is part of the main tasks of a Collaboration Grant of the Ministry of Education. A study has been made about Fresnel lenses and their possible application in the field of medicine through the use of ultrasound waves. This work begins with a theoretical development of the equations that define the behavior of Fresnel lenses with spherical waves and the subsequent design of the characteristics of the lens used throughout the project. Moreover, simulations have also been carried out to demonstrate the correct functioning of the experiment and develop a configuration system of focalization patterns in ultrasonic lenses. Finally, using a robotic aquatic system designed in the ETSIT Physics laboratory and various transducers, different measures have been done to demonstrate experimentally the effectiveness of our idea. Throughout the three clearly differentiated blocks: theoretical model, simulations and experiments, it has been necessary to apply various concepts related to signal processing and Matlab application design, remote control of electronic instruments, and use of Comsol Multiphysics.



ÍNDICE DE CONTENIDOS

Capítulo 1.	Introducción	6
Capítulo 2.	Modelo teórico	7
2.1	Presentación e historia	7
2.2	Fundamentos físicos de una lente Fresnel	8
Capítulo 3.	Simulaciones con Comsol Multiphysics	11
3.1	Descripción del método de los elementos finitos	11
3.2	Definición del modelo de simulación	11
3.3	Método de procesado de las simulaciones	12
3.4	Aplicación de una señal de 250 kHz	16
3.5	Simulaciones utilizando varias frecuencias a la vez	17
3.6	Aplicación de dos señales de 240 kHz y 275 kHz simultáneamente	19
Capítulo 4.	Resultados experimentales	21
4.1	Sistema original Arquímedes	21
4.1.1	Robot y control del movimiento	21
4.1.2	Sistema de adquisición de datos	23
4.1.3	Transductores y amplificadores	23
4.2	Preparación del experimento	26
4.2.1	Conexión de los equipos	26
4.2.2	Preparación del experimento	26
4.2.3	Obtención de los datos	28
4.2.4	Conversión de muestras del formato 3Dreams a Matlab	29
4.2.5	Procesado básico de las medidas	30
4.3	Desarrollo de una aplicación en Matlab para controlar de manera remota el generador de funciones Agilent 33220A	32
4.4	Primera medida experimental: Emisión de una señal sinusoidal de 250kHz	35
4.4.1	Programación del generador de funciones remotamente	35
4.4.2	Realización de la medida	36
4.4.3	Procesado de las medidas	36
4.5	Segunda medida experimental: Emisión de dos señales sinusoidales de 240 kHz y 275 kHz	38
4.5.1	Programación del generador de funciones remotamente	38
4.5.2	Realización de la medida	39
4.5.3	Procesado de las medidas	39
4.6	Comparación de los resultados de las simulaciones y las medidas experimentales	41
Capítulo 5.	Conclusiones y líneas futuras	43
Capítulo 6.	Agradecimientos	44
Capítulo 7.	Bibliografía	45



Capítulo 8.	Anexos	47
8.1	Procesado de simulaciones de Comsol Multiphysics con App Designer de Matlab.	47
8.2	Control remoto del generador de funciones Agilent 33220A con App Designer	52
8.3	Procesado de medidas experimentales realizadas con App Designer	57

ÍNDICE DE FIGURAS

Figura 1.	Diagrama ilustrativo del espectro de frecuencias y campos de aplicación.	6
Figura 2.	Principales parámetros físicos del funcionamiento de una lente de Fresnel	8
Figura 3.	Lente Fresnel utilizada en los experimentos del proyecto	10
Figura 4.	Esquema del experimento simulado en Comsol Multiphysics	12
Figura 5.	Aplicación para procesar simulaciones de Comsol Multiphysics	13
Figura 6.	Planos de presión absoluta (Pa) simulados en Comsol Multiphysics utilizando frecuencias de 275 kHz (arriba), 250 kHz (centro) y 225 kHz (abajo)	14
Figura 7.	Gráficas de presión acústica frente a distancia longitudinal a 275 kHz (izquierda), 250 kHz (centro) y 225 kHz (derecha)	15
Figura 8.	Gráfica distancia focal vs frecuencia de operación	15
Figura 9.	Planos de presión con lente (izquierda) y sin lente (derecha) en Pascales tras aplicar una señal sinusoidal de 250 kHz	16
Figura 10.	Plano de ganancia en dB tras simular una señal sinusoidal de 250 kHz	16
Figura 11.	Corte longitudinal (izquierda) y corte transversal (derecha) de la Figura 9a tras simular una señal sinusoidal de 250 kHz	17
Figura 12.	Gráficas de presión acústica frente a la distancia longitudinal superpuestas	18
Figura 13.	Gráfica de la suma de señales de 250 kHz, 275 kHz y 225 kHz para conformar un único foco	18
Figura 14.	Gráfica de la suma de señales de 240 kHz, 250 kHz y 260 kHz para conformar un único foco más intenso	19
Figura 15.	Planos de presión con lente (izquierda) y sin lente (derecha) en Pascales tras simular una suma de señales sinusoidales de 240 kHz y 275 kHz	19
Figura 16.	Ganancia en dB tras simular una suma de señales sinusoidales de 240 kHz y 275 kHz	20
Figura 17.	Corte longitudinal (izquierda) y corte transversal (derecha) de la Figura 15a tras simular una suma de señales sinusoidales de 240 kHz y 275 kHz	20
Figura 18.	Esquema del robot de medida	22
Figura 19.	Influencia del espesor del transductor en la frecuencia de resonancia	23
Figura 20.	Factor de calidad en un transductor de ultrasonidos	24
Figura 21.	Fotografía del hidrófono y el preamplificador empleados	24
Figura 22.	Función de transferencia del transductor Inmasonic 250 kHz	25
Figura 23.	Esquema de conexiones del sistema completo de medidas	25



Figura 24. Script Matlab para crear un plano de medida	26
Figura 25. Contenido de un fichero de puntos de medida	27
Figura 26. Ejemplo de plano de medida	27
Figura 27. Captura del menú de calibración del programa de control del robot.....	28
Figura 28. Captura del menú principal del programa de control del robot donde se visualiza el plano de medida	29
Figura 29. Captura del programa utilizado para convertir archivos de formato .3dr a .mat	29
Figura 30. Aplicación desarrollada para procesar las medidas del sistema Arquímedes.....	30
Figura 31. Esquema del cambio de ejes propuesto para facilitar la interpretación de los resultados	31
Figura 32. Aplicación desarrollada para controlar el generador Agilent 33220A remotamente .	33
Figura 33. Señal generada por el Pulser 5077PR.....	34
Figura 34. Captura de la aplicación para controlar el generador de funciones tras enviar una señal sinusoidal de 250 kHz y 20 Vpp	35
Figura 35. Planos de presión con lente (izquierda) y sin lente (derecha) en Pascales tras aplicar una señal sinusoidal de 250 kHz	36
Figura 36. Plano de ganancia en dB tras aplicar una señal sinusoidal de 250 kHz	37
Figura 37. Corte longitudinal (izquierda) y corte transversal (derecha) tras aplicar una señal sinusoidal de 250 kHz	37
Figura 38. Captura de la aplicación para controlar el generador de funciones tras enviar una suma de señales sinusoidales de 240 kHz y 10 Vpp, y 275 kHz y 8 Vpp.....	38
Figura 39. Planos de presión con lente (izquierda) y sin lente (derecha) en Pascales tras aplicar una suma de señales sinusoidales de 240 kHz y 275 kHz	39
Figura 40. Plano de ganancia en dB tras aplicar una suma de señales sinusoidales de 240 kHz y 275 kHz	40
Figura 41. Corte longitudinal (izquierda) y corte transversal (derecha) tras aplicar una suma de señales sinusoidales de 240 kHz y 275 kHz	40
Figura 42. Cortes longitudinales al aplicar una señal de 250 kHz (izquierda) y dos señales simultáneas de 240 kHz y 275 kHz (derecha) superponiendo resultados experimentales (azul) y simulaciones (rojo).....	42
Figura 43. Cortes transversales al aplicar una señal de 250 kHz (izquierda) y dos señales simultáneas de 240 kHz y 275 kHz (derecha) superponiendo resultados experimentales (azul) y simulaciones (rojo).....	42

ÍNDICE DE TABLAS

Tabla 1. Radios de la lente Fresnel diseñada	9
Tabla 2. Plano y parámetros de la medida con una señal sinusoidal a 250 kHz.....	36
Tabla 3. Plano y parámetros de la medida con dos señales sinusoidales a 240 kHz y 275 kHz..	39
Tabla 4. Comparativa de los resultados de las simulaciones y experimentos.....	41

Capítulo 1. Introducción

La acústica estudia el sonido, infrasonido y ultrasonido, es decir, ondas mecánicas que se producen, transmiten e interactúan en un medio. En la actualidad existen diversos campos dentro de la acústica según se puede observar en la Figura 1, entre los cuales se sitúa el estudio de los ultrasonidos. Uno de los medios de propagación de los ultrasonidos suele ser, por las características de propagación de dichas ondas, el agua. Por tanto, es de sumo interés el desarrollo de modelos teóricos y experimentales que permitan la comprensión de los fenómenos físicos que se producen por la interacción de los ultrasonidos con el medio acuático y los dispositivos implementados, para el estudio de sus posibles aplicaciones tecnológicas y médicas. [1]

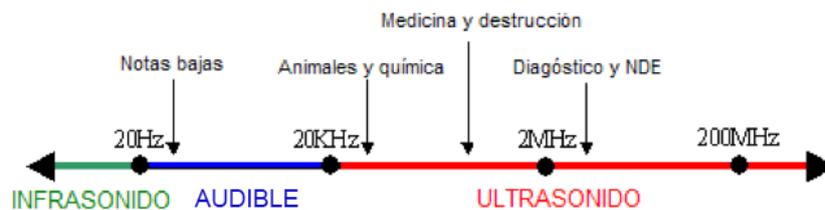


Figura 1. Diagrama ilustrativo del espectro de frecuencias y campos de aplicación [2]

Dentro de la medicina por ultrasonidos se pueden distinguir dos aplicaciones completamente distintas: [3]

- El ultrasonido de diagnóstico es una técnica no invasiva que se utiliza para producir imágenes procedentes del interior del cuerpo. Las sondas de ultrasonido, llamadas transductores, producen ondas sonoras que tienen frecuencias por arriba del umbral del oído humano (más de 20 kHz), aunque la mayoría de los transductores en uso actual operan a frecuencias mucho más altas en el rango de megahercios. La mayoría de las sondas de ultrasonido de diagnóstico se posicionan sobre la piel. Sin embargo, para mejorar la calidad de las imágenes, las sondas pueden colocarse dentro del cuerpo a través del tracto digestivo o la vagina. Además, en ocasiones se utiliza el ultrasonido durante la cirugía como técnica de visualización mediante la colocación de una sonda en la zona donde se efectúa la operación.
- El ultrasonido terapéutico también utiliza ondas sonoras por encima del rango del oído humano, pero no produce imágenes. Su objetivo es interactuar con los tejidos del cuerpo para que puedan ser modificados o destruidos. Entre las modificaciones posibles están: mover o calentar el tejido, disolver los coágulos, o administrar fármacos a sitios específicos en el cuerpo. Estas funciones son posibles mediante el uso de ondas de muy alta intensidad que pueden destruir los tejidos enfermos o anormales como los tumores. La ventaja de utilizar terapias de ultrasonido es que, en la mayoría de los casos, no son invasivas. Como consecuencia, no se necesita realizar cortes o incisiones en la piel, de manera que no quedan heridas o cicatrices.

Una de las subclases de ultrasonido terapéutico utiliza haces de sonido de alta intensidad que están muy bien orientados y se le llama Ultrasonido Focalizado de Alta Intensidad (HIFU por sus siglas en inglés). Actualmente, el HIFU está siendo investigado como un método para modificar o destruir los tejidos enfermos o anormales dentro del cuerpo sin tener que abrir la piel y ocasionar daños al tejido circundante.

Así pues, nuestro Trabajo Fin de Grado estaría incluido dentro de la aplicación de ultrasonidos terapéuticos. En él, utilizando los medios disponibles en la UPV, hemos desarrollado un modelo teórico, simulaciones y pruebas experimentales sobre una técnica para conseguir un patrón de focalización de ultrasonidos con lentes Fresnel en un medio acuático.



Capítulo 2. Modelo teórico

2.1 Presentación e historia

La focalización de ondas ha atraído siempre el interés científico por sus numerosas aplicaciones en diferentes áreas como óptica, microondas, acústica y, según hemos visto en el apartado anterior, la medicina. Para conseguir la focalización de ondas, pueden utilizarse mecanismos basados tanto en refracción como en difracción. Las lentes de Fresnel (FZPs) focalizan ondas a través de interferencias constructivas de campos difractados y son utilizadas en situaciones donde las lentes convergentes que consiguen la focalización por medio de mecanismos de refracción son difíciles de implementar. [4,5,6] Las FZPs pueden ser clasificadas en dos grandes grupos basados en su eficiencia de transmisión. [7] Por un lado, las FZPs Soret se implementan alternando zonas transparentes y opacas de Fresnel, mientras que las FZP de Rayleigh-Wood reemplazan las zonas opacas por zonas de inversión de fase, aumentando la eficacia de FZP por un factor de cuatro. [8] Las eficiencias de difracción, es decir, las relaciones entre las potencias difractadas y las potencias incidentes, para una lente Fresnel tipo Soret y una tipo Rayleigh-Wood son 10.1% y 40.5%, respectivamente. [1] En este trabajo hemos utilizado FZP tipo Soret debido a su mayor facilidad de fabricación.

Los resultados de Schindel et al. [9] y Welter et al. [10] demostraron la posibilidad de enfocar el ultrasonido a múltiples frecuencias usando lentes Fresnel en el aire. [11] No obstante, las FZPs están diseñadas para funcionar y enfocarse a una frecuencia de diseño. En esta frecuencia, el comportamiento del FZP es óptimo y se logra enfocar a una determinada distancia focal. En aplicaciones médicas que usan lentes, especialmente en técnicas terapéuticas, es crítico tener un control fino y dinámico en la longitud focal. [12]

Más recientemente, Peng et al. [13] diseñaron una lente para enfocar las ondas acústicas estudiando analíticamente la transmisión a través de una rejilla acústica con hendiduras rizadas. Esta rejilla actúa como material con impedancia sintonizable, lo cual permite configurar la capacidad de focalización de la lente.

Cabe decir que el efecto de la frecuencia de operación en la distancia focal ya se ha justificado en óptica. [14] En este Trabajo Fin de Grado, se analiza principalmente la variación de los parámetros de enfoque cuando se trabaja a frecuencias operativas distintas a la frecuencia de diseño para el caso acústico, y se propone un mecanismo simple de control de longitud y posición focal.

2.2 Fundamentos físicos de una lente Fresnel

A lo largo de este proyecto, se ha utilizado una lente Fresnel con una distancia focal $F=5$ cm, una frecuencia de diseño $f_d=250$ kHz y $N=27$ zonas Fresnel. Las regiones doradas representan las zonas opacas de latón, mientras que las regiones transparentes se llenan de agua una vez sumergida en el medio acuático.

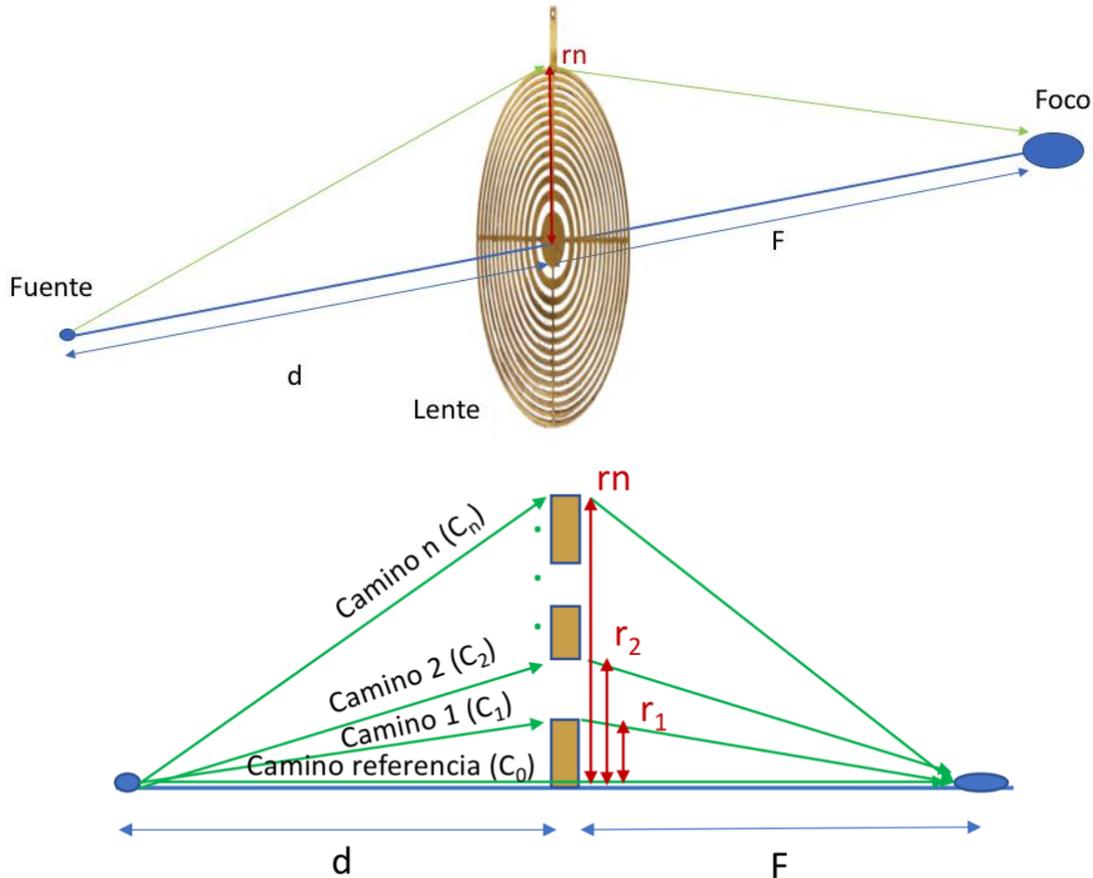


Figura 2. Principales parámetros físicos del funcionamiento de una lente de Fresnel

El funcionamiento de una lente Fresnel se basa en los grandes contrastes de impedancia entre el agua y el material de la lente, lo cual justifica la utilización de lentes de un material como el latón. Estudiando el esquema de la Figura 2 y siendo $\frac{\lambda}{2}$ la diferencia de caminos entre el camino de referencia y C_1 , existe una interferencia destructiva.

$$\Delta r = n \cdot \frac{\lambda}{2} = \frac{\lambda}{2} \quad (\text{Diferencia de caminos } C_1 - C_0 \rightarrow n = 1)$$

$$e^{-jk\Delta r} = e^{-j\frac{2\pi}{\lambda} \cdot \frac{\lambda}{2}} = e^{-j\pi} = -1 \rightarrow \text{Oposición de fase}$$

Siguiendo el mismo razonamiento, si hay una diferencia de λ entre el camino de referencia y C_2 , tiene lugar una situación de interferencia constructiva:

$$\Delta r = n \cdot \frac{\lambda}{2} = 2 \cdot \frac{\lambda}{2} = \lambda \quad (\text{Diferencia de caminos } C_2 - C_0 \rightarrow n = 2)$$

$$e^{-jk\Delta r} = e^{-j\frac{2\pi}{\lambda} \cdot \lambda} = e^{-j2\pi} = 1 \rightarrow \text{En fase}$$

De esta manera, podemos concluir que, entre una zona opaca y una transparente se produce una interferencia destructiva (oposición de fase), y entre una zona transparente y la siguiente zona transparente se produce una situación de interferencia constructiva (en fase). Considerando un camino n , se puede obtener la Ecuación 1 que representa el funcionamiento físico de una lente Fresnel donde d es la distancia entre la fuente y la lente, F es la distancia entre la lente y el foco, y r_n cada uno de los radios que conforman la lente:

$$\begin{aligned} \text{Camino de referencia: } C_0 &= d + F \\ \text{Camino 1: } C_1 &= \sqrt{d^2 + r_1^2} + \sqrt{F^2 + r_1^2} \\ \text{Camino } n: C_n &= \sqrt{d^2 + r_n^2} + \sqrt{F^2 + r_n^2} \text{ para } n = 1, 2, \dots, N \\ C_n - C_0 &= n \cdot \frac{\lambda}{2} \end{aligned}$$

$$d + F + \frac{n\lambda}{2} = \sqrt{d^2 + r_n^2} + \sqrt{F^2 + r_n^2} \text{ para } n = 1, 2, \dots, N \quad (1)$$

Para la situación que hemos descrito en la Figura 2 y, a partir de la Ecuación 1, los radios de la lente Fresnel vienen dados por la siguiente expresión: [8]

$$r_n = \frac{1}{2} \cdot \sqrt{\left(d + F + \frac{n\lambda}{2}\right)^2 + \left(\frac{d^2 - F^2}{d + F + \frac{n\lambda}{2}}\right)^2 - 2(d^2 + F^2)} \quad (2)$$

Aplicando la expresión anterior, podemos obtener los radios que conforman nuestra lente recogidos en la Tabla 1.

Radio	Longitud (m)	Radio	Longitud (m)
r ₁	0.016387	r ₁₅	0.072013
r ₂	0.023429	r ₁₆	0.074920
r ₃	0.028999	r ₁₇	0.077778
r ₄	0.033830	r ₁₈	0.080591
r ₅	0.038199	r ₁₉	0.083365
r ₆	0.042249	r ₂₀	0.086101
r ₇	0.046061	r ₂₁	0.088802
r ₈	0.049689	r ₂₂	0.091472
r ₉	0.053168	r ₂₃	0.094111
r ₁₀	0.056525	r ₂₄	0.096723
r ₁₁	0.059778	r ₂₅	0.099308
r ₁₂	0.062943	r ₂₆	0.101868
r ₁₃	0.066032	r ₂₇	0.104405
r ₁₄	0.069052		

Tabla 1. Radios de la lente Fresnel diseñada

Una vez calculados los radios de nuestra lente con un diámetro $2r_{27} = 20,88$ cm (Figura 3), se ha procedido a fabricarla en el taller del laboratorio de Física de la ETSIT. Para ello, se ha utilizado un procedimiento basado en el mecanizado de una lámina de latón utilizando la fresadora de control numérico Alarsis FR150 [30] capaz de definir los distintos radios.



Figura 3. Lente Fresnel utilizada en los experimentos del proyecto

Capítulo 3. Simulaciones con Comsol Multiphysics

En un Trabajo de Fin de Grado como el que nos ocupa, es necesario contrastar los resultados de modelos teóricos con otras soluciones como pueden ser los modelos numéricos y los modelos experimentales. Como consecuencia, se han llevado a cabo simulaciones en COMSOL Multiphysics Modeling Software, un paquete de software de análisis y resolución de problemas por elementos finitos.

3.1 Descripción del método de los elementos finitos

El método de los elementos finitos es un procedimiento numérico general para la aproximación de soluciones de ecuaciones diferenciales parciales muy utilizado en diversos problemas de ingeniería y física. Permite obtener una solución numérica aproximada sobre un cuerpo, estructura o dominio, sobre el que están definidas ciertas ecuaciones diferenciales que caracterizan el comportamiento físico del problema. De esta manera, el problema se divide en sub-dominios no-intersectantes entre sí denominados elementos finitos. El conjunto de elementos finitos forma una partición del dominio también denominada discretización o mallado. Dentro de cada elemento, se distinguen una serie de puntos denominados nodos. Así pues, el conjunto de nodos considerando sus relaciones de adyacencia se conoce como malla. [34]

En este caso, se trabaja con el módulo acústico de Comsol Multiphysics, aplicando la Ecuación de Helmholtz a cada nodo:

$$\nabla \left(-\frac{1}{\rho} \nabla p \right) = \frac{\omega^2}{\rho c^2} p \quad (3)$$

Por tanto, en el modelo de Comsol se deben definir ρ (densidad del medio de propagación), f (frecuencia de operación) y c (velocidad del sonido en el medio de propagación) para obtener el dato p (presión acústica). Para este proyecto, como se trabaja en un medio acuático, la densidad del agua es $\rho = 1000 \text{ kg/m}^3$ y la velocidad de propagación $c = 1500 \text{ m/s}$.

3.2 Definición del modelo de simulación

En Comsol Multiphysics se pueden realizar simulaciones utilizando modelos geométricos en 3D, 2D, 2D Axisimétrico... En este trabajo, se ha utilizado una definición 2D axisimétrica, ya que nuestro sistema de medida es completamente simétrico y nos permite reducir el coste computacional que supondría realizar una simulación 3D.

Se han definido una serie de condiciones de contorno. En primer lugar, se aplica la condición de Neumann sólido-rígido en la lente Fresnel de 1 mm de espesor (zona negra marcada en Figura 4), en ella, la presión será máxima y la velocidad nula. En las paredes del plano definido (zona verde marcada en Figura 4) se aplica la condición de radiación de Sommerfeld mediante la cual se define el dominio como infinito para evitar dispersiones numéricas a causa de reflexiones. Finalmente, se define una condición de presión (zona lila de la Figura 4) que se comporta como un pistón radiante sobre una superficie, tratando de aproximarse lo máximo posible a un transductor como el que utilizamos en las medidas experimentales del Capítulo 4. Además, esta zona de presión se sitúa a 35 cm de la lente, que coincide con la distancia que se utiliza entre transductor emisor y lente en la parte experimental del Capítulo 4.

Para finalizar el modelo, queda definir la malla que se ha mencionado en el Apartado 3.1. Se ha elegido una malla triangular, teniendo en cuenta que para conseguir una solución convergente en la simulación de Comsol Multiphysics, la distancia entre dos nodos adyacentes debe ser como máximo $\lambda/8$. [31]

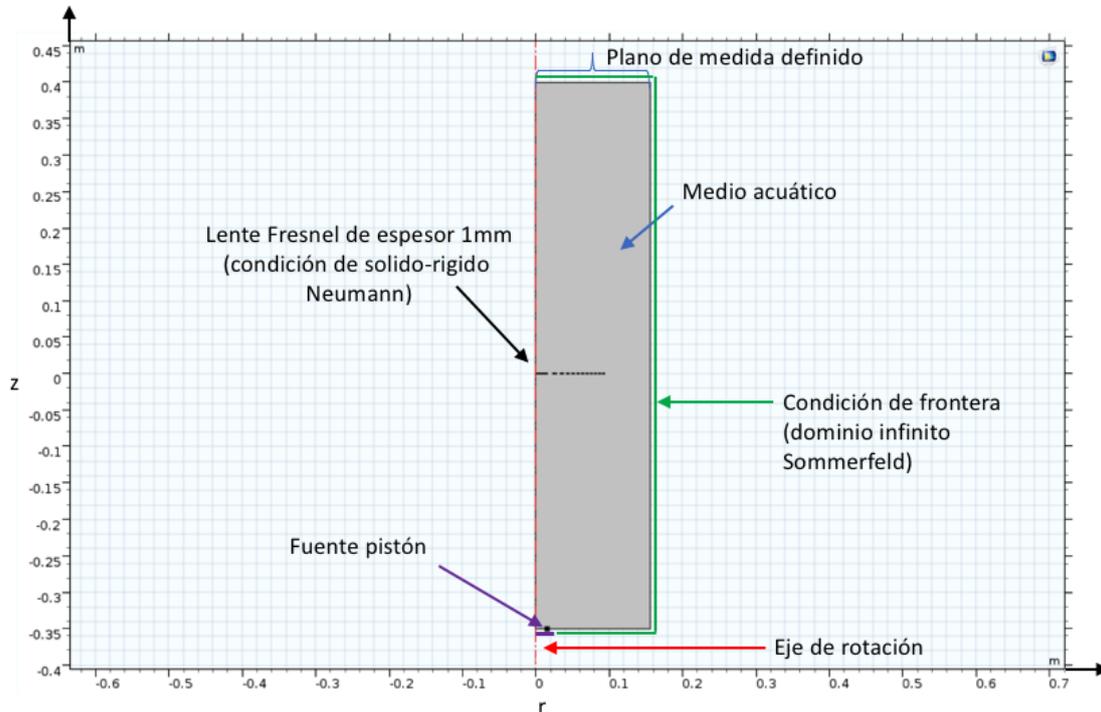


Figura 4. Esquema del experimento simulado en Comsol Multiphysics

3.3 Método de procesamiento de las simulaciones

Con la ayuda de la función de exportar de Comsol podemos procesar los datos simulados en Matlab. Para ello, se exportan los datos a un archivo .txt que contiene la información de los resultados de la simulación distribuidos en tres columnas, una para la coordenada r , otra para la coordenada z y finalmente otra para los valores de presión acústica. Importando este archivo en Matlab, podemos ser capaces de procesar la información de las simulaciones. Con el fin de tener una interfaz más amigable y poder obtener los resultados de manera más sencilla, he desarrollado una aplicación en Matlab usando su herramienta App Designer. App Designer se basa en el código orientado a objetos e integra las dos tareas principales de la creación de una aplicación: la organización de componentes visuales y la programación del comportamiento de la app. De esta manera, las aplicaciones se almacenan en forma de un solo archivo que contiene tanto la información del diseño como el código que define el comportamiento de la app. El código de la aplicación creada se puede encontrar en el Anexo 8.1 y tiene una apariencia como la que se puede observar en la Figura 5.

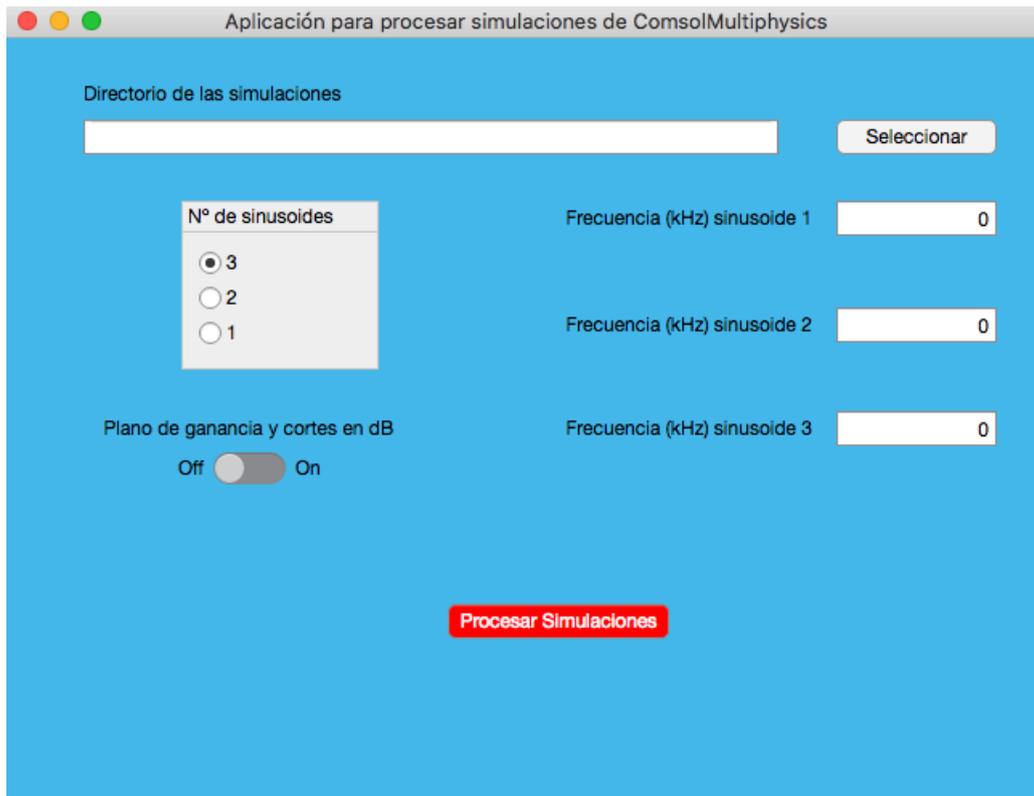


Figura 5. Aplicación para procesar simulaciones de Comsol Multiphysics

Seleccionando el directorio donde se encuentran los archivos .txt generados por Comsol Multiphysics mencionados anteriormente, correspondientes a cada una de las frecuencias que se desean analizar, tan solo tenemos que indicar el número de sinusoides que queremos sumar, la frecuencia de cada una, y si deseamos obtener los datos de manera absoluta en Pascales o preferimos una representación en dB teniendo en cuenta el plano medido aplicando las mismas frecuencias sin lente:

$$G(dB) = 20 \cdot \log\left(\frac{P}{P_0}\right) \quad (4)$$

Donde P es la presión medida al aplicar la lente Fresnel en Pascales y P_0 la presión medida sin lente Fresnel también en Pascales.

Así pues, una vez procesadas las simulaciones obtenemos cinco tipos de gráficas:

- Plano de presión en Pascales con lente Fresnel en la zona definida (ver Figura 4).
- Plano de presión en Pascales sin lente Fresnel en la zona definida (ver Figura 4).
- Plano de ganancia obtenido a partir de la diferencia de los dos anteriores en dB aplicando la Ecuación 4.
- Corte longitudinal: una gráfica en dos dimensiones representando la coordenada z en el eje de las abscisas y, en el eje de las ordenadas, los valores de presión en Pascales o dB correspondientes a los puntos que se encuentran en la coordenada $r=0$ (ver Figura 4).
- Corte transversal: una gráfica en dos dimensiones representando la coordenada r en el eje de las abscisas y, en el eje de las ordenadas, los valores de presión en Pascales o dB a lo largo de la coordenada r que se encuentran en el máximo del corte longitudinal anterior. (ver Figura 4).

Una vez hemos finalizado la aplicación que nos permite procesar la información obtenida en el interfaz de simulación de Comsol Multiphysics, hemos decidido generar tres señales por separado de frecuencias distintas: 275 kHz, 250 kHz y 225 kHz, todas ellas de 1 Pascal de presión emitida, para observar cómo se comporta nuestra lente de Fresnel diseñada para 250 kHz. A continuación, se muestran sus respectivos planos de presión absoluta en la Figura 6.

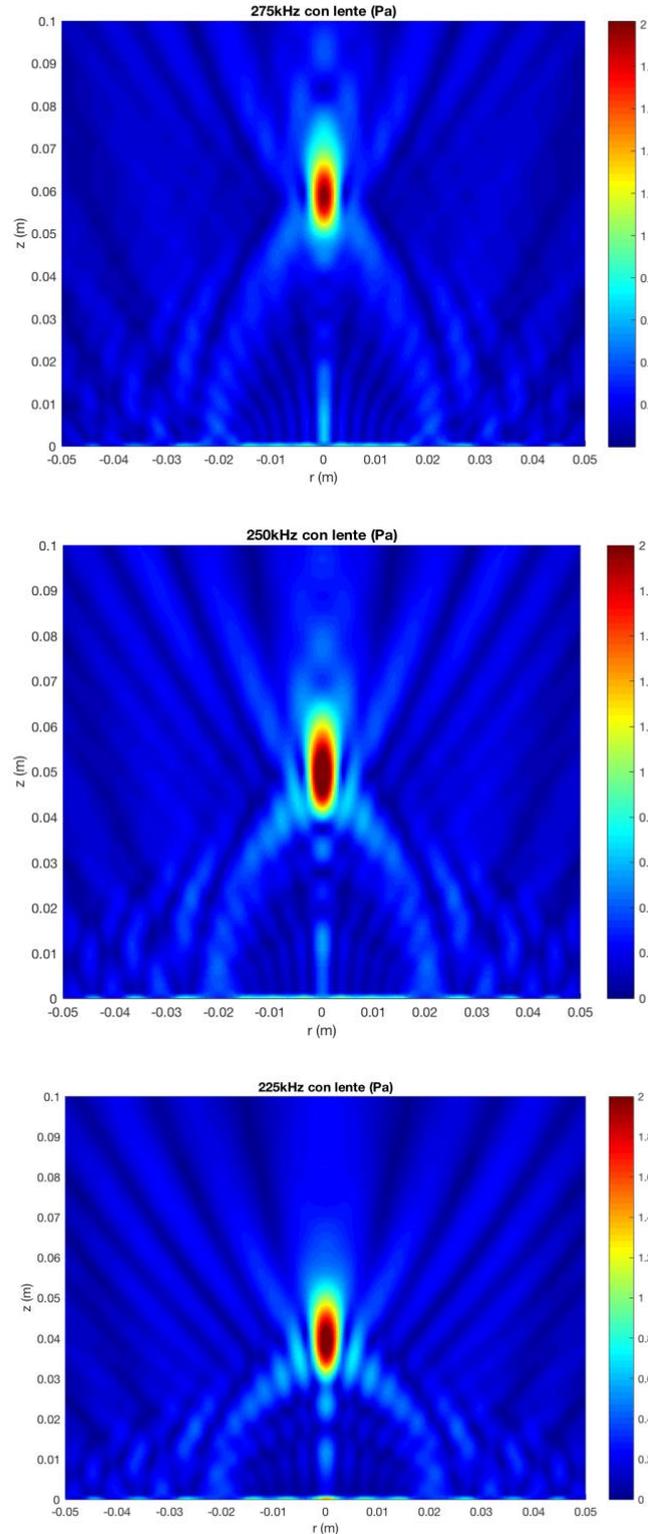


Figura 6. Planos de presión absoluta (Pa) simulados en Comsol Multiphysics utilizando frecuencias de 275 kHz (arriba), 250 kHz (centro) y 225 kHz (abajo)

Para observar el efecto descrito anteriormente acerca del desplazamiento de foco cuando la frecuencia de operación cambia respecto al valor de diseño, es interesante mostrar los cortes longitudinales (Figura 7) de las gráficas anteriores (Figura 6). Para ello, tenemos que localizar los datos de presión a lo largo del eje longitudinal z del foco de la lente. Como se puede observar, la localización del foco de la FZP cambia de 4.9 cm (250 kHz) a 5.9 cm (275 kHz) si aumentamos la frecuencia de trabajo. De la misma manera, la longitud del foco de la FZP también cambia de 4.9 cm (250 kHz) a 3.9 cm (225 kHz) si disminuimos la frecuencia de operación.

Otro efecto que podemos observar claramente con ayuda de estas gráficas es la reducción de presión acústica cuando aplicamos una frecuencia distinta a la de diseño. Este efecto se puede apreciar claramente en las simulaciones a 225 kHz y 275 kHz donde el valor de presión acústica es algo menor que en el de 250kHz, pasando de 2.397 Pa (250 kHz) a 2.043 Pa (275 kHz) o 2.218 Pa (225 kHz).

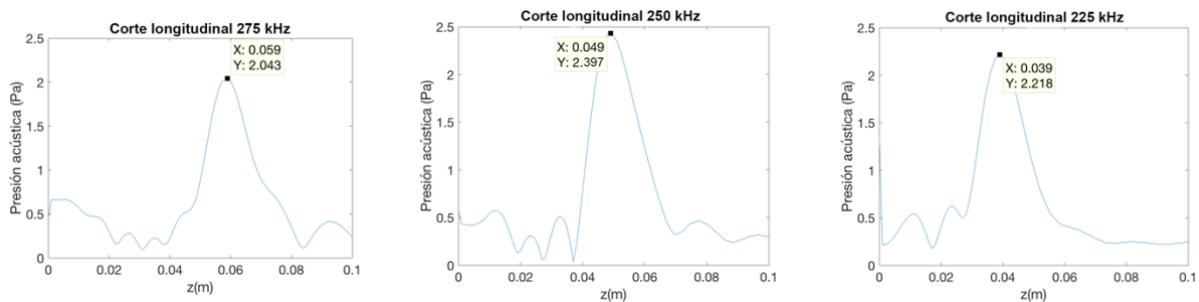


Figura 7. Gráficas de presión acústica frente a distancia longitudinal a 275 kHz (izquierda), 250 kHz (centro) y 225 kHz (derecha)

Así pues, durante estas simulaciones, además de comprobar el efecto de enfoque esperado tras emplear una lente Fresnel, hemos podido observar otros efectos que podemos resumir de la siguiente forma:

- Cuando la frecuencia de operación varía respecto a la frecuencia de diseño, la presión acústica se reduce.
- Cuando la frecuencia de operación cambia respecto al valor de diseño, la localización del foco también se modifica. Realizando diversas simulaciones en las mismas condiciones y aplicando las frecuencias de 200 kHz, 225 kHz, 250 kHz, 275 kHz y 300 kHz, podemos construir una gráfica (Figura 8) en la que se muestra que la longitud focal en una lente Fresnel se desliza casi linealmente con la frecuencia de funcionamiento, convirtiéndose en un parámetro de control muy dinámico.

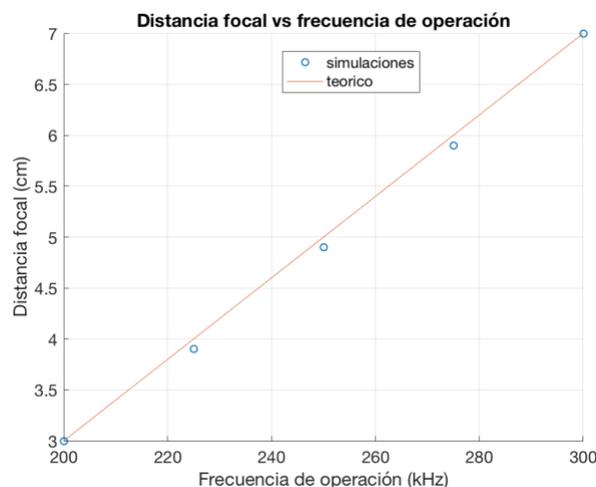


Figura 8. Gráfica distancia focal vs frecuencia de operación

3.4 Aplicación de una señal de 250 kHz

Una vez deducidos algunos de los efectos de enfoque con lentes Fresnel, se ha elegido simular una frecuencia de 250 kHz y 1 Pa para analizar más a fondo sus resultados.

Se han obtenido dos planos de presión en Pascales (Figura 9), uno al aplicar la lente Fresnel y otro sin ella. Debido a la gran diferencia de presión máxima entre ambas gráficas, nos vemos obligados a no utilizar la misma escala de colores para poder apreciar los resultados:

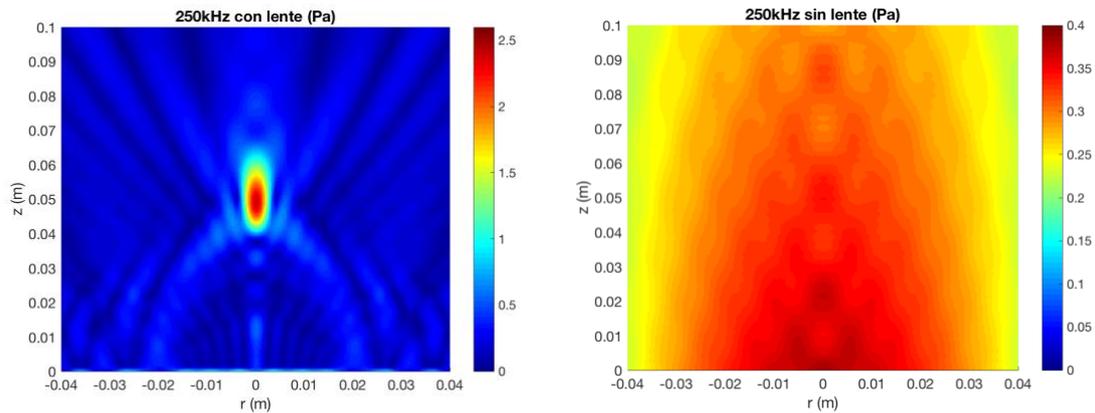


Figura 9. Planos de presión con lente (a-izquierda) y sin lente (b-derecha) en Pascales tras aplicar una señal sinusoidal de 250 kHz

Aplicando la Ecuación 4, se obtiene un plano de la ganancia de la lente en dB (Figura 10) a partir de las dos gráficas anteriores.

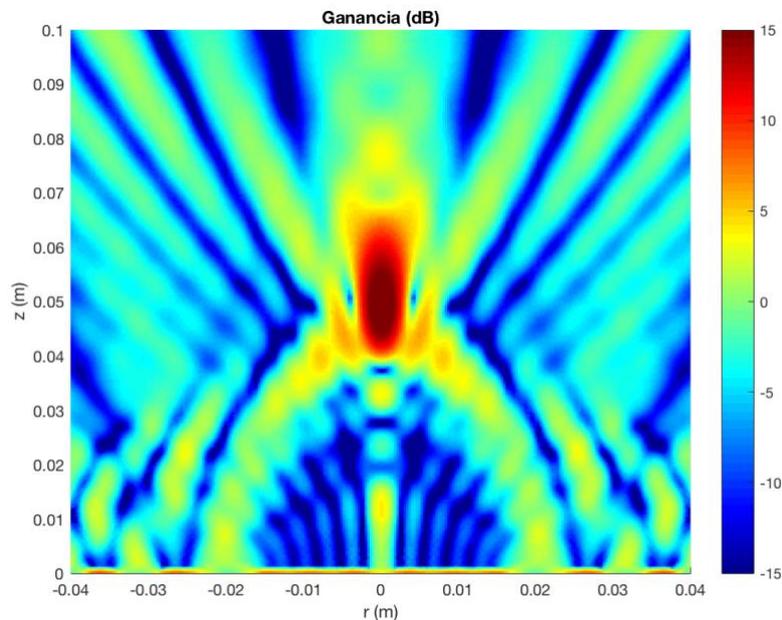


Figura 10. Plano de ganancia en dB tras simular una señal sinusoidal de 250 kHz

Finalmente, mostramos las gráficas correspondientes a los cortes longitudinal y transversal de la Figura 9a.

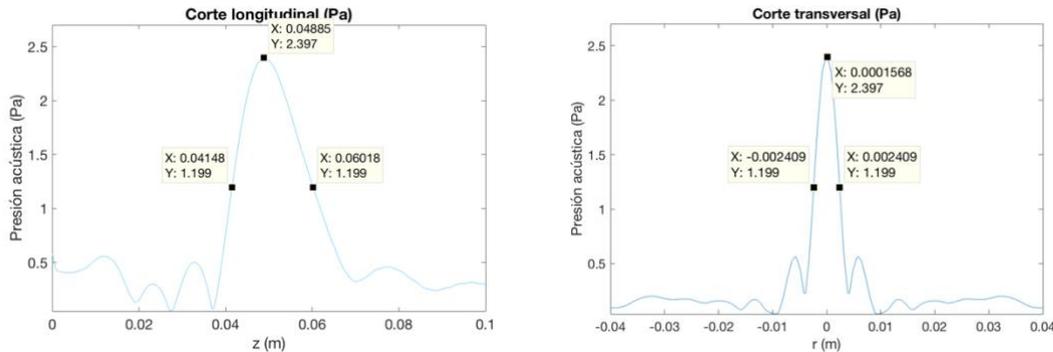


Figura 11. Corte longitudinal (izquierda) y corte transversal (derecha) de la Figura 9a tras simular una señal sinusoidal de 250 kHz

Analizando las gráficas de la Figura 11, podemos observar que el máximo del foco se sitúa en la posición longitudinal de 4.90 cm y transversal 0.00 cm obteniendo un valor máximo de amplitud de 2.397 Pa. Para calcular las dimensiones del foco, se puede utilizar un parámetro denominado FWHM (del inglés “Full Width at Half Maximum”), el cual es una medida de la extensión de una función, que viene dada por la diferencia entre los dos valores extremos de la variable independiente en los que la variable dependiente es igual a la mitad de su valor máximo. [35] En este Trabajo Fin de Grado utilizaremos la denominación $FWHM_{long}$ para la medida aplicada al corte longitudinal y $FWHM_{trans}$ en el caso del corte transversal. Así pues, con la ayuda de los cursores de la Figura 11, tendríamos unos valores de $FWHM_{long} = 6.00 \text{ cm} - 4.15 \text{ cm} = 1.85 \text{ cm}$ y $FWHM_{trans} = 0.25 \text{ cm} - (-0.25 \text{ cm}) = 0.50 \text{ cm}$.

*La resolución de todos los valores de longitud anteriores es 0.05 cm, ya que los compararemos con aquellos obtenidos en las medidas experimentales y, como veremos en el Capítulo 4, Apartado 4.1.1, los pasos de medida del robot son de 0.05 cm. El criterio utilizado para redondear los valores de longitud de las gráficas de cortes longitudinales y transversales, tanto en las dos medidas de simulación como en los experimentos, siendo x un número del 0 al 9, es el siguiente: si la parte decimal es igual o mayor que .x00 cm, pero menor que .x25 cm, se toma como .x0 cm; si es igual o mayor que .x25 cm y menor que .x75 cm, el valor pasa a ser .x5 cm; y si es mayor o igual que .x75 cm, se suma una unidad a la cifra situada justo a la derecha del separador decimal quedando .(x+1)0.

3.5 Simulaciones utilizando varias frecuencias a la vez

El desplazamiento de foco cuando aplicamos una frecuencia distinta a la de diseño, según hemos deducido al inicio de este capítulo, puede ser interpretada como una ventaja para nuestra posible aplicación médica. Este fenómeno puede ser aprovechado para emitir distintas frecuencias de manera controlada, de tal forma que uniendo los distintos focos que producen, consigamos generar un foco de dimensiones más grandes que podremos ajustar según las frecuencias que decidamos generar. Por ejemplo, utilizando las frecuencias anteriores de 225 kHz, 250 kHz y 275 kHz tendríamos un conjunto de focos como el que se puede observar en la Figura 12.

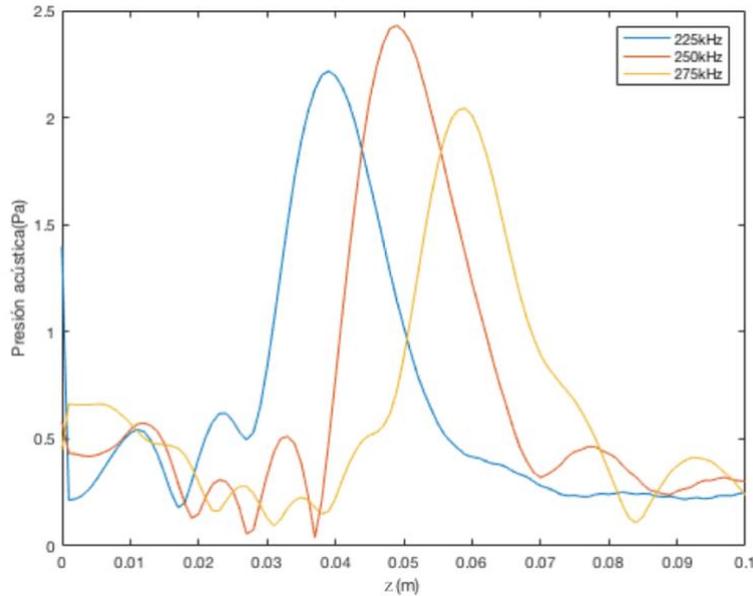


Figura 12. Gráficas de presión acústica frente a la distancia longitudinal superpuestas

Usando estas mismas frecuencias y calculando la función suma obtendríamos un foco expandido conformado por estas tres señales de 225 kHz, 250 kHz y 275 kHz que se situaría entre los 3.29 cm y 6.62 cm en la Figura 13.

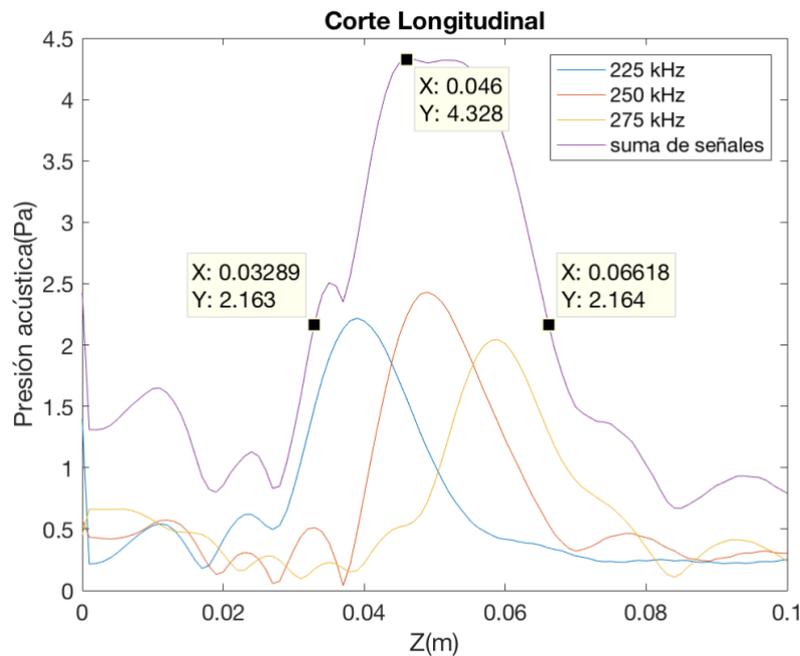


Figura 13. Gráfica de la suma de señales de 250 kHz, 275 kHz y 225 kHz para conformar un único foco

Tal vez, estas tres frecuencias no sean las más idóneas para conformar un foco intenso, ya que, para conseguirlo, las frecuencias de las señales que emitimos tienen que ser mucho más próximas. No obstante, resulta sencillo cambiarlas a nuestro criterio y seleccionar otras frecuencias de manera que se constituya un foco alrededor de 5 cm más centrado. Si seleccionamos las frecuencias 240 kHz, 250 kHz y 260 kHz y emitimos 1 Pa también en cada una de ellas, podemos observar como su suma constituye en la Figura 14 un foco desde 4.08 cm hasta 6.06 cm con presiones que pueden llegar hasta los 6.36 Pa.

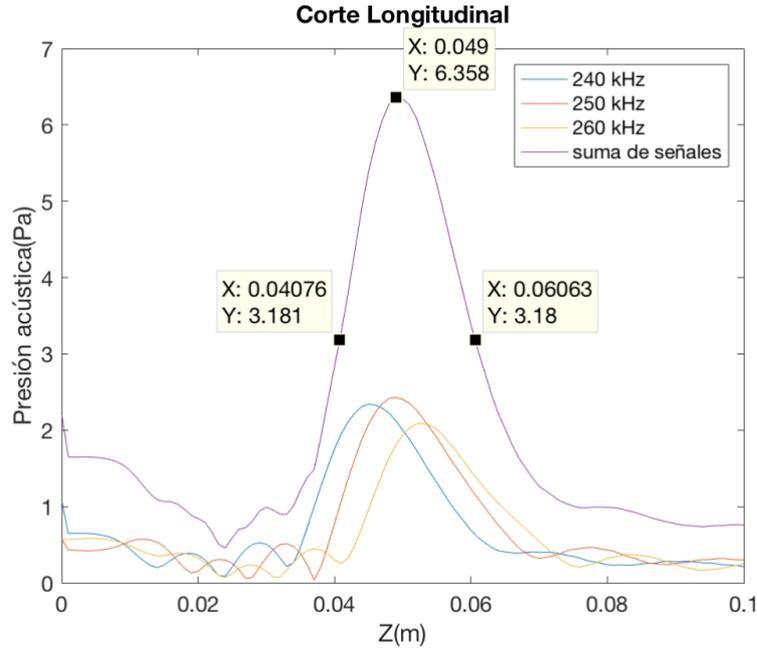


Figura 14. Gráfica de la suma de señales de 240 kHz, 250 kHz y 260 kHz para conformar un único foco más intenso

3.6 Aplicación de dos señales de 240 kHz y 275 kHz simultáneamente

A continuación, se ha estudiado una configuración en la que intentamos construir un foco más amplio usando dos señales de 240 kHz y 275 kHz, simulándolas de manera que la señal de 275 kHz se emita con un 20% menos de presión que la señal de 240 kHz, es decir, 1 Pa en la señal de 240 kHz y 0.8 Pa en la de 275 kHz. De este modo, demostramos que, utilizando varias señales de distinta frecuencia de manera controlada, podemos ser capaces de construir un foco según nuestras necesidades.

Se ha obtenido un plano de presión absoluta en Pascales al aplicar la lente y otro sin ella (Figura 15). En el primero de ellos, ya se observa que el foco es bastante más ancho que cuando emitíamos una única señal sinusoidal:

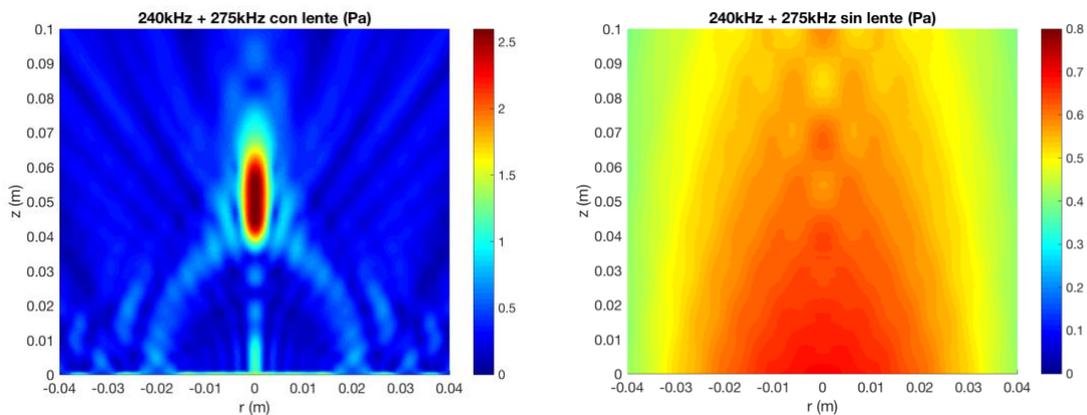


Figura 15. Planos de presión con lente (a-izquierda) y sin lente (b-derecha) en Pascales tras simular una suma de señales sinusoidales de 240 kHz y 275 kHz

Aplicando la Ecuación 4, se presenta un plano de la ganancia de la lente en dB en la Figura 16, obtenido a partir de las dos gráficas anteriores.

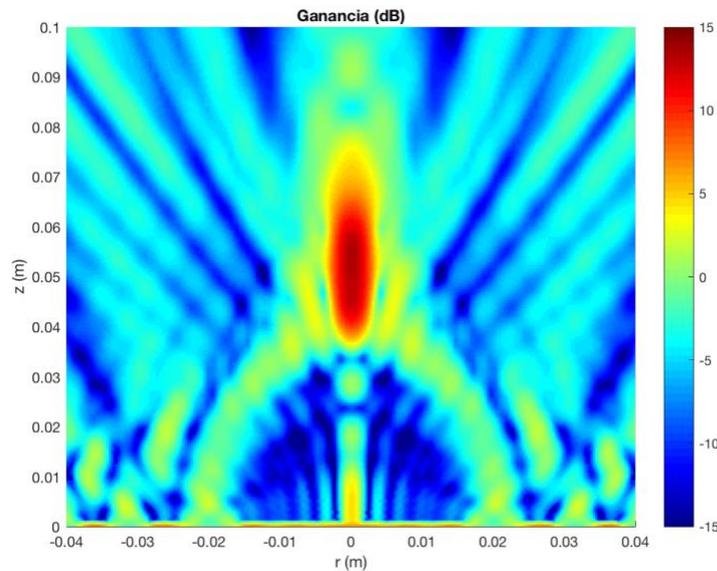


Figura 16. Ganancia en dB tras simular una suma de señales sinusoidales de 240 kHz y 275 kHz

Finalmente, en la Figura 17 mostramos las gráficas correspondientes a los cortes longitudinal y transversal de la Figura 15a.

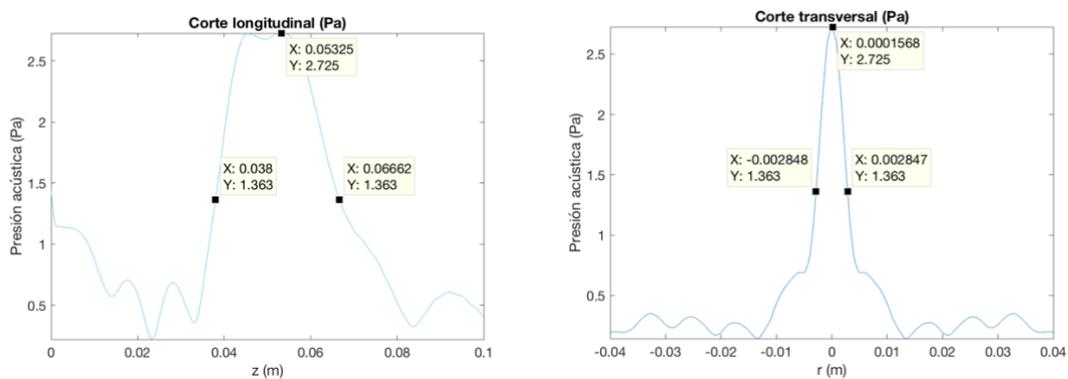


Figura 17. Corte longitudinal (izquierda) y corte transversal (derecha) de la Figura 15a tras simular una suma de señales sinusoidales de 240 kHz y 275 kHz

Analizando las gráficas de la figura anterior, podemos observar que el máximo del foco se sitúa en la posición longitudinal de 5.35 cm y transversal 0.00 cm obteniendo un valor máximo de 2.725 Pa. Calculamos las dimensiones del foco con la ayuda de los cursores de la Figura 17 y tenemos unos valores de $FWHM_{long} = 2.85$ cm y $FWHM_{trans} = 0.60$ cm, destacando el incremento de $FWHM_{long} = 1.85$ cm que teníamos en la primera medida a 2.85 cm. $FWHM_{trans}$, en cambio, permanece estable.

*La resolución de todos los valores de longitud anteriores es 0.05 cm, ya que también los compararemos con aquellos obtenidos en las medidas experimentales.

Capítulo 4. Resultados experimentales

4.1 Sistema original Arquímedes

La obtención de resultados experimentales es fundamental para la validación de los modelos teóricos, ya que permiten aceptar y refutar predicciones teóricas de forma precisa. Para poder realizar los experimentos se requiere un sistema complejo de medidas y adquisición, dadas las dificultades técnicas para poder controlar los dispositivos subacuáticos. Por las características de los experimentos que se realizan en una situación de inmersión, se utiliza un sistema completamente automatizado de adquisición de datos que permite la obtención de datos de la presión en cualquier punto del espacio del tanque de inmersión. Mediante estos resultados es posible caracterizar el comportamiento de los diferentes tipos de lentes desarrolladas por el Centro de Tecnologías Físicas (CTF) de la UPV.

El nombre del sistema completo de adquisición es Arquímedes. El formato de adquisición de los datos es 3Dreams (3 Dimensional e-Acoustic Measurement System), tal y como su propio nombre indica en inglés, un sistema de medida digital tridimensional. Al igual que el sistema que actualmente está funcionando en la cámara anecoica de la Universitat Politècnica de València, Arquímedes ha sido desarrollado con el CPOH (Grupo de Control Predictivo y Optimización Heurística de la UPV), encargándose este grupo de la parte de control de movimiento del robot y del desarrollo del software para la adquisición de datos acústicos. El sistema mecánico del robot ha sido diseñado y construido por la empresa de automatización ARTITECNIC. Arquímedes se puede dividir en tres partes fundamentales: el sistema robotizado con su control de movimiento, el sistema de adquisición de datos, y los transductores y amplificadores que se utilizan para la generación, transmisión y recepción de las ondas ultrasónicas. [15]

4.1.1 Robot y control del movimiento

El robot está construido en base al tamaño del tanque de inmersión donde se realizan los ensayos, que contiene agua destilada (desmineralizada) y tiene unas dimensiones de 0.5 m de ancho por 0.5 m de alto por 1 m de largo. Estas dimensiones suponen que el tanque de inmersión debe de contener en torno a 200 litros de agua destilada para ser funcional y permitir que, tanto los transductores como los dispositivos, estén completamente sumergidos, y evitar reflexiones por los cambios de impedancia producidos por el cambio de medio.

Las partes que constituyen el robot son: la estructura de sujeción o bastidor, tres brazos cartesianos para mover el transductor receptor, el brazo de muestra para sujetar la lente y el brazo manual para fijar el transductor emisor. Arquímedes permite el control de los brazos cartesianos y el brazo de muestra. El brazo de la muestra dispone de un servomotor que otorga un grado de libertad al poder rotar la muestra modificando el azimut del receptor. Los brazos cartesianos permiten controlar y automatizar el posicionamiento en las 3 dimensiones espaciales (X, Y, Z), siendo cada uno de los brazos el encargado del movimiento en uno de los tres ejes. El brazo de muestra, a pesar de contar con el servomotor de rotación, permite un ajuste manual a lo largo de los ejes X, Y, Z para poder colocar la muestra a la distancia y altura deseadas. Por último, el brazo del emisor es completamente manual. Esto permite el ajuste de la posición del transductor emisor de presión en cualquier posición del tanque de inmersión, aunque sin la libertad de poder cambiar su posición de forma automática.

El funcionamiento del robot de la Figura 18 es el siguiente: cada uno de los servomotores está conectado a un controlador que regula el suministro de corriente a los mismos y el paso, de esta forma se controla la dirección del movimiento. Los servomotores permiten la medida en pasos de 0.05 cm en cada uno de los tres ejes espaciales. En el caso del servomotor que controla el azimut se puede trasladar la resolución espacial a una resolución angular de 0.144° . Los controladores están conectados a su vez a una placa controladora modelo PCI-7330, de National Instruments, que se encarga de dirigir los movimientos del robot. El movimiento de los brazos con servomotores está controlado por el software LabVIEW del CPOH (3DREAMSUltra). Dicho programa permite el diseño y la automatización de las medidas experimentales, ya que se pueden establecer planos o volúmenes de medida configurando el movimiento de cada uno de los ejes de posicionamiento. Como medidas de seguridad añadidas al sistema mecánico, están los relés de final de carrera que evitan que un brazo pueda colisionar con otro y cualquier daño producido por un error de programación. Adicionalmente, existe un pulsador de parada de emergencia que detiene inmediatamente el movimiento del robot.

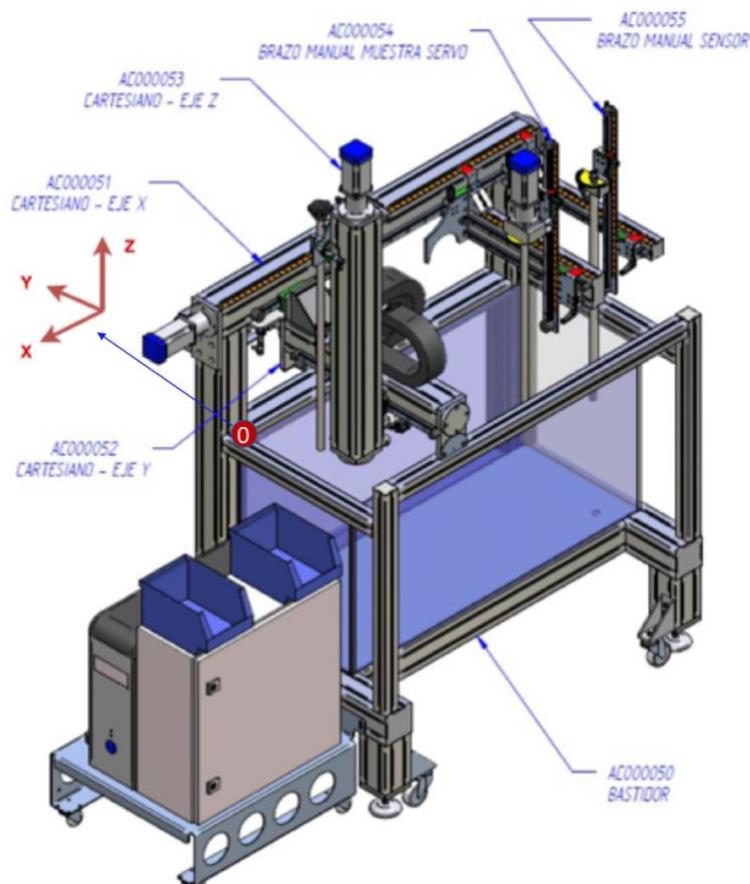


Figura 18. Esquema del robot de medida [32]

Una vez que se enciende el robot y arranca el programa de control 3DREAMSUltra, se realiza una autocalibración del sistema. Esta calibración se basa en colocar en el origen de coordenadas el brazo cartesiano y dejar a un azimut de 0° el brazo de muestra.

Por último, es necesario comentar que el robot está construido conforme a las directivas y normas que rigen en materia de seguridad industrial. Las directivas 2006/42/CE sobre maquinaria; la directiva 73/23/CEE modificada por la 2006/95/CEE de baja tensión, la directiva de compatibilidad electromagnética 92/31/CEE modificada por la 93/68/CEE y 2004/108/CE. Además de las normas armonizadas UNE-EN 1050, UNE-EN 418, UNE-EN 954 y UNE-EN 60204-1 que regulan la seguridad de maquinaria, evaluación de riesgo, paradas de emergencia, diseño y equipo eléctrico respectivamente. [15]

4.1.2 Sistema de adquisición de datos

El sistema de adquisición de datos está formado por la tarjeta de control de movimiento NI PCI-7330 [16], el osciloscopio digital PicoScope 3224 [17] y el programa de adquisición 3DREAMSUltra, desarrollado en LabVIEW.

La tarjeta NI PCI-7330 tiene, como ya hemos comentado, la función de controlar el robot. Su arquitectura es de doble procesador y es capaz de procesar en tiempo real hasta cuatro ejes de movimiento coordinadamente. Esta tarjeta calcula la posición instantánea y controla la aceleración y la velocidad del movimiento de los ejes hasta la posición marcada. Para obtener las señales se utiliza un osciloscopio digital modelo PicoScope 3224 que se conecta por USB al ordenador. El osciloscopio dispone de dos canales y obtiene muestras con una resolución de 12 bits. Tiene un margen dinámico de 72 dB y un ancho de banda de 10 MHz. Tanto la placa de control como el osciloscopio están controladas por el programa 3DREAMSUltra instalado en el ordenador del laboratorio de Física. [15]

4.1.3 Transductores y amplificadores

Para poder medir las muestras es necesario el uso de transductores y amplificadores que posibilitan la conversión de una señal digital en una señal acústica de suficiente potencia en transmisión, y permiten obtener una señal digital a partir de una señal acústica en recepción. En nuestro caso, se utilizan transductores piezoeléctricos (Figura 19), los cuales están fabricados con cristales cerámicos que vibran a la frecuencia de resonancia propia del transductor, tras aplicarles un pulso eléctrico. [29] Esta frecuencia, depende directamente del espesor del transductor definida por:

$$D (\text{espesor}) = \frac{\lambda}{2} \quad (5)$$

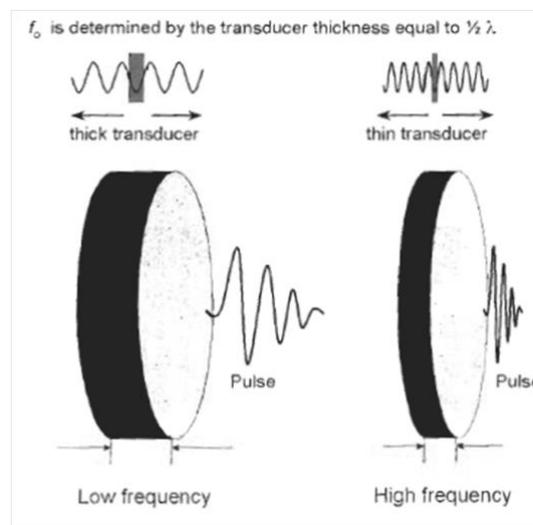


Figura 19. Influencia del espesor del transductor en la frecuencia de resonancia [29]

Además, en nuestra aplicación, interesa que el factor de calidad Q del transductor sea lo más bajo posible ya que, de esta manera, tendremos un rango más amplio de frecuencias para aplicar al paciente según se puede observar en la Figura 20.

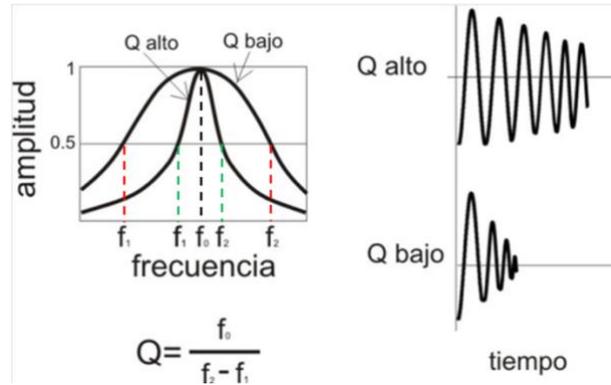


Figura 20. Factor de calidad en un transductor de ultrasonidos [29]

Existen tres tipos de configuraciones para realizar la medida experimental [15]:

1. Utilizar un único transductor como emisor y receptor a la vez. Esta configuración se utiliza para medir la respuesta de eco provocada por la muestra, por ello se le denomina Pulso-Eco.
2. Hacer uso de una pareja de transductores, los cuales se utilizan uno como emisor y otro como receptor indistintamente. Esta configuración está pensada para medir los coeficientes de transmisión de los dispositivos acústicos y todo lo relacionado con los mismos. Con esta configuración, también es posible realizar una medida de un plano o un volumen de presiones, aunque con algunas limitaciones de resolución debido al diámetro del transductor.
3. Situar un transductor como emisor y colocar un hidrófono en el brazo cartesiano para medir líneas, planos o volúmenes de presiones. Gracias al tamaño reducido del hidrófono se puede medir con una resolución mucho mayor que utilizando una pareja de transductores.

Arquímedes permite utilizar cualquiera de las configuraciones mencionadas anteriormente, ya que se puede colocar el tipo de transductor que se desee en los brazos habilitados para ello. En nuestras medidas, hemos utilizado siempre la tercera configuración. A continuación, se van a mostrar los transductores y amplificadores utilizados para realizar los experimentos de este Trabajo Fin de Grado.

Uno de los elementos más importantes para la medición de planos acústicos es el hidrófono. El Centro de Tecnologías Físicas de la UPV cuenta con un hidrófono de Precision Acoustics, modelo 1.0 mm Needle Hydrophone. [18] Este hidrófono es capaz de medir altas frecuencias, aunque tengan un nivel de señal muy débil. La sensibilidad del hidrófono es de 850 nV/Pa con una tolerancia de ± 3 dB. La respuesta en frecuencia es plana ± 2 dB entre 3 y 12 MHz y ± 4 dB entre 200 kHz hasta 15 MHz. El ancho de banda va de 5 kHz hasta 15 MHz. [15]

Para poder utilizar el hidrófono se requiere de un preamplificador [21] y un acoplador DC [22]. El preamplificador está acoplado directamente a la salida del hidrófono como podemos observar en la Figura 21. El acoplador DC, conectado al preamplificador, se encarga de suministrar la potencia necesaria al preamplificador.



Figura 21. Fotografía del hidrófono y el preamplificador empleados [15]

Los transductores utilizados pueden ser sumergidos durante largos periodos de tiempo. En nuestro Trabajo de Fin de Grado, todo nuestro estudio está basado en el transductor Imasonic 250 kHz [19] cuya función de transferencia se puede consultar en la Figura 22.

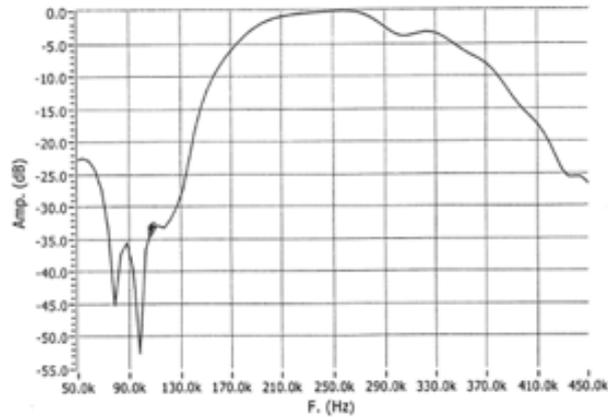


Figura 22. Función de transferencia del transductor Imasonic 250 kHz [19]

Para poder generar y amplificar las señales se suele utilizar un generador de pulsos, el Pulser modelo 5077PR de Parametrics [20], el cual cuenta con un amplificador integrado y dos puertos, uno de salida y otro de entrada. Permite generar pulsos con frecuencias comprendidas entre 100 kHz y 20 MHz, una PRF (Pulse Repetition Frequency) desde 100 Hz hasta 5 kHz y una amplitud de pulso entre 100 y 400 V. [15] El hándicap del uso de este generador-amplificador es que solo se pueden generar señales predefinidas. Uno de los objetivos de este proyecto, es precisamente sustituir este “pulser” por un generador de onda continua configurable remotamente por conexión USB desde nuestro ordenador, el Agilent 33220A. Con este avance, tenemos más flexibilidad a la hora de insertar señales en el experimento y corregimos algunos de los problemas de calidad del “pulser” que veremos en el Apartado 4.3.

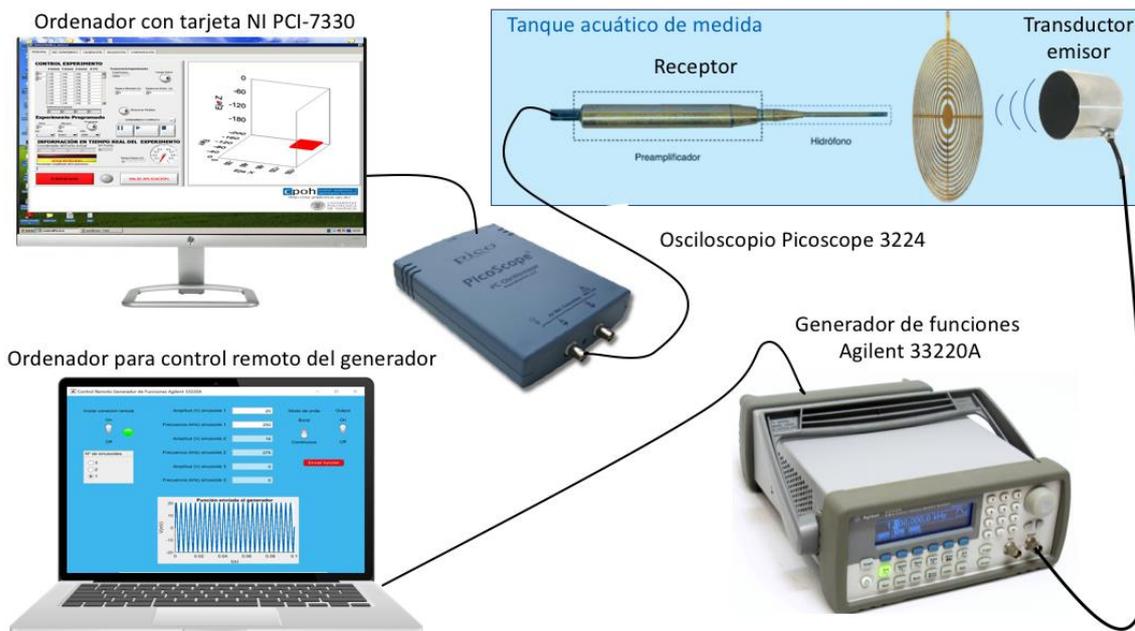


Figura 23. Esquema de conexiones del sistema completo de medidas

4.2 Preparación del experimento

Una vez descritos los distintos componentes del sistema de medida Arquímedes, pasamos a realizar el experimento práctico que intenta demostrar nuestro objetivo relacionado con la emisión de varias señales para conformar un foco controlado con una lente Fresnel y su posible aplicación médica. Para ello, necesitamos utilizar tres programas ya creados por el CTF de la Universitat Politècnica de València (Figuras 24, 27 y 29), y finalmente dos aplicaciones realizadas por mí en Matlab para controlar el generador Agilent 33220A remotamente y procesar las medidas obtenidas (ver Figuras 31 y 33).

4.2.1 Conexión de los equipos

En primer lugar, se enciende el sistema del robot. Seguidamente, se conecta el transductor emisor elegido a la salida del generador Agilent 33220A y el hidrófono a la entrada del osciloscopio digital Picoscope 3224, según se puede observar en la Figura 23.

4.2.2 Preparación del experimento

A continuación, debemos definir el plano de medida que se va a utilizar en el experimento. El plano tiene que ser definido por tres coordenadas X, Y, Z (ver Figura 18) teniendo en cuenta, por supuesto, que el foco de la lente entre dentro del plano propuesto. Para ello, se utiliza la aplicación de la Figura 24.

Figura 24. Script Matlab para crear un plano de medida

Tras introducir parámetros como el delay, tiempo de adquisición, promediado y definir las medidas del plano elegido, la aplicación crea en el directorio y nombre elegido un archivo .txt con las coordenadas que ha de seguir el robot para ir adquiriendo todas las muestras. El contenido del fichero es similar a la Figura 25.

1	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
2	-50.000000	-80.000000	-220.000000	0.000000	200.000000	20.000000	50.000000
3	-50.000000	-80.000000	-225.000000	0.000000	200.000000	20.000000	50.000000
4	-50.000000	-80.000000	-230.000000	0.000000	200.000000	20.000000	50.000000
5	-50.000000	-80.000000	-235.000000	0.000000	200.000000	20.000000	50.000000
6	-50.000000	-80.000000	-240.000000	0.000000	200.000000	20.000000	50.000000
7	-50.000000	-80.000000	-245.000000	0.000000	200.000000	20.000000	50.000000
8	-50.000000	-80.000000	-250.000000	0.000000	200.000000	20.000000	50.000000
9	-50.000000	-80.000000	-255.000000	0.000000	200.000000	20.000000	50.000000
10	-50.000000	-80.000000	-260.000000	0.000000	200.000000	20.000000	50.000000

Figura 25. Contenido de un fichero de puntos de medida

En él, los datos están distribuidos de la siguiente manera:

Posición x(mm)	Posición Y(mm)	Posición Z (mm)	Rotación (°)	Tiempo de adquisición (μ s)	Delay (μ s)	Promediado
-------------------	-------------------	--------------------	-----------------	--	---------------------	------------

El script anterior origina un plano a partir de unir diversos puntos de medida en forma de zigzag (Figura 26) para evitar grandes desplazamientos de un lado a otro del motor que se mueve en el eje Y. Si eliminamos desplazamientos tan largos se evitan muchas vibraciones del hidrófono en el momento de detenerse el motor y, por tanto, se reduce el error de medida. Este recorrido en forma de zigzag se debe tener muy en cuenta a la hora de procesar las medidas en Matlab.

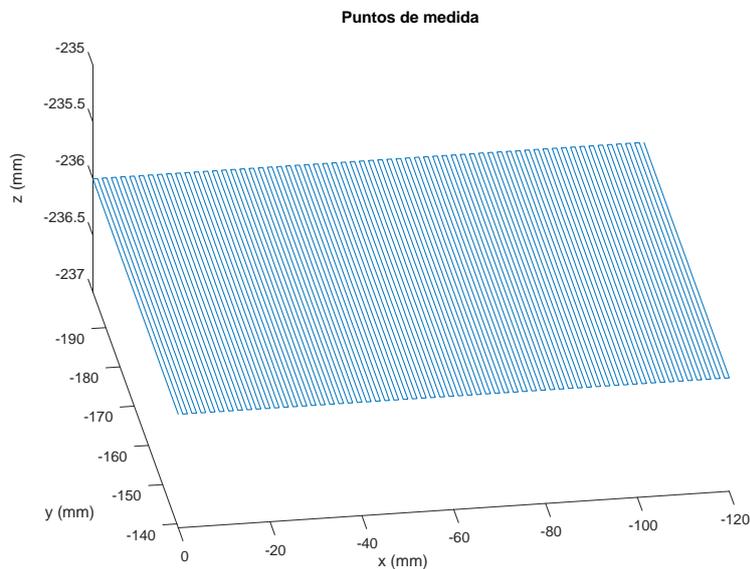


Figura 26. Ejemplo de plano de medida

4.2.3 Obtención de los datos

Utilizando el programa 3DREAMSUltra, podemos llevar a cabo una medida experimental. Desde el menú calibración (ver Figura 27), se puede mover el brazo mecánico del receptor a la posición deseada. Esto es especialmente útil para comprobar que ninguno de los puntos extremos definidos en el paso anterior colisiona con la lente.

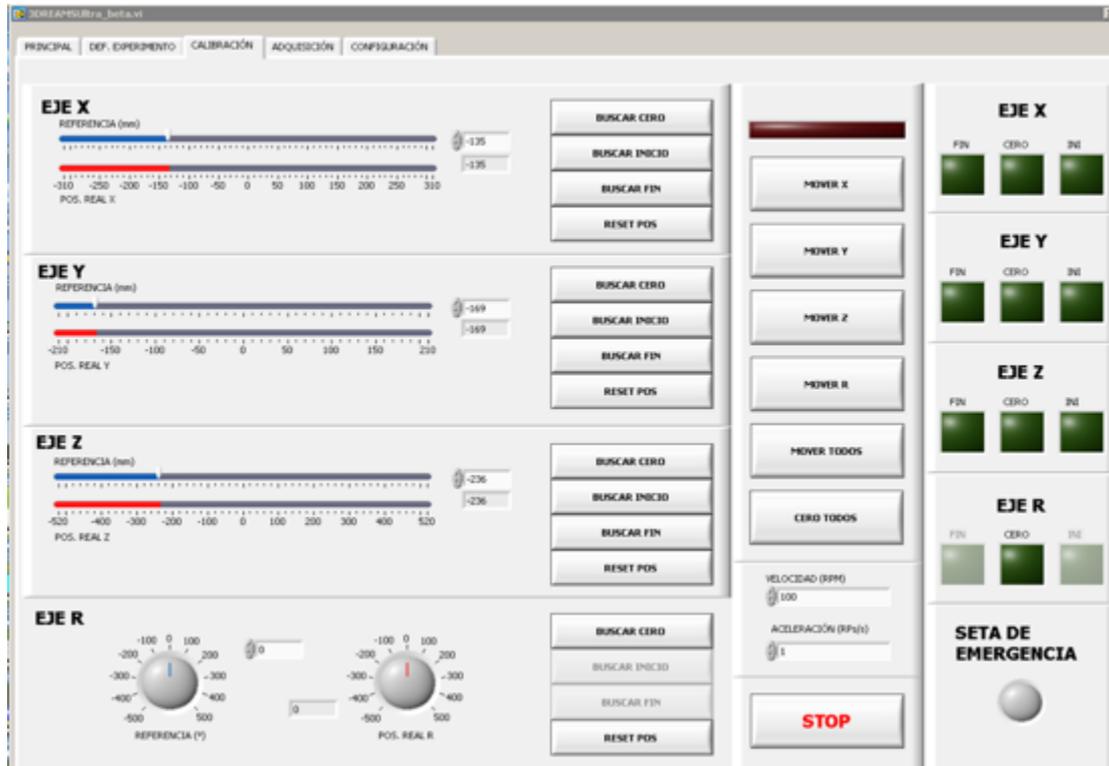


Figura 27. Captura del menú de calibración del programa de control del robot

Como ya hemos explicado en el Apartado 4.1.1, el brazo del emisor es completamente manual, lo cual permite el ajuste del transductor emisor de presión en cualquier posición del tanque de inmersión. Así pues, desplazamos el brazo del transductor emisor hasta una posición lo más centrada posible del tanque de inmersión y alineándola con el hidrófono que se moverá con los tres brazos cartesianos que se indican en la Figura 18.

Finalmente, desde la pestaña “principal” de la aplicación (ver Figura 28) seleccionamos el archivo .txt generado anteriormente con el plano a medir y un directorio para almacenar las muestras obtenidas en los distintos puntos recorridos por el hidrófono. En la parte derecha de esta ventana se observa el plano definido en color rojo, permitiendo visualizar los puntos medidos y realizar un seguimiento del experimento, ya que cada punto que ha sido medido cambia a color azul.

De esta manera, se crea un archivo .3dr por cada punto medido, el cual almacena varios valores en función del tiempo de adquisición que elijamos. Además, debemos seleccionar un tiempo de espera por vibración igual o superior a 1 segundo, de lo contrario el plano medido contendrá mucho ruido causado por el movimiento del propio robot. El programa también tiene una opción para añadir un tiempo de espera por rotación. No obstante, este tiempo de espera es nulo, ya que en las medidas utilizadas no se ha realizado ningún movimiento rotatorio.

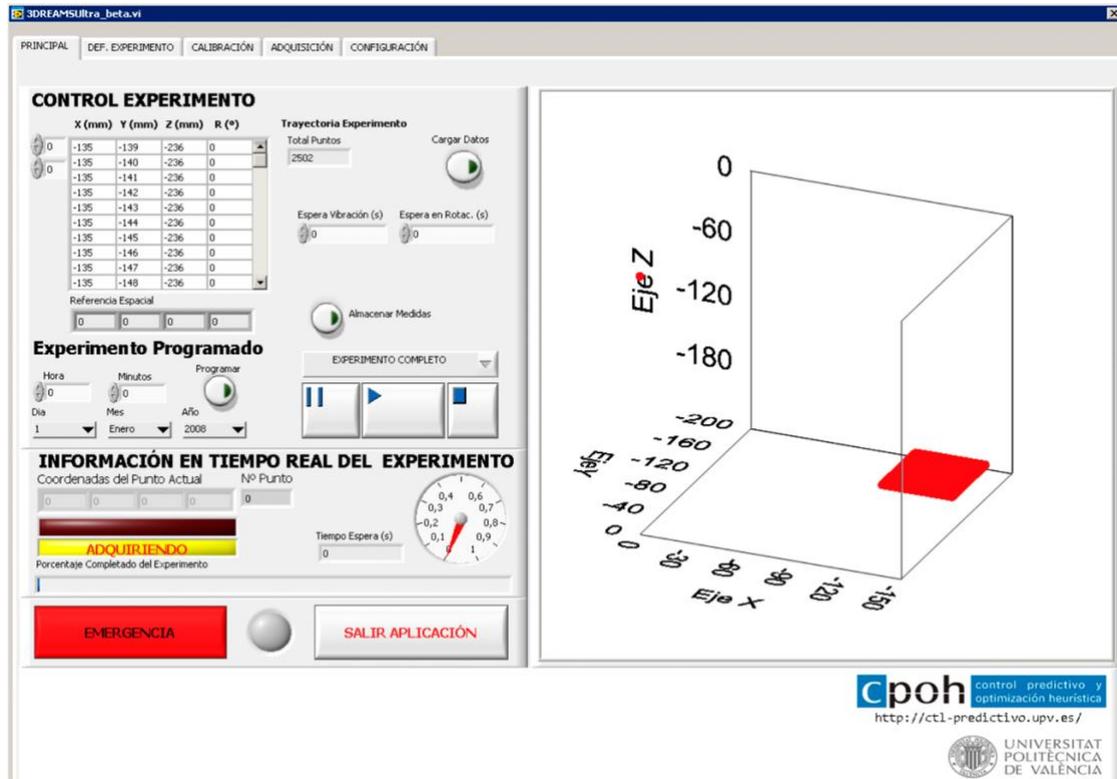


Figura 28. Captura del menú principal del programa de control del robot donde se visualiza el plano de medida

4.2.4 Conversión de muestras del formato 3Dreams a Matlab

Utilizando el programa exp2mat de LabView (Figura 29), seleccionamos la carpeta de muestras .3dr obtenidas para convertirlas a formato .mat y poder ser procesadas en Matlab como vamos a ver en el Apartado 4.2.5.



Figura 29. Captura del programa utilizado para convertir archivos de formato .3dr a .mat

4.2.5 *Procesado básico de las medidas*

Para procesar las medidas de una manera más interactiva, he desarrollado una aplicación (Figura 30) con App Designer de Matlab cuyo código completo se encuentra en el Anexo 8.3.

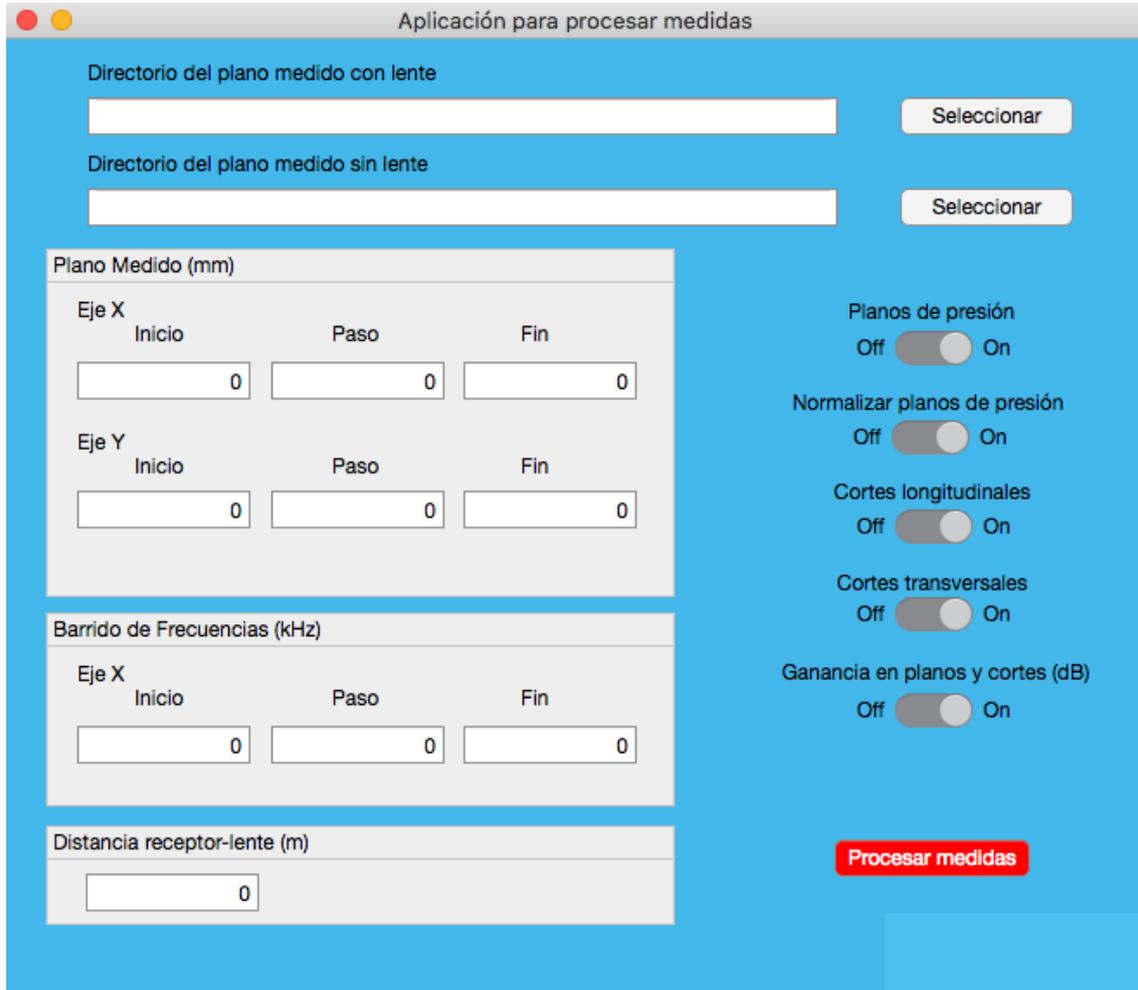


Figura 30. Aplicación desarrollada para procesar las medidas del sistema Arquímedes

El método estándar para procesar una medida como las que hemos realizado a lo largo del proyecto empieza con la selección de los directorios en los que se encuentran las medidas de los planos obtenidos al aplicar una lente Fresnel y sin ella. A continuación, es necesario definir las dimensiones X e Y de los planos medidos, así como las frecuencias que deseamos barrer.

Seguidamente, se define la distancia mínima a la que hemos situado el receptor de la lente durante la creación del plano de medida, normalmente este valor suele estar alrededor de los 0.5 cm para evitar posibles colisiones con la lente en la medida. Además, en la parte derecha de la ventana, podemos seleccionar los distintos resultados que deseamos obtener:

- Planos de presión

En este apartado se obtiene un plano de presiones en Pascales con lente Fresnel para cada una de las frecuencias seleccionadas en la casilla de barrido (realizando la FFT de la señal temporal). Además, se obtiene un plano de presiones con los máximos de la señal temporal en cada punto del plano medido con y sin lente.

- Planos de presión normalizados

Se obtienen los mismos planos del punto anterior pero normalizados a la unidad, dividiendo los valores entre el máximo registrado.

- Plano de ganancia en dB

Se calcula un plano de ganancia a partir de la diferencia de los planos medidos con y sin lente en dB aplicando la Ecuación 4.

- Cortes longitudinales y transversales

En este caso se utiliza un procedimiento ligeramente distinto a las simulaciones para obtener los cortes longitudinales y transversales, ya que en una medida experimental no podemos tener un alineamiento perfecto del foco en el centro de la coordenada r , y no sería válido realizar un corte a lo largo de este punto. Como consecuencia, es necesario calcular el máximo del plano medido.

Así pues, el corte longitudinal es una gráfica en dos dimensiones representando la coordenada z (ver Figura 31) en el eje de las abscisas y, en el eje de las ordenadas, los valores de presión en Pascales o dB correspondientes a los puntos que se encuentran a lo largo del eje z que comparten la misma coordenada r que el máximo calculado. (ver Figura 31).

En el caso del corte transversal, se utiliza también el máximo del plano medido. Una vez detectado, se dibuja una gráfica en dos dimensiones representando la coordenada r (ver Figura 31) en el eje de las abscisas y, en el eje de las ordenadas, los valores de presión en Pascales o dB correspondientes a los puntos que se encuentran a lo largo del eje r que comparten la misma coordenada z que el máximo calculado. (ver Figura 31).

En todas estas gráficas, se utilizan ejes de coordenadas denominados z y r , relativos a la posición de la lente. De esta manera, $r=0$ estaría localizado en el centro de la lente. El eje z está situado sobre el eje X (ver Figura 31) pero $z=0$ estaría en la posición x donde se ha colocado la lente. Utilizando estos ejes de coordenadas relativos, tendremos una visión más clara de aquello que está sucediendo en el experimento y las conclusiones serán expuestas con más claridad.

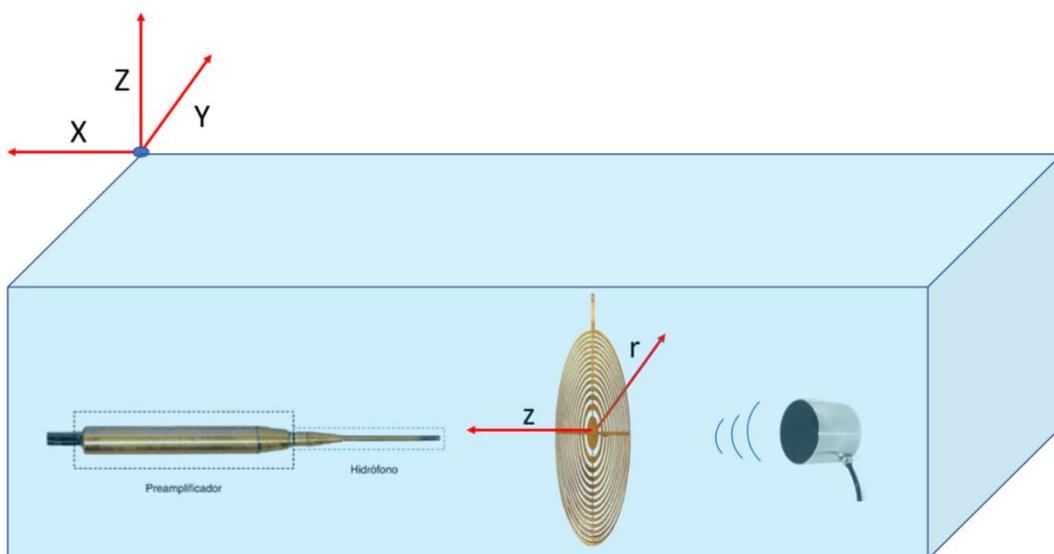


Figura 31. Esquema del cambio de ejes propuesto para facilitar la interpretación de los resultados

4.3 Desarrollo de una aplicación en Matlab para controlar de manera remota el generador de funciones Agilent 33220A

Hasta el momento en el CTF de la UPV se han utilizado dos tipos de generación de señales: bien a través de un generador de funciones externo, o bien a partir de un “pulser”. No obstante, en la primera de las opciones siempre se han utilizado las formas de onda predefinidas en el generador, generando una única señal. Uno de los grandes objetivos que perseguíamos en este trabajo, era precisamente poder configurar un generador de funciones para poder sumar dos o más señales a distintas frecuencias y conseguir que una misma lente presente varios focos simultáneamente o un foco más extenso definido por varias frecuencias.

Precisamente, utilizando el Instrument Control Tool de Matlab se ha comprobado que es posible conectar el generador de funciones Agilent 33220A con Matlab. Tras instalar los drivers necesarios [24] y seguir las recomendaciones de los tutoriales de Matlab sobre cómo configurar instrumentos Agilent [25, 27, 28], he podido desarrollar un script que sea capaz de generar una “Arbitrary Waveform” compuesta por la suma de varias funciones sinusoidales de distintas frecuencias. La estructura básica de nuestro programa de generación de señales es:

1. Conexión remota con el generador: conociendo la dirección USB de nuestro generador podemos generar un código como el siguiente que nos permita establecer una conexión remota con el generador:

```
MyFGen=fgen();  
MyFGen.Resource='USB0::0x0957::0x0407::MY44018984::INSTR';  
connect(MyFGen);
```

2. Definición de la señal a introducir: en este apartado se define la señal que queremos transferir al generador, la cual puede estar formada por un número cualquiera de funciones senoidales. En este caso, hemos elegido dos:

```
t1=0:0.0001:1;
```

```
%Definimos la amplitud pico-pico y frecuencia de cada una de las  
señales sinusoidales
```

```
amplitudePP1 = 10;  
frequency1 = 200; %Unidad relativa a 1kHz
```

```
amplitudePP2 = 8;  
frequency2 = 300; %Unidad relativa a 1kHz
```

```
%Se definen ambas señales sinusoidales  
waveform1 = (amplitudePP1/2)*sin(2*pi*frequency1*t1);  
waveform2 = (amplitudePP2/2)*sin(2*pi*frequency2*t1);
```

```
%Obtenemos la señal suma de las dos anteriores  
waveformArray = waveform1+waveform2;
```

3. Transferir señal al generador: en este último paso, utilizando las herramientas del Instrument Control Tool de Matlab podemos transferir la señal definida al generador de manera remota:

```
MyFGen.Frequency=1000; %Habilitamos una escala de frecuencias de 1kHz  
MyFGen.Waveform='Arb'; %Seleccionamos el modo de onda arbitrario  
downloadWaveform(MyFGen,waveformArray);%Cargamos la onda en el  
generador
```

`enableOutput(MyFGen); %Se habilita la salida del generador.`

Para estructurar todo el código anterior y ofrecer una interfaz más amigable al usuario (Figura 32), se ha utilizado también App Designer de Matlab.

Este programa, por tanto, ha sido realizado de manera que se puede seleccionar el número de señales sinusoidales que queremos que produzca el generador, habilitando la inserción de parámetros como su amplitud y frecuencia. También tenemos tres conmutadores que nos permiten iniciar la conexión remota (en cuyo caso el LED cambiará de color), el modo de onda y la habilitación de salida (On y Off). Si elegimos el modo de onda Burst, se utilizarán los parámetros por defecto, es decir, 10ms de periodo y un 50% de “duty cycle”. No obstante, en nuestros experimentos siempre hemos trabajado con onda continua. Finalmente, al pulsar el botón “Enviar Función” transmitiremos la onda al generador y se dibujará una vista previa de la misma en la parte inferior central de la ventana. El código íntegro del programa se puede encontrar en el Anexo 8.2 y está basado en las instrucciones que hemos descrito anteriormente.

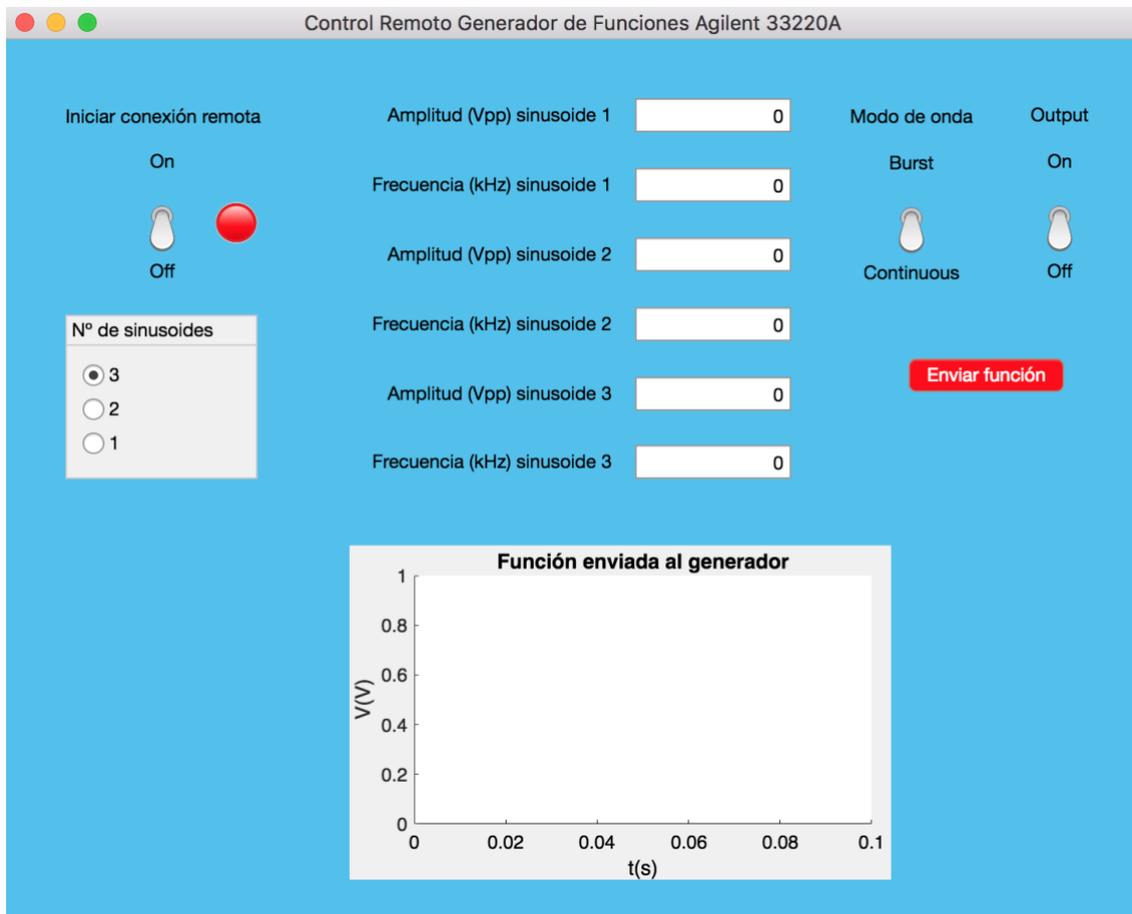


Figura 32. Aplicación desarrollada para controlar el generador Agilent 33220A remotamente

Tras varias pruebas utilizando este modo de generación de señal juntamente con el osciloscopio Gw-Instek 1054B [13] hemos podido concluir:

- El máximo nivel de tensión pico a pico capaz de ofrecer el generador Agilent 33220A es de $20V_{pp}$. Esta conclusión también está corroborada por el manual del generador de funciones Agilent 33220A en la página 172 [33].
- El máximo número de puntos que puede contener la función arbitraria que carguemos desde Matlab al generador de funciones serán 65000.

La segunda conclusión sabíamos que, en principio, no causaría ningún problema al desarrollo del experimento. No obstante, la limitación de $20V_{pp}$ tal vez podría ser conflictiva, ya que estaríamos trabajando a unos niveles de tensión menores de lo habitual. Para comprobar este fenómeno, hemos conectado el “pulser” que se había utilizado anteriormente en el laboratorio al osciloscopio GWinstek 1054B [26]:



Figura 33. Señal generada por el Pulser 5077PR

El Pulser 5077PR que se utilizaba en experimentos anteriores proporcionaba una $V_{ppMax}=28,8V$, nosotros en cambio con el generador de Agilent 33220A hemos podido trabajar como máximo a $20V_{pp}$. No obstante, nuestro generador de funciones ofrece una señal mucho más regular y fiable que el “pulser”, ya que como podemos observar en la Figura 33, el “pulser” no ofrece unas amplitudes constantes entre un periodo y el siguiente.

4.4 Primera medida experimental: Emisión de una señal sinusoidal de 250kHz

En primer lugar, tras haber desarrollado la aplicación que nos permite controlar el generador Agilent 33220A remotamente y conocer sus funcionalidades, hemos decidido emitir un seno con una frecuencia de 250 kHz utilizando el transductor Imasonic 250 kHz que hemos descrito en el Apartado 4.1.3.

4.4.1 Programación del generador de funciones remotamente

Seleccionamos, primeramente, una única señal sinusoidal de 250 kHz y 20 V_{pp} (Figura 34), la cual es la máxima amplitud que puede entregar nuestro generador según hemos deducido en el Apartado 4.3. Además, esta frecuencia coincide con la de la primera simulación realizada en el Capítulo 3, Apartado 3.2.

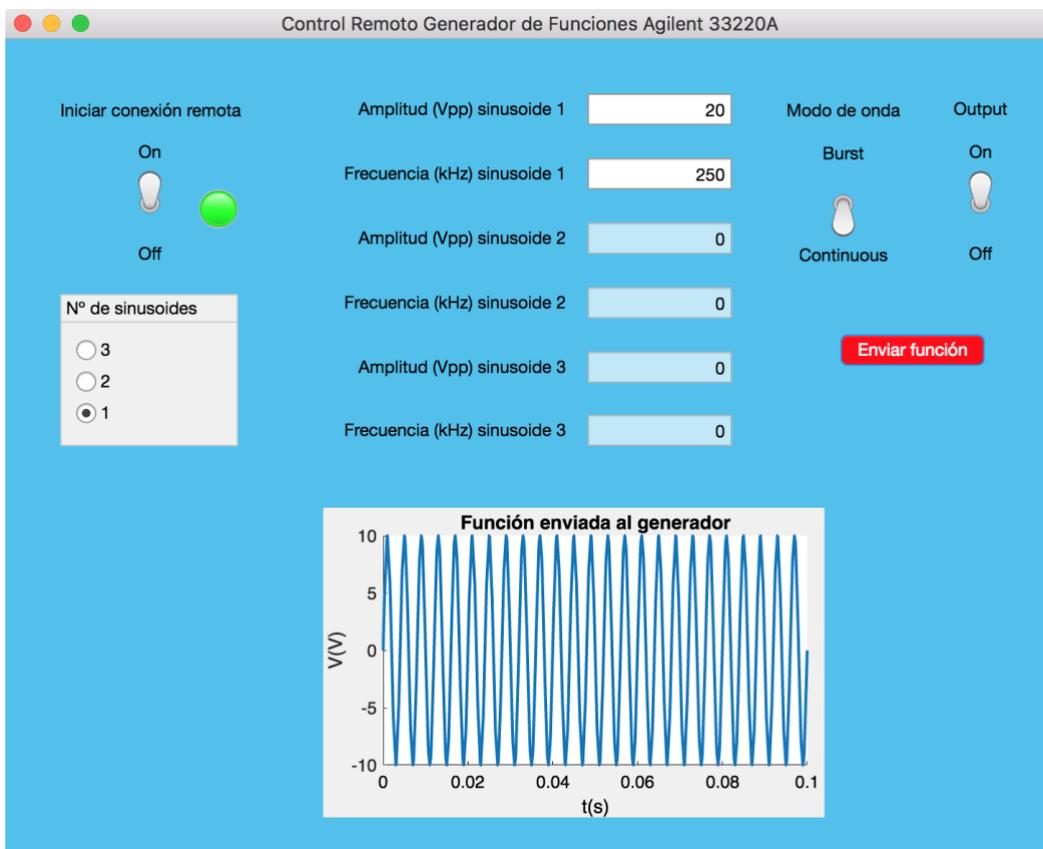


Figura 34. Captura de la aplicación para controlar el generador de funciones tras enviar una señal sinusoidal de 250 kHz y 20 V_{pp}

4.4.2 Realización de la medida

Siguiendo las indicaciones explicadas al inicio de este mismo capítulo, podemos configurar el script “Crear Plano” de la Figura 24 para crear un plano y unos parámetros de medida como los que se exponen en la Tabla 2.

PLANO	
Z	-236 mm
Y	-209:1: -129 mm
X	-135:1:0 mm
Rotación	0°

PARÁMETROS	
Delay	150 μ s
Tiempo de adquisición	300 μ s
Promediado	100 muestras

Tabla 2. Plano y parámetros de la medida con una señal sinusoidal a 250 kHz

4.4.3 Procesado de las medidas

Utilizando la aplicación desarrollada en el Apartado 4.2.5, podemos procesar los resultados obtenidos tras realizar las medidas en el sistema Arquímedes. Dos de las gráficas que aparecerán serán los planos de presión en Pascales con lente Fresnel y sin ella (Figura 35).

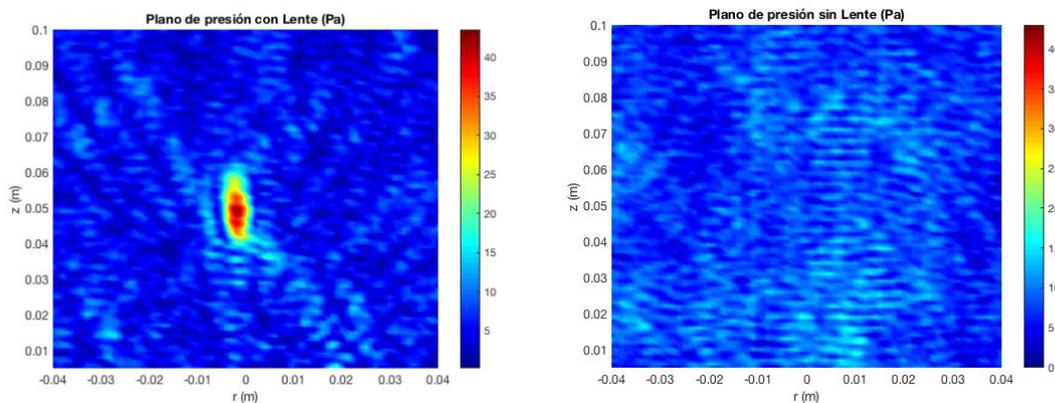


Figura 35. Planos de presión con lente (a-izquierda) y sin lente (b-derecha) en Pascales tras aplicar una señal sinusoidal de 250 kHz

Si aplicamos la Ecuación 4 donde P es la presión en Pascales del plano de presión con lente y P_0 la presión en Pascales del plano de presión sin lente, obtendremos un plano de ganancia en dB (Figura 36) a partir de los dos anteriores.

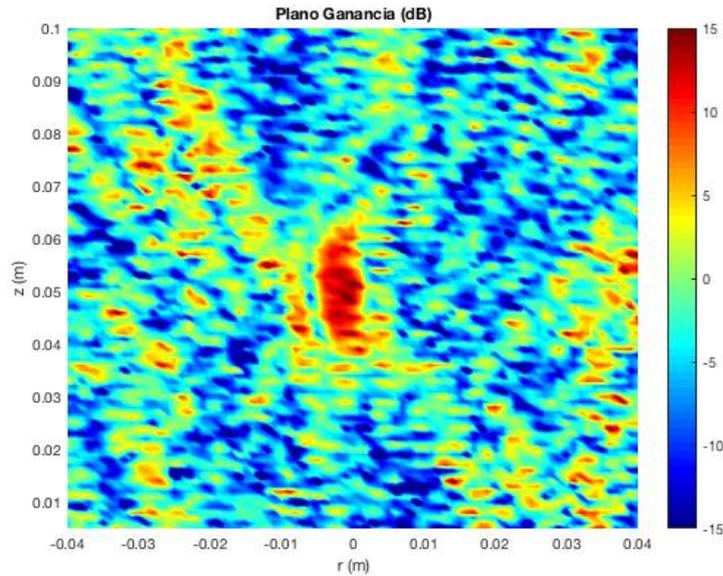


Figura 36. Plano de ganancia en dB tras aplicar una señal sinusoidal de 250 kHz

Según hemos descrito en el Apartado 4.2.5, las otras dos gráficas que obtendremos serán las de corte longitudinal y corte transversal de la Figura 35a. Estas gráficas nos serán de gran ayuda en la parte de conclusiones.

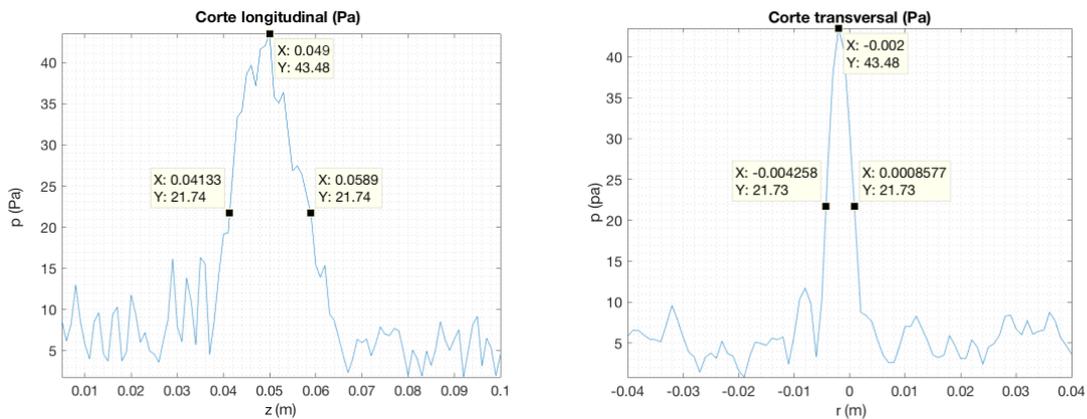


Figura 37. Corte longitudinal (izquierda) y corte transversal (derecha) tras aplicar una señal sinusoidal de 250 kHz

Procediendo de la misma manera que con las simulaciones, podemos observar que el máximo del foco se sitúa en la posición longitudinal de 4.90 cm y transversal -0.20 cm obteniendo un valor máximo de 43.48 Pa. Para calcular las dimensiones del foco, con la ayuda de los cursores de la Figura 37, tendríamos unos valores de $FWHM_{long} = 1.75$ cm y $FWHM_{trans} = 0.55$ cm.

*La resolución de todas las longitudes anteriores es 0.05 cm, ya que los pasos de medida del robot son de 0.05 cm.

4.5 Segunda medida experimental: Emisión de dos señales sinusoidales de 240 kHz y 275 kHz

En este apartado, tras haber comprobado que se puede emitir de manera correcta una única señal senoidal con el generador Agilent 33220A controlado remotamente, hemos pasado a verificar la posibilidad de emitir dos señales senoidales sumadas, lo cual, será la primera vez que se realice en el laboratorio de Física de la ETSIT.

4.5.1 Programación del generador de funciones remotamente

La segunda medida se ha realizado también con el transductor Imasonic 250 kHz. Como se puede ver en la Figura 22, este transductor es capaz de emitir a frecuencias de 200 kHz y 300 kHz con una atenuación máxima de 3dB.

Para ser coherentes con la simulación realizada en el Capítulo 3, Apartado 3.4, hemos seleccionado dos señales de las mismas frecuencias con las que hemos simulado: 240 kHz y 275 kHz. Al igual que hemos hecho en las simulaciones, la señal de 275 kHz tiene un 20% menos de amplitud que la señal de 240 kHz. Teniendo en cuenta que la suma de señales emitidas por el generador Agilent 33220A no puede superar los 20 V_{pp}, seleccionamos 10 V_{pp} en la señal de 240 kHz y 8 V_{pp} para la señal de 275 kHz.

La señal que hemos introducido en el generador será, por tanto, la suma de ambas señales senoidales como se muestra en la Figura 38:

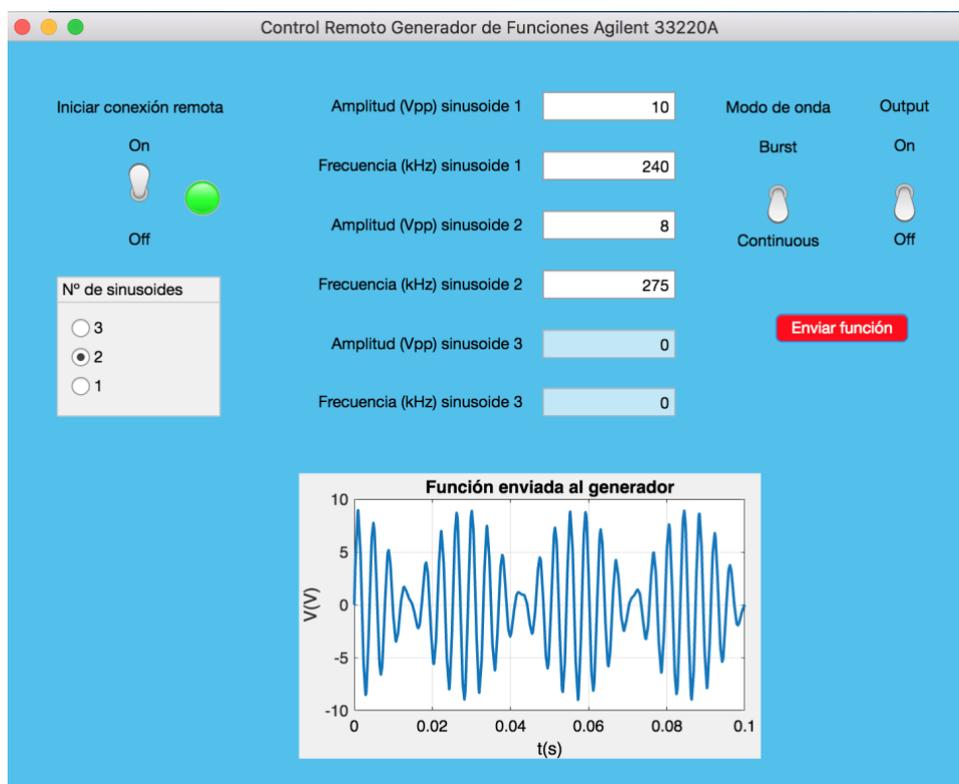


Figura 38. Captura de la aplicación para controlar el generador de funciones tras enviar una suma de señales sinusoidales de 240 kHz y 10 V_{pp}, y 275 kHz y 8 V_{pp}

4.5.2 Realización de la medida

Siguiendo las directrices que hemos descrito en el inicio del Capítulo 4 podemos ser capaces de llevar a cabo esta medida. Para ello, configuramos el script “CrearPlano” de la Figura 24 para crear un plano y unos parámetros de medida como los que se exponen en la Tabla 3.

PLANO	
Z	-236 mm
Y	-209:1: -129 mm
X	-135:1:0 mm
Rotación	0°

PARÁMETROS	
Delay	150 μ s
Tiempo de adquisición	300 μ s
Promediado	100 muestras

Tabla 3. Plano y parámetros de la medida con dos señales sinusoidales a 240 kHz y 275 kHz

4.5.3 Procesado de las medidas

En este apartado también obtenemos los planos de presión en Pascales con lente Fresnel y sin ella (Figura 39). La primera información que nos salta a la vista es el ancho del foco, ya que se ha visto aumentado al aplicar dos señales sinusoidales a la vez:

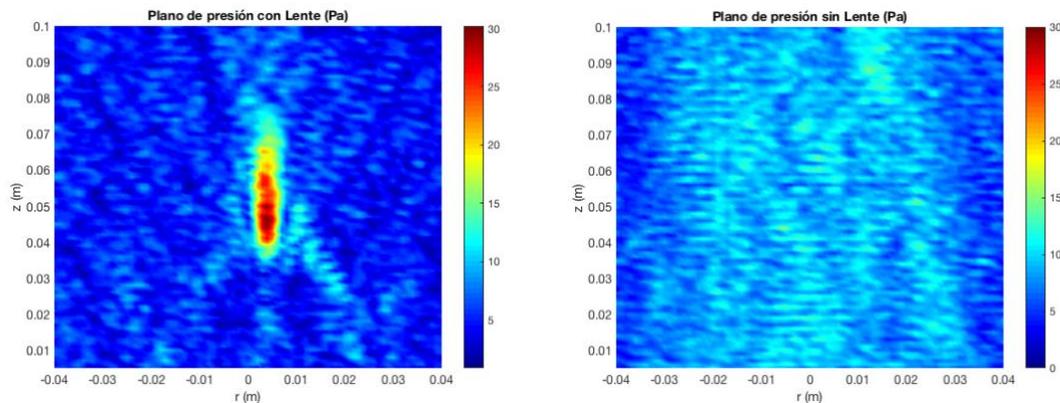


Figura 39. Planos de presión con lente (a-izquierda) y sin lente (b-derecha) en Pascales tras aplicar una suma de señales sinusoidales de 240 kHz y 275 kHz

Aplicando la Ecuación 4 podemos obtener el plano de ganancia en dB a partir de los dos anteriores en la Figura 40.

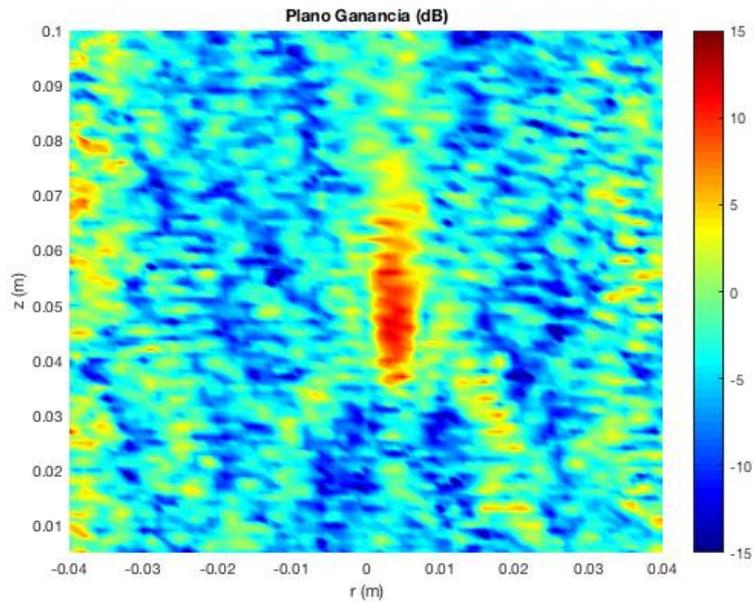


Figura 40. Plano de ganancia en dB tras aplicar una suma de señales sinusoidales de 240 kHz y 275 kHz

Finalmente, mostramos en la Figura 41 las gráficas correspondientes a los cortes longitudinal y transversal de la Figura 39a, gracias a ellas podemos calcular el ancho del foco a partir de valores como el $FWHM_{trans}$ y $FWHM_{long}$

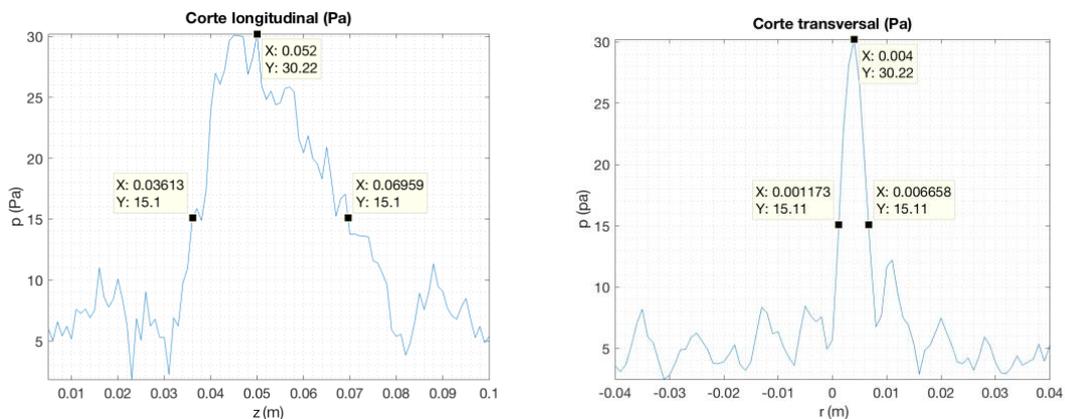


Figura 41. Corte longitudinal (izquierda) y corte transversal (derecha) tras aplicar una suma de señales sinusoidales de 240 kHz y 275 kHz

Se puede observar que el máximo del foco se sitúa en la posición longitudinal de 5.20 cm y transversal 0.40 cm obteniendo un valor máximo de 30.22 Pa. Para calcular las dimensiones del foco, con la ayuda de los cursores de la Figura 41, tendríamos unos valores de $FWHM_{long} = 3.35$ cm y $FWHM_{trans} = 0.55$ cm.

Si contrastamos los resultados del experimento de una señal sinusoidal y los resultados del experimento con dos señales sinusoidales simultaneas $FWHM_{long}$ pasa de tener un valor de 1.75 cm a 3.35 cm. Al mismo tiempo, $FWHM_{trans}$ permanece inalterada.

4.6 Comparación de los resultados de las simulaciones y experimentos en ambas medidas

Los valores de presión en Pascales obtenidos en simulaciones y medidas experimentales no son comparables entre sí, ya que en las simulaciones se han utilizado valores de emisión de 1 Pascal y en los experimentos el generador de funciones ha emitido señales de hasta 20 V_{pp}, sin tener un control directo sobre la presión emitida en Pascales. Sí son objeto de comparación el resto de parámetros. Para comparar la posición del foco en los ejes longitudinal y transversal, y sus dimensiones se ha creado la Tabla 4.

	UNA SEÑAL SINUSOIDAL DE 250 kHz			DOS SEÑALES SINUSOIDALES SIMULTÁNEAS DE 240 kHz Y 275 kHz		
	SIMULACIÓN	EXPERIMENTAL	Error absoluto	SIMULACIÓN	EXPERIMENTAL	Error absoluto
Posición long. del foco (cm)	4.90	4.90	± 0.00	5.35	5.20	± 0.15
Posición trans. del foco (cm)	0.00	-0.20	± 0.20	0.00	0.40	± 0.40
FWHM _{long} (cm)	1.85	1.75	± 0.10	2.85	3.35	± 0.50
FWHM _{trans} (cm)	0.50	0.55	± 0.05	0.60	0.55	± 0.05

Tabla 4. Comparativa de los resultados de las simulaciones y experimentos en ambas medidas

Comparando los resultados de la simulación y la parte experimental destacamos los errores tan reducidos entre las medidas experimentales y las simulaciones. Cabe decir que estas pequeñas diferencias se deben sobre todo a imprecisiones en el alineamiento del sistema experimental de medida.

Además, se demuestra que tanto en las simulaciones como en los experimentos se produce un efecto de ensanchamiento de foco al comparar la medida de una señal sinusoidal de 250 kHz con la medida de dos señales sinusoidales de 240 kHz y 275 kHz. Para ello, hemos creado dos gráficas (Figura 42) superponiendo los resultados experimentales (azul) y las simulaciones (rojo) en las dos medidas realizadas: una señal de 250 kHz y dos señales simultaneas de 240 kHz y 275 kHz.

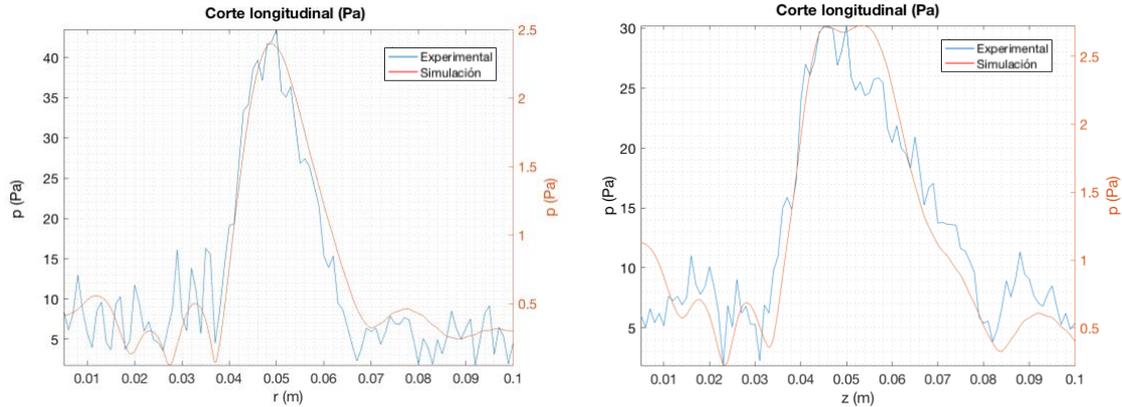


Figura 42. Cortes longitudinales al aplicar una señal de 250 kHz (izquierda) y dos señales simultáneas de 240 kHz y 275 kHz (derecha) superponiendo resultados experimentales (azul) y simulaciones (rojo)

De la misma manera, los cortes transversales superpuestos con los resultados experimentales y las simulaciones (Figura 43) demuestran por una parte la dificultad de alinear perfectamente el origen del eje transversal en las medidas experimentales, y por otra la invariabilidad de $FWHM_{trans}$ comparando las medidas con una señal sinusoidal de 250 kHz y dos señales sinusoidales de 240 kHz y 275 kHz.

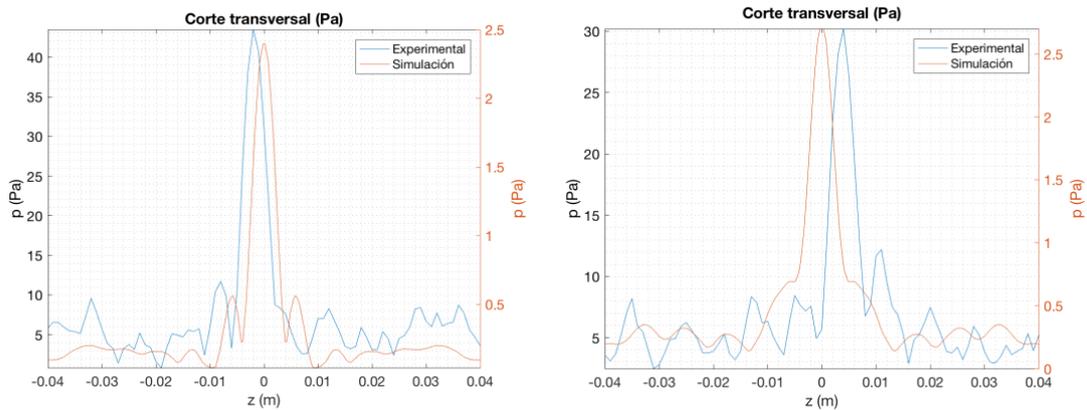


Figura 43. Cortes transversales al aplicar una señal de 250 kHz (izquierda) y dos señales simultáneas de 240 kHz y 275 kHz (derecha) superponiendo resultados experimentales (azul) y simulaciones (rojo)



Capítulo 5. Conclusiones y líneas futuras

Fruto del análisis teórico, simulaciones en Comsol Multiphysics y medidas experimentales realizadas utilizando el sistema Arquímedes del laboratorio de Física de la ETSIT, podemos enunciar varias conclusiones:

- Si la frecuencia de operación cambia respecto al valor de diseño, la localización del foco también se modifica. Además, se desplaza casi linealmente con la frecuencia de funcionamiento, convirtiéndose ésta en un parámetro de control muy dinámico.
- El sistema de control remoto desarrollado para el generador de funciones Agilent 33220A funciona correctamente, lo cual permite definir la señal emitida según las necesidades.
- Se ha comprobado con simulaciones y experimentalmente el efecto de ensanchamiento de foco (aumento de $FWHM_{long}$) para el caso de emisión de dos señales senoidales de 240 kHz y 275 kHz en comparación con una señal sinusoidal de 250 kHz. De la misma manera, se comprueba que $FWHM_{trans}$ permanece prácticamente inalterado de una medida a otra.
- Los cortes transversales superpuestos con los resultados experimentales y las simulaciones demuestran la dificultad de alinear perfectamente el origen del eje transversal en el sistema experimental.
- Comparando los resultados de la simulación y la parte experimental obtenidos con los respectivos programas desarrollados para procesar las medidas, destacamos los errores tan reducidos entre las medidas experimentales y las simulaciones. Estas pequeñas diferencias se deben sobre todo a imprecisiones en el alineamiento del sistema experimental de medida.

Una vez recogidas las conclusiones, podemos finalizar indicando posibles líneas futuras a seguir:

- Mejora del proceso de alineamiento en las medidas experimentales con un dispositivo de medición laser.
- Diseñar y construir una lente Fresnel tipo Rayleigh-Wood, la cual cuenta con una mayor eficiencia, pero también mayor dificultad de diseño.
- Utilizar tecnologías de impresión 3D para construir la misma lente Fresnel con un material distinto al latón como puede ser el ácido poliláctico (PLA). Las lentes de PLA serían compatibles con la técnica de HIFU guiado con resonancia magnética, donde no se pueden utilizar lentes metálicas.
- Finalmente, otra línea a seguir sería la sustitución del generador de funciones Agilent 33220A debido a sus limitaciones comentadas al final del Apartado 4.3. Para ello, se podría adquirir otro generador de funciones más sofisticado o intentar montar un sistema generador de señales cuyo elemento principal sea un DSP, lo cual, en principio, sería una opción más económica.



Capítulo 6. Agradecimientos

Al Ministerio de Educación, Cultura y Deporte de España, porque este proyecto ha estado financiado a través de sus Becas de Colaboración 2017-2018 para alumnos de estudios universitarios.

A mis tutoras Pilar Candelas y Constanza Rubio, por su inestimable ayuda en el desarrollo de este trabajo, ya que en todo momento se mostraron dispuestas a apoyarme en esta etapa final del grado. También quisiera mencionar al doctorando Daniel Tarrazó y al estudiante de máster Sergio Pérez, ambos miembros del Centro de Tecnologías Físicas de la UPV, que me ayudaron a incorporarme a este nuevo grupo.

A mis padres, hermana y familia en general, por su constante apoyo a lo largo de estos años universitarios.

A todas las personas y experiencias que, de una manera u otra, han enriquecido mis estudios del Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación. He de destacar al CERN, por brindarme la increíble posibilidad de formar parte de su equipo gracias a las becas “Summer Student”, etapa en la cual crecí como profesional, pero sobre todo como persona. Para finalizar, no me puedo olvidar de la ETSIT, por ofrecerme estos cuatro años tan fructíferos en conocimiento y madurez.

Capítulo 7. Bibliografía

- [1] José Miguel Fuster, Pilar Candelas, Sergio Castiñeira-Ibáñez, Sergio Pérez-López y Constanza Rubio. Analysis of Fresnel Zone Plates Focusing Dependence on Operating Frequency. *Sensors*, 2017.
- [2] <http://chaosrequiem.com/main/ultrasonidos-que-los-adultos-no-pueden-escuchar/>
- [3] <https://www.nibib.nih.gov/espanol/temas-cientificos/ultrasonido>
- [4] Cao, Q.; Jahns, J. Comprehensive focusing analysis of various Fresnel zone plates. *J. Opt. Soc. Am. A* 2004, 21, 561–571.
- [5] O’Shea, D.C.; Suleski, T.J.; Kathman, A.D.; Prather, D.W. *Diffraction Optics: Design, Fabrication, and Test*; SPIE Press: Bellingham, WA, USA, 2003; p. 260. ISBN 9780819451712.
- [6] Zhang, S.; Yin, L.; Fang, N. Focusing Ultrasound with an Acoustic Metamaterial Network. *Phys. Rev. Lett.* 2009, 102, 194301.
- [7] Hristov, D. *Fresnel Zones in Wireless Links, Zone Plate Lenses and Antennas*, 1st ed.; Artech House, Inc.: Norwood, MA, USA, 2000; ISBN 0890068496.
- [8] Calvo, D.C.; Thangawng, A.L.; Nicholas, M.; Layman, C.N. Thin Fresnel zone plate lenses for focusing underwater sound. *Appl. Phys. Lett.* 2015, 107, 014103.
- [9] Schindel, D.; Bashford, A.; Hutchins, D. Focusing of ultrasonic waves in air using a micromachined Fresnel zone-plate. *Ultrasonics* 1997, 35, 275–285.
- [10] Welter, J.T.; Sathish, S.; Christensen, D.E.; Brodrick, P.G.; Heebl, J.D.; Cherry, M.R. Focusing of longitudinal ultrasonic waves in air with an aperiodic lens. *J. Acoust. Soc. Am.* 2011, 130, 2789–2796.
- [11] Welter, J.T.; Sathish, S.; Dierken, J.M.; Brodrick, P.G.; Cherry, M.R.; Heebl, J.D. Broadband aperiodic air coupled ultrasonic lens. *Appl. Phys. Lett.* 2012, 100, 214102.
- [12] Ter Haar, G.; Coussios, C. High intensity focused ultrasound: Physical principles and devices. *Int. J. Hyperth.* 2007, 23, 89–104.
- [13] Peng, P.; Xiao, B.; Wu, Y. Flat acoustic lens by acoustic grating with curled slits. *Phys. Lett. A* 2014, 378, 3389–3392.
- [14] Kennedy, J.E.; Wu, F.; Ter Haar, G.R.; Gleeson, F.V.; Phillips, R.R.; Middleton, M.R.; Markovich, H.; Filonov, D.; Shishkin, I.; Ginzburg, P. Bifocal Fresnel Lens Based on the Polarization-sensitive metasurface.
- [15] Daniel Tarrazó-Serrano, Pilar Candelas, José Miguel Fuster y Sergio Pérez-López, “Sistema de Posicionamiento y Adquisición de Datos de Alta Precisión para Ultrasonidos en Inmersión Controlada”, *Tecniacústica*, 2017, pp.1-6.
- [16] Hardware de adquisición NI PCI-7330. <http://www.ni.com/pdf/manuals/370837b.pdf>
- [17] Osciloscopio digital (PicoScope 3224: Hoja Características (Farnell)) <http://www.farnell.com/datasheets/37501.pdf>
- [18] Hidrófono Precision Acoustics (1.0mm Needle Hydrophone) <https://www.acoustics.co.uk/product/1mm-needle-hydrophone>
- [19] Manual Imasonic Transducer 250kHz
- [20] Pulser Panametrics 5077PR. http://www.olympusims.com/es/.downloads/download/?file=285213012&fl=en_US
- [21] Preamplificador Preamplifier. <http://www.acoustics.co.uk/product/preamplifier/>
- [22] DC Coupler Precision Acoustics. <http://www.acoustics.co.uk/product/dc-coupler/>
- [23] P. Candelas, J. M. Fuster, C. Rubio, S. Castiñeira-Ibáñez, and D. Tarrazó-Serrano, “Analysis and Design of Fresnel Zone Plates with Multiple Focus,” 173rd Meeting of the Acoustical Society of America and the 8th Forum Acusticum, 2017.
- [24] Driver Instrument Control Toolbox Agilent 33220A <https://es.mathworks.com/matlabcentral/fileexchange/12863-agilent-33220a-function-generator-driver>
- [25] Using MATLAB with Agilent Instruments: <https://es.mathworks.com/videos/using-matlab-with-agilent-instruments-68775.html>



- [26] GWInstek 1054B Oscilloscope: http://www.gwinstek.com/en-global/products/Oscilloscopes/Digital_Storage_Oscilloscopes/GDS-1000B
- [27] <https://es.mathworks.com/help/supportpkg/nivisaandicp/ug/quick-control-function-generator-functions.html>
- [28] <https://es.mathworks.com/help/supportpkg/nivisaandicp/ug/quick-control-function-generator-properties.html>
- [29] Guillem, M. Instrumentación Biomédica, Tema 10: Imagen médica por ultrasonidos.
- [30] <http://www.alarsis.com/catalogo/fr150/>
- [31] O.C. Zienkiewicz y R. L. Taylor: The Finite Element Method: Its Basis and Fundamentals.
- [32] Manual del Robot de Medida
- [33] <https://web.sonoma.edu/ese/manuals/33220-90002.pdf>
- [34] https://es.wikipedia.org/wiki/Método_de_los_elementos_finitos
- [35] https://es.wikipedia.org/wiki/Anchura_a_media_altura

Capítulo 8. Anexos

8.1 Procesado de simulaciones de Comsol Multiphysics con App Designer de Matlab.

```
classdef procesarSimu < matlab.apps.AppBase
    % Properties that correspond to app components
    properties (Access = public)
        AplicacionparaprocesarsimulacionesdeComsolMultiphysicsUIFigure matlab.ui.Figure
        SeleccionarButton matlab.ui.control.Button
        Directoriotexto matlab.ui.control.EditField
        DirectoriodelassimulacionesLabel matlab.ui.control.Label
        NdesinusoidesButtonGroup matlab.ui.container.ButtonGroup
        Button matlab.ui.control.RadioButton
        Button_2 matlab.ui.control.RadioButton
        Button_3 matlab.ui.control.RadioButton
        FrecuenciakHzsinusoide1EditFieldLabel matlab.ui.control.Label
        FrecuenciakHzsinusoide1EditField matlab.ui.control.NumericEditField
        FrecuenciakHzsinusoide2EditFieldLabel matlab.ui.control.Label
        FrecuenciakHzsinusoide2EditField matlab.ui.control.NumericEditField
        FrecuenciakHzsinusoide3EditFieldLabel matlab.ui.control.Label
        FrecuenciakHzsinusoide3EditField matlab.ui.control.NumericEditField
        ProcesarSimulacionesButton matlab.ui.control.Button
        PlanodegananciaycortesendBSwitchLabel matlab.ui.control.Label
        PlanodegananciaycortesendBSwitch matlab.ui.control.Switch
    end

    methods (Access = private)
        % Button pushed function: SeleccionarButton
        function SeleccionarButtonPushed(app, event)
            %Cargar directorio donde estan las simulaciones
            folder=uigetdir();
            app.Directoriotexto.Value=folder;
        end

        % Button pushed function: ProcesarSimulacionesButton
        function ProcesarSimulacionesButtonPushed(app, event)
            %Comprobar cuantas señales se van a procesar y cargar sus frecuencias

            if (isequal(app.FrecuenciakHzsinusoide1EditField.Editable,'off'))
                f1=0;
            else
                f1=app.FrecuenciakHzsinusoide1EditField.Value;
            end

            if (isequal(app.FrecuenciakHzsinusoide2EditField.Editable,'off'))
                f2=0;
            else
                f2=app.FrecuenciakHzsinusoide2EditField.Value;
            end
            if (isequal(app.FrecuenciakHzsinusoide3EditField.Editable,'off'))
                f3=0;
            else
                f3=app.FrecuenciakHzsinusoide3EditField.Value;
            end

            %Cargar las señales de las frecuencias seleccionadas
            senyal1=load([app.Directoriotexto.Value,'/',num2str(f1),'.txt']);
            senyal1sinLente=load([app.Directoriotexto.Value,'/',num2str(f1),'.sin.txt']);

            selectedButton = app.NdesinusoidesButtonGroup.SelectedObject;

            if (selectedButton==app.Button || selectedButton==app.Button_2)
                senyal2=load([app.Directoriotexto.Value,'/',num2str(f2),'.txt']);
                senyal2sinLente=load([app.Directoriotexto.Value,'/',num2str(f2),'.sin.txt']);
            end

            if (selectedButton==app.Button)
                senyal3=load([app.Directoriotexto.Value,'/',num2str(f3),'.txt']);
                senyal3sinLente=load([app.Directoriotexto.Value,'/',num2str(f3),'.sin.txt']);
            end

            Rd = senyal1(:,1);%Para almacenar los datos R o Z elegimos una de las señales, puesto que seran los mismos
            Zd = senyal1(:,2);

            %Se almacenan los datos de presion de las señales seleccionadas
            if (selectedButton==app.Button)
                Id = senyal1(:,3) + senyal2(:,3) + senyal3(:,3);
            elseif (selectedButton==app.Button_2)
                Id = senyal1(:,3) + senyal2(:,3);
            else
                Id = senyal1(:,3);
            end
        end
    end
end
```



```
Rdsin=senal1sinLente(:,1);
Zdsin=senal1sinLente(:,2);

%Se almacenan los datos de presion sin lente de las señales seleccionadas
if (selectedButton==app.Button)
    Idsin=senal1sinLente(:,3) + senyal2sinLente(:,3) + senyal3sinLente(:,3);
elseif (selectedButton==app.Button_2)
    Idsin=senal1sinLente(:,3) + senyal2sinLente(:,3);
else
    Idsin=senal1sinLente(:,3);
end

%Se calculan los maximos de las coordenadas r y z
maxR = max(max(Rd));
maxZ = max(max(Zd));

%% Gridding
resolucion = 1000;

R = linspace(-maxR,maxR,resolucion);
Z = linspace(0,maxZ,resolucion);
[Rq,Zq] = meshgrid(R,Z);

Iq = griddata(Rd,Zd,Id,Rq,Zq);
Iqsin = griddata(Rdsin,Zdsin,Idsin,Rq,Zq);

LIqres=Iq;

posicionR=length(LIqres)/2; %Encontramos el centro del eje R.
[maxim,posicionZ]=max(LIqres(:,posicionR));%Encontramos el maximo del eje Z en el punto del eje R hallado en el paso anterior.

mkdir([app.Directoriotexto.Value,'/graficasProcesado']);
scrsz = get(0,'ScreenSize');
figure('Position',[1 scrsz(4)/2 scrsz(3)/2 scrsz(4)/2-90])
surf(Rq,Zq,Iq);
title('250kHz con lente (Pa)');
colormap jet;
shading flat;
axis tight;
view(0,90);
ylabel('z (m)');
xlabel('r (m)');
xlim([-0.05 0.05]);
ylim([0 0.1]);
caxis([0 2.6]);
colorbar;
saveas(gcf,[app.Directoriotexto.Value,'/graficasProcesado/plano con lente.fig']);

figure('Position',[scrsz(3) 1 scrsz(3)/2 scrsz(4)/2-90])
surf(Rq,Zq,Iqsin);
title('250kHz sin lente (Pa)');
colormap jet;
shading flat;
axis tight;
view(0,90);
ylabel('z (m)');
xlabel('r (m)');
xlim([-0.05 0.05]);
ylim([0 0.1]);
caxis([0 2.6]);
colorbar;
saveas(gcf,[app.Directoriotexto.Value,'/graficasProcesado/plano sin lente.fig']);

figure('Position',[1 1 scrsz(3)/2 scrsz(4)/2-90])
plot(Zq,LIqres(:,posicionR));
title('Corte longitudinal (Pa)');
```

```

ylabel('Presión acústica (Pa)');
colormap jet;
shading flat;
axis tight;
view(0,90);
xlabel('z (m)');
xlim([0 0.1]);
saveas(gcf,[app.Directoriotexto.Value,'/graficasProcesado/longitudinal.fig']);

figure('Position',[scrsz(3) 1 scrsz(3)/2 scrsz(4)/2-90])
plot(Rq(1,:),LIqres(posicionZ,:));
title('Corte transversal (Pa)');
ylabel('Presión acústica (Pa)');
colormap jet;
shading flat;
axis tight;
view(0,90);
xlabel('r (m)');
xlim([-0.04 0.04]);
saveas(gcf,[app.Directoriotexto.Value,'/graficasProcesado/transversal.fig']);

%Valores en dB y ganancia
if (isequal(app.PlanodegananciaycortesendB5Switch.Value,'On'))

    LIq = 20*log10(Iq);
    LIqsin = 20*log10(Iqsin);

    LIqres=LIq-LIqsin;

    figure('Position',[scrsz(3) scrsz(4)/2 scrsz(3)/2 scrsz(4)/2-90])
    surf(Rq,Zq,LIqres);
    title('Ganancia (dB)');
    colormap jet;
    shading flat;
    axis tight;
    view(0,90);
    ylabel('z (m)');
    xlabel('r (m)');
    xlim([-0.05 0.05]);
    ylim([0 0.1]);
    colorbar;
    saveas(gcf,[app.Directoriotexto.Value,'/graficasProcesado/ganancia(dB).fig']);

    figure('Position',[scrsz(3) 1 scrsz(3)/2 scrsz(4)/2-90])
    plot(Rq(1,:),LIqres(posicionZ,:));
    title('Ganancia en corte transversal (dB)');
    ylabel('Presión acústica (dB)');
    colormap jet;
    shading flat;
    axis tight;
    view(0,90);
    xlabel('r (m)');
    xlim([-0.04 0.04]);
    saveas(gcf,[app.Directoriotexto.Value,'/graficasProcesado/transversal(dB).fig']);

    figure('Position',[1 1 scrsz(3)/2 scrsz(4)/2-90])
    plot(Zq,LIqres(:,posicionR));
    title('Ganancia en corte longitudinal (dB)');
    ylabel('Presión acústica (dB)');
    colormap jet;
    shading flat;
    axis tight;
    view(0,90);
    xlabel('z (m)');
    xlim([0 0.1]);
    saveas(gcf,[app.Directoriotexto.Value,'/graficasProcesado/longitudinal(dB).fig']);

end

end

% Selection changed function: NdesinusoidesButtonGroup
function NdesinusoidesButtonGroupSelectionChanged(app, event)
%Altera la posibilidad de edicion de los campos de frecuencia segun la cantidad de señales seleccionadas
selectedButton = app.NdesinusoidesButtonGroup.SelectedObject;
if (selectedButton==app.Button)
    app.FrecuenciakHzsinusoide1EditField.Editable='on';
    app.FrecuenciakHzsinusoide2EditField.Editable='on';
    app.FrecuenciakHzsinusoide3EditField.Editable='on';

elseif (selectedButton==app.Button_2)
    app.FrecuenciakHzsinusoide1EditField.Editable='on';
    app.FrecuenciakHzsinusoide2EditField.Editable='on';
    app.FrecuenciakHzsinusoide3EditField.Editable='off';
else
    app.FrecuenciakHzsinusoide1EditField.Editable='on';
    app.FrecuenciakHzsinusoide2EditField.Editable='off';
    app.FrecuenciakHzsinusoide3EditField.Editable='off';
end
end
end
end

```

```
% App initialization and construction
methods (Access = private)

% Create UIFigure and components
function createComponents(app)

% Create AplicacinparaprocesarsimulacionesdeComsolMultiphysicsUIFigure
app.AplicacinparaprocesarsimulacionesdeComsolMultiphysicsUIFigure = uifigure;
app.AplicacinparaprocesarsimulacionesdeComsolMultiphysicsUIFigure.Color = [0.302 0.749 0.9294];
app.AplicacinparaprocesarsimulacionesdeComsolMultiphysicsUIFigure.Colormap = [0.2431 0.149 0.6588;0.251 0.1647 0.7059;0.2588 0.1804 0.7529];
app.AplicacinparaprocesarsimulacionesdeComsolMultiphysicsUIFigure.Position = [100 100 651 480];
app.AplicacinparaprocesarsimulacionesdeComsolMultiphysicsUIFigure.Name = 'Aplicación para procesar simulaciones de ComsolMultiphysics';

% Create SeleccionarButton
app.SeleccionarButton = uibutton(app.AplicacinparaprocesarsimulacionesdeComsolMultiphysicsUIFigure, 'push');
app.SeleccionarButton.ButtonPushedFcn = createCallbackFcn(app, @SeleccionarButtonPushed, true);
app.SeleccionarButton.Position = [519 408 100 22];
app.SeleccionarButton.Text = 'Seleccionar';

% Create Directoriotexto
app.Directoriotexto = uieditfield(app.AplicacinparaprocesarsimulacionesdeComsolMultiphysicsUIFigure, 'text');
app.Directoriotexto.Position = [50 408 433 22];

% Create DirectoriodelassimulacionesLabel
app.DirectoriodelassimulacionesLabel = uilabel(app.AplicacinparaprocesarsimulacionesdeComsolMultiphysicsUIFigure);
app.DirectoriodelassimulacionesLabel.Position = [50 439 166 15];
app.DirectoriodelassimulacionesLabel.Text = 'Directorio de las simulaciones';

% Create NdesinusoidesButtonGroup
app.NdesinusoidesButtonGroup = uibuttongroup(app.AplicacinparaprocesarsimulacionesdeComsolMultiphysicsUIFigure);
app.NdesinusoidesButtonGroup.SelectionChangedFcn = createCallbackFcn(app, @NdesinusoidesButtonGroupSelectionChanged, true);
app.NdesinusoidesButtonGroup.Title = 'Nº de sinusoides';
app.NdesinusoidesButtonGroup.BackgroundColor = [0.9412 0.9412 0.9412];
app.NdesinusoidesButtonGroup.Position = [111 273 123 106];

% Create Button
app.Button = uiradiobutton(app.NdesinusoidesButtonGroup);
app.Button.Text = '3';
app.Button.Position = [11 60 58 15];
app.Button.Value = true;

% Create Button_2
app.Button_2 = uiradiobutton(app.NdesinusoidesButtonGroup);
app.Button_2.Text = '2';
app.Button_2.Position = [11 38 65 15];

% Create Button_3
app.Button_3 = uiradiobutton(app.NdesinusoidesButtonGroup);
app.Button_3.Text = '1';
app.Button_3.Position = [11 16 65 15];

% Create FrecuenciakHzsinusoide1EditFieldLabel
app.FrecuenciakHzsinusoide1EditFieldLabel = uilabel(app.AplicacinparaprocesarsimulacionesdeComsolMultiphysicsUIFigure);
app.FrecuenciakHzsinusoide1EditFieldLabel.HorizontalAlignment = 'right';
app.FrecuenciakHzsinusoide1EditFieldLabel.Position = [345 361 159 15];
app.FrecuenciakHzsinusoide1EditFieldLabel.Text = 'Frecuencia (kHz) sinusoide 1';

% Create FrecuenciakHzsinusoide1EditField
app.FrecuenciakHzsinusoide1EditField = uieditfield(app.AplicacinparaprocesarsimulacionesdeComsolMultiphysicsUIFigure, 'numeric');
app.FrecuenciakHzsinusoide1EditField.Position = [519 357 100 22];

% Create FrecuenciakHzsinusoide2EditFieldLabel
app.FrecuenciakHzsinusoide2EditFieldLabel = uilabel(app.AplicacinparaprocesarsimulacionesdeComsolMultiphysicsUIFigure);
app.FrecuenciakHzsinusoide2EditFieldLabel.HorizontalAlignment = 'right';
app.FrecuenciakHzsinusoide2EditFieldLabel.Position = [345 294 159 15];
app.FrecuenciakHzsinusoide2EditFieldLabel.Text = 'Frecuencia (kHz) sinusoide 2';

% Create FrecuenciakHzsinusoide2EditField
app.FrecuenciakHzsinusoide2EditField = uieditfield(app.AplicacinparaprocesarsimulacionesdeComsolMultiphysicsUIFigure, 'numeric');
app.FrecuenciakHzsinusoide2EditField.Position = [519 290 100 22];

% Create FrecuenciakHzsinusoide3EditFieldLabel
app.FrecuenciakHzsinusoide3EditFieldLabel = uilabel(app.AplicacinparaprocesarsimulacionesdeComsolMultiphysicsUIFigure);
app.FrecuenciakHzsinusoide3EditFieldLabel.HorizontalAlignment = 'right';
app.FrecuenciakHzsinusoide3EditFieldLabel.Position = [345 229 159 15];
app.FrecuenciakHzsinusoide3EditFieldLabel.Text = 'Frecuencia (kHz) sinusoide 3';

% Create FrecuenciakHzsinusoide3EditField
app.FrecuenciakHzsinusoide3EditField = uieditfield(app.AplicacinparaprocesarsimulacionesdeComsolMultiphysicsUIFigure, 'numeric');
app.FrecuenciakHzsinusoide3EditField.Position = [519 225 100 22];

% Create ProcesarSimulacionesButton
app.ProcesarSimulacionesButton = uibutton(app.AplicacinparaprocesarsimulacionesdeComsolMultiphysicsUIFigure, 'push');
app.ProcesarSimulacionesButton.ButtonPushedFcn = createCallbackFcn(app, @ProcesarSimulacionesButtonPushed, true);
app.ProcesarSimulacionesButton.BackgroundColor = [1 0 0];
app.ProcesarSimulacionesButton.FontColor = [1 1 1];
app.ProcesarSimulacionesButton.Position = [277 104 138 22];
app.ProcesarSimulacionesButton.Text = 'Procesar Simulaciones';

% Create PlanodegananciaycortesendBSwitchLabel
app.PlanodegananciaycortesendBSwitchLabel = uilabel(app.AplicacinparaprocesarsimulacionesdeComsolMultiphysicsUIFigure);
app.PlanodegananciaycortesendBSwitchLabel.HorizontalAlignment = 'center';
app.PlanodegananciaycortesendBSwitchLabel.Position = [59 229 191 15];
app.PlanodegananciaycortesendBSwitchLabel.Text = 'Plano de ganancia y cortes en dB ';

% Create PlanodegananciaycortesendBSwitch
app.PlanodegananciaycortesendBSwitch = uiswitch(app.AplicacinparaprocesarsimulacionesdeComsolMultiphysicsUIFigure, 'slider');
app.PlanodegananciaycortesendBSwitch.Position = [131 201 45 20];
```



```
end
end

methods (Access = public)

% Construct app
function app = procesarSimu

% Create and configure components
createComponents(app)

% Register the app with App Designer
registerApp(app, app.AplicacinparaprocesarsimulacionesdeComsolMultiphysicsUIFigure)

if nargin == 0
clear app
end
end

% Code that executes before app deletion
function delete(app)

% Delete UIFigure when app is deleted
delete(app.AplicacinparaprocesarsimulacionesdeComsolMultiphysicsUIFigure)
end
end
end
```

8.2 Control remoto del generador de funciones Agilent 33220A con App Designer.

```
classdef arbWaveform < matlab.apps.AppBase

    % Properties that correspond to app components
    properties (Access = public)
        ControlRemotoGeneradordeFuncionesAgilent33220AUIFigure matlab.ui.Figure
        UIAxes matlab.ui.control.UIAxes
        Lamp matlab.ui.control.Lamp
        IniciarconexinremotaSwitchLabel matlab.ui.control.Label
        IniciarconexinremotaSwitch matlab.ui.control.ToggleSwitch
        EnviarfuncinButton matlab.ui.control.Button
        AmplitudVsinusoide1EditFieldLabel matlab.ui.control.Label
        AmplitudVsinusoide1EditField matlab.ui.control.NumericEditField
        FrecuenciakHzsinusoide1EditFieldLabel matlab.ui.control.Label
        FrecuenciakHzsinusoide1EditField matlab.ui.control.NumericEditField
        AmplitudVsinusoide2EditFieldLabel matlab.ui.control.Label
        AmplitudVsinusoide2EditField matlab.ui.control.NumericEditField
        FrecuenciakHzsinusoide2EditFieldLabel matlab.ui.control.Label
        FrecuenciakHzsinusoide2EditField matlab.ui.control.NumericEditField
        MododeondaSwitchLabel matlab.ui.control.Label
        MododeondaSwitch matlab.ui.control.ToggleSwitch
        NdesinusoideosButtonGroup matlab.ui.container.ButtonGroup
        Button matlab.ui.control.RadioButton
        Button_2 matlab.ui.control.RadioButton
        Button_3 matlab.ui.control.RadioButton
        AmplitudVsinusoide3EditFieldLabel matlab.ui.control.Label
        AmplitudVsinusoide3EditField matlab.ui.control.NumericEditField
        FrecuenciakHzsinusoide3EditField_2Label matlab.ui.control.Label
        FrecuenciakHzsinusoide3EditField matlab.ui.control.NumericEditField
        OutputSwitchLabel matlab.ui.control.Label
        OutputSwitch matlab.ui.control.ToggleSwitch
    end

    properties (Access = private)
        MyFGen=fgen(); % Description
    end

    methods (Access = private)

        % Button pushed function: EnviarfuncinButton
        function EnviarfuncinButtonPushed(app, event)
            %Almacenar valores de amplitud y frecuencia segun la cantidad de señales
            if (isequal(app.AmplitudVsinusoide1EditField.Editable,'off'))
                a1=0;
            else
                a1=app.AmplitudVsinusoide1EditField.Value;
            end

            if (isequal(app.AmplitudVsinusoide2EditField.Editable,'off'))
                a2=0;
            else
                a2=app.AmplitudVsinusoide2EditField.Value;
            end

            if (isequal(app.AmplitudVsinusoide3EditField.Editable,'off'))
                a3=0;
            else
                a3=app.AmplitudVsinusoide3EditField.Value;
            end

            if (isequal(app.FrecuenciakHzsinusoide1EditField.Editable,'off'))
                f1=0;
            else
                f1=app.FrecuenciakHzsinusoide1EditField.Value;
            end

            if (isequal(app.FrecuenciakHzsinusoide2EditField.Editable,'off'))
                f2=0;
            else
                f2=app.FrecuenciakHzsinusoide2EditField.Value;
            end

            if(isequal(app.FrecuenciakHzsinusoide3EditField.Editable,'off'))
                f3=0;
            else
                f3=app.FrecuenciakHzsinusoide3EditField.Value;
            end

            t1=0:0.0001:1;
        end
    end
end
```



```
waveform1 = (a1/2)*sin(2*pi*f1*t1);  
waveform2 = (a2/2)*sin(2*pi*f2*t1);  
waveform3 = (a3/2)*sin(2*pi*f3*t1);  
  
waveformArray = waveform1+waveform2+waveform3;  
  
plot(app.UIAxes,t1,waveformArray,'LineWidth',2)  
app.UIAxes.Title.String='Función enviada al generador';  
  
app.MyFGen.Frequency=1000;%Habilitamos una frecuencia de 1kHz como referencia.  
app.MyFGen.Waveform='Arb';  
downloadWaveform(app.MyFGen,waveformArray);
```

```
end
```

```
% Value changed function: IniciarconexinremotaSwitch  
function IniciarconexinremotaSwitchValueChanged(app, event)  
value = app.IniciarconexinremotaSwitch.Value;
```

```
if(isequal(value,'Off'))  
    app.Lamp.Color='red';  
    disconnect(app.MyFGen);  
elseif (isequal(value,'On'))  
    app.MyFGen.Resource='USB0::0x0957::0x0407::MY44018984::INSTR';  
    connect(app.MyFGen);  
    app.Lamp.Color='green';  
end
```

```
end
```

```
% Value changed function: IniciarconexinremotaSwitch  
function IniciarconexinremotaSwitchValueChanged(app, event)  
value = app.IniciarconexinremotaSwitch.Value;
```

```
%Conectar o desconectar conexion remota con generador  
if(isequal(value,'Off'))  
    app.Lamp.Color='red';  
    disconnect(app.MyFGen);  
else  
    app.MyFGen.Resource='USB0::0x0957::0x0407::MY44018984::INSTR';  
    connect(app.MyFGen);  
    app.Lamp.Color='green';  
end
```

```
end
```

```
% Value changed function: MododeondaSwitch  
function MododeondaSwitchValueChanged(app, event)  
value = app.MododeondaSwitch.Value;
```

```
%Modo de onda continuo o burst  
if(isequal(value,'Continuous'))  
    app.MyFGen.Mode='continuous';  
else  
    app.MyFGen.Mode='burst';  
    app.MyFGen.BurstCount=1;  
    app.MyFGen.TriggerRate=100e-3;
```

```
end
```

```
end
```

```
% Selection changed function: NdesinusoidesButtonGroup  
function NdesinusoidesButtonGroupSelectionChanged(app, event)  
selectedButton = app.NdesinusoidesButtonGroup.SelectedObject;
```

```
%Alterar la edicion de los campos de frecuencia y amplitud segun la cantidad de senyales que queremos  
if (selectedButton==app.Button)  
    app.AmplitudVsinusoide1EditField.Editable='on';  
    app.FrecuenciaiKHzsinusoide1EditField.Editable='on';  
    app.AmplitudVsinusoide2EditField.Editable='on';  
    app.FrecuenciaiKHzsinusoide2EditField.Editable='on';  
    app.AmplitudVsinusoide3EditField.Editable='on';  
    app.FrecuenciaiKHzsinusoide3EditField.Editable='on';  
  
elseif (selectedButton==app.Button_2)  
    app.AmplitudVsinusoide1EditField.Editable='on';  
    app.FrecuenciaiKHzsinusoide1EditField.Editable='on';  
    app.AmplitudVsinusoide2EditField.Editable='on';  
    app.FrecuenciaiKHzsinusoide2EditField.Editable='on';
```

```
app.AmplitudVsinusoide3EditField.Editable='off';
app.FrecuenciakHzsinusoide3EditField.Editable='off';
else
app.AmplitudVsinusoide1EditField.Editable='on';
app.FrecuenciakHzsinusoide1EditField.Editable='on';
app.AmplitudVsinusoide2EditField.Editable='off';
app.FrecuenciakHzsinusoide2EditField.Editable='off';
app.AmplitudVsinusoide3EditField.Editable='off';
app.FrecuenciakHzsinusoide3EditField.Editable='off';
end
end

% Value changed function: OutputSwitch
function OutputSwitchValueChanged(app, event)
value = app.OutputSwitch.Value;

%Habilitar o no la salida del generador
if(isequal(value,'Off'))
disableOutput(app.MyFGen);
else
enableOutput(app.MyFGen);
end
end

end

% App initialization and construction
methods (Access = private)

% Create UIFigure and components
function createComponents(app)

% Create ControlRemotoGeneradordeFuncionesAgilent33220AUIFigure
app.ControlRemotoGeneradordeFuncionesAgilent33220AUIFigure = uifigure;
app.ControlRemotoGeneradordeFuncionesAgilent33220AUIFigure.Color = [0.302 0.749 0.9294];
app.ControlRemotoGeneradordeFuncionesAgilent33220AUIFigure.Position = [100 100 731 571];
app.ControlRemotoGeneradordeFuncionesAgilent33220AUIFigure.Name = 'Control Remoto Generador de Funciones Agilent 33220A';

% Create UIAxes
app.UIAxes = uiaxes(app.ControlRemotoGeneradordeFuncionesAgilent33220AUIFigure);
title(app.UIAxes, 'Función enviada al generador')
xlabel(app.UIAxes, 't(s)')
ylabel(app.UIAxes, 'V(mV)')
app.UIAxes.XLim = [0 0.1];
app.UIAxes.XColor = [0 0 0];
app.UIAxes.Position = [223 29 347 216];

% Create Lamp
app.Lamp = uilamp(app.ControlRemotoGeneradordeFuncionesAgilent33220AUIFigure);
app.Lamp.Position = [137 440 27 27];
app.Lamp.Color = [1 0 0];

% Create IniciarconexinremotaSwitchLabel
app.IniciarconexinremotaSwitchLabel = uilabel(app.ControlRemotoGeneradordeFuncionesAgilent33220AUIFigure);
app.IniciarconexinremotaSwitchLabel.HorizontalAlignment = 'center';
app.IniciarconexinremotaSwitchLabel.Position = [38 515 131 15];
app.IniciarconexinremotaSwitchLabel.Text = 'Iniciar conexión remota';

% Create IniciarconexinremotaSwitch
app.IniciarconexinremotaSwitch = uiswitch(app.ControlRemotoGeneradordeFuncionesAgilent33220AUIFigure, 'toggle');
app.IniciarconexinremotaSwitch.ValueChangedFcn = createCallbackFcn(app, @IniciarconexinremotaSwitchValueChanged, true);
app.IniciarconexinremotaSwitch.Position = [93 435 20 45];

% Create EnviarfuncinButton
app.EnviarfuncinButton = uibutton(app.ControlRemotoGeneradordeFuncionesAgilent33220AUIFigure, 'push');
app.EnviarfuncinButton.ButtonPushedFcn = createCallbackFcn(app, @EnviarfuncinButtonPushed, true);
app.EnviarfuncinButton.BackgroundColor = [1 0 0];
app.EnviarfuncinButton.FontColor = [1 1 1];
app.EnviarfuncinButton.Position = [581 344 100 22];
app.EnviarfuncinButton.Text = 'Enviar función';

% Create AmplitudVsinusoide1EditFieldLabel
app.AmplitudVsinusoide1EditFieldLabel = uilabel(app.ControlRemotoGeneradordeFuncionesAgilent33220AUIFigure);
app.AmplitudVsinusoide1EditFieldLabel.HorizontalAlignment = 'right';
app.AmplitudVsinusoide1EditFieldLabel.Position = [256 516 135 15];
app.AmplitudVsinusoide1EditFieldLabel.Text = 'Amplitud (V) sinusoide 1';

% Create AmplitudVsinusoide1EditField
app.AmplitudVsinusoide1EditField = uieditfield(app.ControlRemotoGeneradordeFuncionesAgilent33220AUIFigure, 'numeric');
app.AmplitudVsinusoide1EditField.Position = [406 512 100 22];

% Create FrecuenciakHzsinusoide1EditFieldLabel
app.FrecuenciakHzsinusoide1EditFieldLabel = uilabel(app.ControlRemotoGeneradordeFuncionesAgilent33220AUIFigure);
app.FrecuenciakHzsinusoide1EditFieldLabel.HorizontalAlignment = 'right';
app.FrecuenciakHzsinusoide1EditFieldLabel.Position = [232 471 159 15];
app.FrecuenciakHzsinusoide1EditFieldLabel.Text = 'Frecuencia (kHz) sinusoide 1';

% Create FrecuenciakHzsinusoide1EditField
app.FrecuenciakHzsinusoide1EditField = uieditfield(app.ControlRemotoGeneradordeFuncionesAgilent33220AUIFigure, 'numeric');
app.FrecuenciakHzsinusoide1EditField.Position = [406 467 100 22];

% Create AmplitudVsinusoide2EditFieldLabel
app.AmplitudVsinusoide2EditFieldLabel = uilabel(app.ControlRemotoGeneradordeFuncionesAgilent33220AUIFigure);
app.AmplitudVsinusoide2EditFieldLabel.HorizontalAlignment = 'right';
```

```
app.AmplitudVsinusoide2EditFieldLabel.Position = [256 426 135 15];
app.AmplitudVsinusoide2EditFieldLabel.Text = 'Amplitud (V) sinusoide 2';

% Create AmplitudVsinusoide2EditField
app.AmplitudVsinusoide2EditField = uieditfield(app.ControlRemotoGeneradordeFuncionesAgilent33220AUIFigure, 'numeric');
app.AmplitudVsinusoide2EditField.Position = [406 422 100 22];

% Create FrecuenciakHsinusoide2EditFieldLabel
app.FrecuenciakHsinusoide2EditFieldLabel = uilabel(app.ControlRemotoGeneradordeFuncionesAgilent33220AUIFigure);
app.FrecuenciakHsinusoide2EditFieldLabel.HorizontalAlignment = 'right';
app.FrecuenciakHsinusoide2EditFieldLabel.Position = [232 381 159 15];
app.FrecuenciakHsinusoide2EditFieldLabel.Text = 'Frecuencia (kHz) sinusoide 2';

% Create FrecuenciakHsinusoide2EditField
app.FrecuenciakHsinusoide2EditField = uieditfield(app.ControlRemotoGeneradordeFuncionesAgilent33220AUIFigure, 'numeric');
app.FrecuenciakHsinusoide2EditField.Position = [406 377 100 22];

% Create MododeondaSwitchLabel
app.MododeondaSwitchLabel = uilabel(app.ControlRemotoGeneradordeFuncionesAgilent33220AUIFigure);
app.MododeondaSwitchLabel.HorizontalAlignment = 'center';
app.MododeondaSwitchLabel.Position = [541 515 84 15];
app.MododeondaSwitchLabel.Text = 'Modo de onda';

% Create MododeondaSwitch
app.MododeondaSwitch = uiswitch(app.ControlRemotoGeneradordeFuncionesAgilent33220AUIFigure, 'toggle');
app.MododeondaSwitch.Items = {'Continuous', 'Burst'};
app.MododeondaSwitch.ValueChangedFcn = createCallbackFcn(app, @MododeondaSwitchValueChanged, true);
app.MododeondaSwitch.Position = [573 434 20 45];
app.MododeondaSwitch.Value = 'Continuous';

% Create NdesinusoidesButtonGroup
app.NdesinusoidesButtonGroup = uibuttongroup(app.ControlRemotoGeneradordeFuncionesAgilent33220AUIFigure);
app.NdesinusoidesButtonGroup.SelectionChangedFcn = createCallbackFcn(app, @NdesinusoidesButtonGroupSelectionChanged, true);
app.NdesinusoidesButtonGroup.Title = 'Nº de sinusoides';
app.NdesinusoidesButtonGroup.BackgroundColor = [0.9412 0.9412 0.9412];
app.NdesinusoidesButtonGroup.Position = [41 288 123 106];

% Create Button
app.Button = uiradiobutton(app.NdesinusoidesButtonGroup);
app.Button.Text = '3';
app.Button.Position = [11 60 58 15];
app.Button.Value = true;

% Create Button_2
app.Button_2 = uiradiobutton(app.NdesinusoidesButtonGroup);
app.Button_2.Text = '2';
app.Button_2.Position = [11 38 65 15];
% Create Button_3
app.Button_3 = uiradiobutton(app.NdesinusoidesButtonGroup);
app.Button_3.Text = '1';
app.Button_3.Position = [11 16 65 15];

% Create AmplitudVsinusoide3EditFieldLabel
app.AmplitudVsinusoide3EditFieldLabel = uilabel(app.ControlRemotoGeneradordeFuncionesAgilent33220AUIFigure);
app.AmplitudVsinusoide3EditFieldLabel.HorizontalAlignment = 'right';
app.AmplitudVsinusoide3EditFieldLabel.Position = [256 336 135 15];
app.AmplitudVsinusoide3EditFieldLabel.Text = 'Amplitud (V) sinusoide 3';

% Create AmplitudVsinusoide3EditField
app.AmplitudVsinusoide3EditField = uieditfield(app.ControlRemotoGeneradordeFuncionesAgilent33220AUIFigure, 'numeric');
app.AmplitudVsinusoide3EditField.Position = [406 332 100 22];

% Create FrecuenciakHsinusoide3EditField_2Label
app.FrecuenciakHsinusoide3EditField_2Label = uilabel(app.ControlRemotoGeneradordeFuncionesAgilent33220AUIFigure);
app.FrecuenciakHsinusoide3EditField_2Label.HorizontalAlignment = 'right';
app.FrecuenciakHsinusoide3EditField_2Label.Position = [232 292 159 15];
app.FrecuenciakHsinusoide3EditField_2Label.Text = 'Frecuencia (kHz) sinusoide 3';

% Create FrecuenciakHsinusoide3EditField
app.FrecuenciakHsinusoide3EditField = uieditfield(app.ControlRemotoGeneradordeFuncionesAgilent33220AUIFigure, 'numeric');
app.FrecuenciakHsinusoide3EditField.Position = [406 288 100 22];

% Create OutputSwitchLabel
app.OutputSwitchLabel = uilabel(app.ControlRemotoGeneradordeFuncionesAgilent33220AUIFigure);
app.OutputSwitchLabel.HorizontalAlignment = 'center';
app.OutputSwitchLabel.Position = [636 516 84 15];
app.OutputSwitchLabel.Text = 'Output';

% Create OutputSwitch
app.OutputSwitch = uiswitch(app.ControlRemotoGeneradordeFuncionesAgilent33220AUIFigure, 'toggle');
app.OutputSwitch.ValueChangedFcn = createCallbackFcn(app, @OutputSwitchValueChanged, true);
app.OutputSwitch.Position = [668 435 20 45];

end
end

methods (Access = public)

% Construct app
function app = arbWaveform

% Create and configure components
createComponents(app)
```



```
% Register the app with App Designer
registerApp(app, app.ControlRemotoGeneradordeFuncionesAgilent33220AUIFigure)

if nargin == 0
    clear app
end
end

% Code that executes before app deletion
function delete(app)

    % Delete UIFigure when app is deleted
    delete(app.ControlRemotoGeneradordeFuncionesAgilent33220AUIFigure)
end
end
end
```

8.3 Procesado de medidas experimentales con App Designer.

```

classdef procesarmedidas < matlab.apps.AppBase

    % Properties that correspond to app components
    properties (Access = public)
        AplicacionparaprocesarmedidasUIFigure matlab.ui.Figure
        PlanoMedidomPanel matlab.ui.container.Panel
        Xinicio matlab.ui.control.NumericEditField
        Xpaso matlab.ui.control.NumericEditField
        Xfin matlab.ui.control.NumericEditField
        Yinicio matlab.ui.control.NumericEditField
        Ypaso matlab.ui.control.NumericEditField
        Yfin matlab.ui.control.NumericEditField
        EjeXLabel matlab.ui.control.Label
        InicioLabel matlab.ui.control.Label
        PasoLabel matlab.ui.control.Label
        FinLabel matlab.ui.control.Label
        InicioLabel_2 matlab.ui.control.Label
        PasoLabel_2 matlab.ui.control.Label
        FinLabel_2 matlab.ui.control.Label
        EjeYLabel matlab.ui.control.Label
        BarridodeFrecuenciaskHzPanel matlab.ui.container.Panel
        FrecInicio matlab.ui.control.NumericEditField
        FrecPaso matlab.ui.control.NumericEditField
        FrecFin matlab.ui.control.NumericEditField
        EjeXLabel_2 matlab.ui.control.Label
        InicioLabel_3 matlab.ui.control.Label
        PasoLabel_3 matlab.ui.control.Label
        FinLabel_3 matlab.ui.control.Label
        ProcesarmedidasButton matlab.ui.control.Button
        SeleccionarButton matlab.ui.control.Button
        DirectoriodelplanomedidoconlenteLabel matlab.ui.control.Label
        Directoriotexto matlab.ui.control.EditField
        PlanosdepresinSwitchLabel matlab.ui.control.Label
        PlanospresionSwitch matlab.ui.control.Switch
        NormalizarplanosdepresinSwitchLabel matlab.ui.control.Label
        normalizarSwitch matlab.ui.control.Switch
        CorteslongitudinalesSwitchLabel matlab.ui.control.Label
        CorteslongitudinalesSwitch matlab.ui.control.Switch
        CortestransversalesSwitchLabel matlab.ui.control.Label
        CortestransversalesSwitch matlab.ui.control.Switch
        DistanciareceptorlentePanel matlab.ui.container.Panel
        dRecLent matlab.ui.control.NumericEditField
        GananciaenplanosycortesdBSwitchLabel matlab.ui.control.Label
        GananciaenplanosycortesdBSwitch matlab.ui.control.Switch
        Directoriotexto_2 matlab.ui.control.EditField
        DirectoriodelplanomedidosinlenteLabel matlab.ui.control.Label
        SeleccionarButton_2 matlab.ui.control.Button
    end

    methods (Access = private)

        % Button pushed function: SeleccionarButton
        function SeleccionarButtonPushed(app, event)
            %Cargar directorio de las medidas con lente
            folder=uigetdir();
            app.Directoriotexto.Value=folder;
        end

        % Button pushed function: ProcesarmedidasButton
        function ProcesarmedidasButtonPushed(app, event)
            fichero=[app.Directoriotexto.Value, '/matfile/punto'];

            ficherosinLente=[app.Directoriotexto_2.Value, '/matfile/punto'];

            %Frecuencias a barrer
            frec_inicio=app.FrecInicio.Value;
            frec_paso=app.FrecPaso.Value;
            frec_fin=app.FrecFin.Value;

            frec=frec_inicio:frec_paso:frec_fin;

            %Ejes del plano medido
            ejex_inicio=app.Xinicio.Value;
            ejex_paso=app.Xpaso.Value;
            ejex_fin=app.Xfin.Value;

            ejey_inicio=app.Yinicio.Value;
            ejey_paso=app.Ypaso.Value;
            ejey_fin=app.Yfin.Value;

            ejeX=ejex_inicio:ejex_paso:ejex_fin;
            ejeY=ejey_inicio:ejey_paso:ejey_fin;
        end
    end
end

```



```
distanciaRecLente=app.dRecLent.Value;

%Inicializamos los planos
plano=zeros(length(ejeX),length(ejeY));
planosinLente=zeros(length(ejeX),length(ejeY));
planoGanancia=zeros(length(ejeX),length(ejeY));

%Inicializamos los planos frecuenciales
for n=1:length(frec)
    eval(['TF',num2str(frec(n)),'=plano;']);
end

%Creamos la barra de carga
h=waitbar(0,'Procesando medidas...');
puntos_totales=length(ejeX)*length(ejeY);

cuentaPunto=1; %Punto actual
for puntoX=1:length(ejeX)

    %Actualizamos la barra de carga
    waitbar(cuentaPunto/puntos_totales);

    if mod(ejeX(puntoX),2)==0
        for puntoY=1:length(ejeY)
            load([fichero,num2str(cuentaPunto)]);
            tmp=max(abs(dato.senyal)); %En los puntos del plano de la señal temporal sin fft se toma el valor maximo

            N=10000; %Puntos fft
            fs=1/dato.dt; %Frec. de muestreo
            X=abs(fft(dato.senyal,N)); %FFT

            %Guardamos las componentes frecuenciales en los planos
            for n=1:length(frec)
                indice=(frec(n)*1e3/fs)*N;
                eval(['TF',num2str(frec(n)),'(puntoX,puntoY)=X(',num2str(indice+1),')']);
            end

            plano(puntoX,puntoY)=tmp;
            cuentaPunto=cuentaPunto+1;
        end
    else
        for puntoY=length(ejeY):-1:1
            load([fichero,num2str(cuentaPunto)]);
            tmp=max(abs(dato.senyal));

            N=10000; %Puntos de la fft
            fs=1/dato.dt; %Frec. de muestreo
            X=abs(fft(dato.senyal,N)); %FFT

            %Guardamos las componentes frecuenciales en los planos
            for n=1:length(frec)
                indice=(frec(n)*1e3/fs)*N;
                eval(['TF',num2str(frec(n)),'(puntoX,puntoY)=X(',num2str(indice+1),')']);
            end

            %Guardamos el plano de la señal temporal sin fft
            plano(puntoX,puntoY)=tmp;
            cuentaPunto=cuentaPunto+1;
        end
    end
end

%Repetimos lo anterior con el plano sin lente

cuentaPunto=1; %Punto actual
for puntoX=1:length(ejeX)

    %Actualizamos la barra de carga
    waitbar(cuentaPunto/puntos_totales);

    if mod(ejeX(puntoX),2)==0
        for puntoY=1:length(ejeY)
            load([ficherosinLente,num2str(cuentaPunto)]);
            tmp=max(abs(dato.senyal));

            planosinLente(puntoX,puntoY)=tmp;
            cuentaPunto=cuentaPunto+1;
        end
    else
        for puntoY=length(ejeY):-1:1
            load([ficherosinLente,num2str(cuentaPunto)]);
            tmp=max(abs(dato.senyal));

            planosinLente(puntoX,puntoY)=tmp;
            cuentaPunto=cuentaPunto+1;
        end
    end
end
end
```



```
%Calcular plano de ganancia a partir del plano de la señal temporal con y sin lente
if (isequal(app.GananciaenplanosycortesdBSwitch.Value,'0n'))
    planoGanancia=20*log10(plano)-20*log10(planosinLente);

end

waitbar(1,h,'Guardando resultados...');

% Creamos los ejes
dTot=abs(max(ejeX)-min(ejeX))*1e-3;
rTot=abs(max(ejeY)-min(ejeY))*1e-3;

EjeZ=linspace(distanciaRecLente,distanciaRecLente + dTot,length(ejeX));
EjeR=linspace(-rTot/2,rTot/2,length(ejeY));

% Guardar los planos en PNG (misma escala en todos)
%Encontramos el máximo entre todos los planos de presión para normalizar
max_TF=0; %Valor máximo
for n=1:length(frec)
    eval(['max_tmp=max(max(TF',num2str(frec(n)),')');]);
    if max_tmp>max_TF
        max_TF=max_tmp;
    end
end

if (isequal(app.normalizarSwitch.Value,'0n'))
    %Normalizamos los planos respecto al máximo
    for n=1:length(frec)
        eval(['TF',num2str(frec(n)),'=TF',num2str(frec(n)), './max_TF;']);
    end
end

%Dibujamos planos frecuenciales
if (isequal(app.PlanospresionSwitch.Value,'0n'))
    mkdir([app.Directoriotexto.Value,'/planosBarrido']);
    for n=1:length(frec)
        figure;
        eval(['surf(EjeR,EjeZ,TF',num2str(frec(n)),')');]);
        grid minor;
        axis tight;
        colormap jet;
        shading interp;
        colorbar;

        %Normalizado
        if (isequal(app.normalizarSwitch.Value,'0n'))
            caxis([0 1]);
        end

        view(0,90);
        xlabel('r (m)');
        ylabel('z (m)');
        title([num2str(frec(n)),' kHz']);
        saveas(gcf,[app.Directoriotexto.Value,'/planosBarrido/plano',num2str(frec(n)),'khz.png']);

        close;
    end
    %Planos de la señal temporal sin fft con y sin lente
    figure;
    surf(EjeR,EjeZ,plano);
    grid minor;
    axis tight;
    colormap jet;
    shading interp;
    colorbar;
    view(0,90);
    xlabel('r (m)');
    ylabel('z (m)');
    title('Temporal (Pa)');
    saveas(gcf,[app.Directoriotexto.Value,'/planosBarrido/temporal.png']);
    saveas(gcf,[app.Directoriotexto.Value,'/planosBarrido/temporal.fig']);
    close;
```



```
figure;
surf(EjeR,EjeZ,planosinLente);
grid minor;
axis tight;
colormap jet;
shading interp;
colorbar;
view(0,90);
xlabel('r (m)');
ylabel('z (m)');
title('Temporal sin Lente (Pa)');
saveas(gcf,[app.Directoriotexto.Value,'/planosBarrido/temporal sin Lente.png']);
saveas(gcf,[app.Directoriotexto.Value,'/planosBarrido/temporal sin Lente.fig']);
close;

%Dibujar plano de ganancia
if (isequal(app.GananciaenplanosydbSwitch.Value,'On'))
    figure;
    surf(EjeR,EjeZ,planoGanancia);
    grid minor;
    axis tight;
    colormap jet;
    shading interp;
    colorbar;
    view(0,90);
    xlabel('r (m)');
    ylabel('z (m)');
    title('Plano Ganancia Temporal(dB)');
    saveas(gcf,[app.Directoriotexto.Value,'/planosBarrido/Ganancia temporal dB.png']);
    saveas(gcf,[app.Directoriotexto.Value,'/planosBarrido/Ganancia temporal dB.fig']);

    close;
end
end

% Ejes transversal y longitudinal
if (isequal(app.transversalesSwitch.Value,'On'))
    %Sacamos los transversales a cada frecuencia
    mkdir([app.Directoriotexto.Value,'/Transversales']);
    for n=1:length(frec)
        eval(['indiceX,indiceY=find(TF',num2str(frec(n)),',==max(max(TF',num2str(frec(n)),'));');]);

        figure;
        eval(['plot(EjeR,TF',num2str(frec(n)),',(indiceX,:));']);
        axis tight;
        xlabel('r (m)');
        ylabel('|p|/|p|_m_a_x');
        grid minor;

        if (isequal(app.normalizarSwitch.Value,'On'))
            ylim([0 1]);
        end

        title(['Corte transversal a ',num2str(frec(n)), ' kHz en z=',num2str(EjeZ(indiceX)), ' m']);

        saveas(gcf,[app.Directoriotexto.Value,'/cortesTransversales/transversal',num2str(frec(n)), 'khz.png']);

        close;
    end
end

%Corte transversal de la señal temporal sin fft
figure;
eval(['M,I = max(plano(:));']);
eval(['I_row, I_col = ind2sub(size(plano),I);']);
eval(['plot(EjeR,plano(I_row,:));']);

axis tight;
xlabel('z (m)');
ylabel('p (pa)');
grid minor;

title(['Corte transversal temporal (Pa)']);

saveas(gcf,[app.Directoriotexto.Value,'/cortesTransversales/transversal temporal.png']);
saveas(gcf,[app.Directoriotexto.Value,'/cortesTransversales/transversal temporal.fig']);

close;

%Transversal en dB
if (isequal(app.GananciaenplanosycortesdbSwitch.Value,'On'))

    figure;
    eval(['plot(EjeR,planoGanancia(I_row,:));']);

    axis tight;
    xlabel('z (m)');
    ylabel('p (dB)');
    grid minor;

    title(['Ganancia en corte transversal (dB)']);

    saveas(gcf,[app.Directoriotexto.Value,'/cortesTransversales/ganancia en corte transversal.png']);
    saveas(gcf,[app.Directoriotexto.Value,'/cortesTransversales/ganancia en corte transversal.fig']);
    close;

end
end
```



```
if (isequal(app.CortesLongitudinalesSwitch.Value, 'On'))
    %Sacamos los cortes longitudinales de cada frecuencia
    mkdir([app.Directoriotexto.Value, '/cortesLongitudinales']);
    for n=1:length(frec)
        eval(['[indiceX, indiceY]=find(TF', num2str(frec(n)), '==max(max(TF', num2str(frec(n)), '));');]);

        figure;
        eval(['plot(EjeZ, TF', num2str(frec(n)), '(:, indiceY));']);
        axis tight;
        xlabel('z (m)');
        ylabel('|p|/|p|_m_a_x');
        grid minor;

        if (isequal(app.normalizarSwitch.Value, 'On'))
            ylim([0 1]);
        end

        title(['Corte longitudinal a ', num2str(frec(n)), ' KHz en r=', num2str(EjeR(indiceY)), ' m']);

        saveas(gcf, [app.Directoriotexto.Value, '/cortesLongitudinales/longitudinal', num2str(frec(n)), 'khz.png']);
    end
end

%Corte longitudinal de la señal temporal sin fft
figure;
eval(['A, indiceY]=max(max(plano));']);
eval(['plot(EjeZ, plano(:, indiceY));']);
axis tight;
xlabel('r (m)');
ylabel('p (Pa)');
grid minor;

title(['Corte longitudinal temporal (Pa)']);

saveas(gcf, [app.Directoriotexto.Value, '/cortesLongitudinales/longitudinal temporal.png']);
saveas(gcf, [app.Directoriotexto.Value, '/cortesLongitudinales/longitudinal temporal.fig']);

close;
%Longitudinal en dB
if (isequal(app.GananciaenplanosycortesdBSwitch.Value, 'On'))
    figure;

    eval(['plot(EjeZ, planoGanancia(:, indiceY));']);
    axis tight;
    xlabel('r (m)');
    ylabel('p (dB)');
    grid minor;
    title(['Ganancia en corte longitudinal (dB)']);

    saveas(gcf, [app.Directoriotexto.Value, '/cortesLongitudinales/ganancia corte longitudinal.png']);
    saveas(gcf, [app.Directoriotexto.Value, '/cortesLongitudinales/ganancia corte longitudinal.fig']);

    close;

end

end

%Cerramos la barra de carga
close(h);

end
```

end

```
% Button pushed function: SeleccionarButton_2
function SeleccionarButton_2Pushed(app, event)
    %Cargar directorio de las medidas sin lente
    folder=uigetdir();
    app.Directoriotexto_2.Value=folder;
end
```

end

```
% App initialization and construction
methods (Access = private)
```

```
% Create UIFigure and components
function createComponents(app)
```

```
% Create AplicacinparaprocesarmedidasUIFigure
app.AplicacinparaprocesarmedidasUIFigure = uifigure;
app.AplicacinparaprocesarmedidasUIFigure.Color = [0.302 0.749 0.9294];
app.AplicacinparaprocesarmedidasUIFigure.Position = [100 100 662 561];
app.AplicacinparaprocesarmedidasUIFigure.Name = 'Aplicación para procesar medidas';
app.AplicacinparaprocesarmedidasUIFigure.Resize = 'off';

% Create PlanoMedidommPanel
app.PlanoMedidommPanel = uipanel(app.AplicacinparaprocesarmedidasUIFigure);
app.PlanoMedidommPanel.Title = 'Plano Medido (mm)';
app.PlanoMedidommPanel.Position = [26 237 363 203];

% Create Xinicio
app.Xinicio = uieditfield(app.PlanoMedidommPanel, 'numeric');
app.Xinicio.Position = [18 115 100 22];
```



```
% Create Xpaso
app.Xpaso = uieditfield(app.PlanoMedidommPanel, 'numeric');
app.Xpaso.Position = [130 115 100 22];

% Create Xfin
app.Xfin = uieditfield(app.PlanoMedidommPanel, 'numeric');
app.Xfin.Position = [241 115 100 22];

% Create Yinicio
app.Yinicio = uieditfield(app.PlanoMedidommPanel, 'numeric');
app.Yinicio.Position = [18 40 100 22];

% Create Ypaso
app.Ypaso = uieditfield(app.PlanoMedidommPanel, 'numeric');
app.Ypaso.Position = [130 40 100 22];

% Create Yfin
app.Yfin = uieditfield(app.PlanoMedidommPanel, 'numeric');
app.Yfin.Position = [241 40 100 22];

% Create EjeXLabel
app.EjeXLabel = uilabel(app.PlanoMedidommPanel);
app.EjeXLabel.Position = [18 160 33 15];
app.EjeXLabel.Text = 'Eje X';

% Create InicioLabel
app.InicioLabel = uilabel(app.PlanoMedidommPanel);
app.InicioLabel.Position = [51 146 34 15];
app.InicioLabel.Text = 'Inicio';

% Create PasoLabel
app.PasoLabel = uilabel(app.PlanoMedidommPanel);
app.PasoLabel.Position = [165 146 33 15];
app.PasoLabel.Text = 'Paso';

% Create FinLabel
app.FinLabel = uilabel(app.PlanoMedidommPanel);
app.FinLabel.Position = [274 146 25 15];
app.FinLabel.Text = 'Fin';

% Create InicioLabel_2
app.InicioLabel_2 = uilabel(app.PlanoMedidommPanel);
app.InicioLabel_2.Position = [51 69 34 15];
app.InicioLabel_2.Text = 'Inicio';

% Create PasoLabel_2
app.PasoLabel_2 = uilabel(app.PlanoMedidommPanel);
app.PasoLabel_2.Position = [165 69 33 15];
app.PasoLabel_2.Text = 'Paso';

% Create FinLabel_2
app.FinLabel_2 = uilabel(app.PlanoMedidommPanel);
app.FinLabel_2.Position = [274 69 25 15];
app.FinLabel_2.Text = 'Fin';

% Create EjeYLabel
app.EjeYLabel = uilabel(app.PlanoMedidommPanel);
app.EjeYLabel.Position = [18 83 33 15];
app.EjeYLabel.Text = 'Eje Y';

% Create BarridodeFrecuenciaskHzPanel
app.BarridodeFrecuenciaskHzPanel = uipanel(app.AplicacinparaprocesarmedidasUIFigure);
app.BarridodeFrecuenciaskHzPanel.Title = 'Barrido de Frecuencias (kHz)';
app.BarridodeFrecuenciaskHzPanel.Position = [26 115 363 113];

% Create FrecInicio
app.FrecInicio = uieditfield(app.BarridodeFrecuenciaskHzPanel, 'numeric');
app.FrecInicio.Position = [18 25 100 22];

% Create FrecPaso
app.FrecPaso = uieditfield(app.BarridodeFrecuenciaskHzPanel, 'numeric');
app.FrecPaso.Position = [130 25 100 22];

% Create FrecFin
app.FrecFin = uieditfield(app.BarridodeFrecuenciaskHzPanel, 'numeric');
app.FrecFin.Position = [241 25 100 22];

% Create EjeXLabel_2
app.EjeXLabel_2 = uilabel(app.BarridodeFrecuenciaskHzPanel);
app.EjeXLabel_2.Position = [18 70 33 15];
app.EjeXLabel_2.Text = 'Eje X';

% Create InicioLabel_3
app.InicioLabel_3 = uilabel(app.BarridodeFrecuenciaskHzPanel);
app.InicioLabel_3.Position = [51 56 37 15];
app.InicioLabel_3.Text = 'Inicio';

% Create PasoLabel_3
app.PasoLabel_3 = uilabel(app.BarridodeFrecuenciaskHzPanel);
app.PasoLabel_3.Position = [165 56 33 15];
app.PasoLabel_3.Text = 'Paso';

% Create FinLabel_3
app.FinLabel_3 = uilabel(app.BarridodeFrecuenciaskHzPanel);
app.FinLabel_3.Position = [274 56 25 15];
app.FinLabel_3.Text = 'Fin';
```

```
% Create ProcesarmedidasButton
app.ProcesarmedidasButton = uibutton(app.AplicacinparaprocesarmedidasUIFigure, 'push');
app.ProcesarmedidasButton.ButtonPushedFcn = createCallbackFcn(app, @ProcesarmedidasButtonPushed, true);
app.ProcesarmedidasButton.BackgroundColor = [1 0 0];
app.ProcesarmedidasButton.FontColor = [1 1 1];
app.ProcesarmedidasButton.Position = [481 74 113 22];
app.ProcesarmedidasButton.Text = 'Procesar medidas';

% Create SeleccionarButton
app.SeleccionarButton = uibutton(app.AplicacinparaprocesarmedidasUIFigure, 'push');
app.SeleccionarButton.ButtonPushedFcn = createCallbackFcn(app, @SeleccionarButtonPushed, true);
app.SeleccionarButton.Position = [519 506 100 22];
app.SeleccionarButton.Text = 'Seleccionar';

% Create DirectoriodelplanomedidoconlenteLabel
app.DirectoriodelplanomedidoconlenteLabel = uilabel(app.AplicacinparaprocesarmedidasUIFigure);
app.DirectoriodelplanomedidoconlenteLabel.Position = [50 535 208 15];
app.DirectoriodelplanomedidoconlenteLabel.Text = 'Directorio del plano medido con lente';

% Create Directoriotexto
app.Directoriotexto = uieditfield(app.AplicacinparaprocesarmedidasUIFigure, 'text');
app.Directoriotexto.Position = [50 506 433 22];

% Create PlanosdepresinSwitchLabel
app.PlanosdepresinSwitchLabel = uilabel(app.AplicacinparaprocesarmedidasUIFigure);
app.PlanosdepresinSwitchLabel.HorizontalAlignment = 'center';
app.PlanosdepresinSwitchLabel.Position = [486 396 102 15];
app.PlanosdepresinSwitchLabel.Text = 'Planos de presión';

% Create PlanospresionSwitch
app.PlanospresionSwitch = uiswitch(app.AplicacinparaprocesarmedidasUIFigure, 'slider');
app.PlanospresionSwitch.Position = [516 372 45 20];
app.PlanospresionSwitch.Value = 'On';

% Create NormalizarplanosdepresinSwitchLabel
app.NormalizarplanosdepresinSwitchLabel = uilabel(app.AplicacinparaprocesarmedidasUIFigure);
app.NormalizarplanosdepresinSwitchLabel.HorizontalAlignment = 'center';
app.NormalizarplanosdepresinSwitchLabel.Position = [454 343 162 15];
app.NormalizarplanosdepresinSwitchLabel.Text = 'Normalizar planos de presión';

% Create normalizarSwitch
app.normalizarSwitch = uiswitch(app.AplicacinparaprocesarmedidasUIFigure, 'slider');
app.normalizarSwitch.Position = [514 320 45 20];
app.normalizarSwitch.Value = 'On';

% Create CorteslongitudinalesSwitchLabel
app.CorteslongitudinalesSwitchLabel = uilabel(app.AplicacinparaprocesarmedidasUIFigure);
app.CorteslongitudinalesSwitchLabel.HorizontalAlignment = 'center';
app.CorteslongitudinalesSwitchLabel.Position = [478 291 119 15];
app.CorteslongitudinalesSwitchLabel.Text = 'Cortes longitudinales';

% Create CorteslongitudinalesSwitch
app.CorteslongitudinalesSwitch = uiswitch(app.AplicacinparaprocesarmedidasUIFigure, 'slider');
app.CorteslongitudinalesSwitch.Position = [516 268 45 20];
app.CorteslongitudinalesSwitch.Value = 'On';

% Create CortestransversalesSwitchLabel
app.CortestransversalesSwitchLabel = uilabel(app.AplicacinparaprocesarmedidasUIFigure);
app.CortestransversalesSwitchLabel.HorizontalAlignment = 'center';
app.CortestransversalesSwitchLabel.Position = [480 238 115 15];
app.CortestransversalesSwitchLabel.Text = 'Cortes transversales';

% Create CortestransversalesSwitch
app.CortestransversalesSwitch = uiswitch(app.AplicacinparaprocesarmedidasUIFigure, 'slider');
app.CortestransversalesSwitch.Position = [516 217 45 20];
app.CortestransversalesSwitch.Value = 'On';

% Create DistanciareceptorlentePanel
app.DistanciareceptorlentePanel = uipanel(app.AplicacinparaprocesarmedidasUIFigure);
app.DistanciareceptorlentePanel.Title = 'Distancia receptor-lente (m)';
app.DistanciareceptorlentePanel.Position = [26 46 363 58];

% Create dReclent
app.dReclent = uieditfield(app.DistanciareceptorlentePanel, 'numeric');
app.dReclent.Position = [23 8 100 22];

% Create GananciaenplanosycortesdBswitchLabel
app.GananciaenplanosycortesdBswitchLabel = uilabel(app.AplicacinparaprocesarmedidasUIFigure);
app.GananciaenplanosycortesdBswitchLabel.HorizontalAlignment = 'center';
app.GananciaenplanosycortesdBswitchLabel.Position = [448 186 183 15];
app.GananciaenplanosycortesdBswitchLabel.Text = 'Ganancia en planos y cortes (dB)';

% Create GananciaenplanosycortesdBswitch
app.GananciaenplanosycortesdBswitch = uiswitch(app.AplicacinparaprocesarmedidasUIFigure, 'slider');
app.GananciaenplanosycortesdBswitch.Position = [516 161 45 20];
app.GananciaenplanosycortesdBswitch.Value = 'On';

% Create Directoriotexto_2
app.Directoriotexto_2 = uieditfield(app.AplicacinparaprocesarmedidasUIFigure, 'text');
app.Directoriotexto_2.Position = [50 453 433 22];
```



```
% Create DirectoriodelplanomedidosinlenteLabel
app.DirectoriodelplanomedidosinlenteLabel = uilabel(app.AplicacinparaprocesarmedidasUIFigure);
app.DirectoriodelplanomedidosinlenteLabel.Position = [50 482 203 15];
app.DirectoriodelplanomedidosinlenteLabel.Text = 'Directorio del plano medido sin lente';

% Create SeleccionarButton_2
app.SeleccionarButton_2 = uibutton(app.AplicacinparaprocesarmedidasUIFigure, 'push');
app.SeleccionarButton_2.ButtonPushedFcn = createCallbackFcn(app, @SeleccionarButton_2Pushed, true);
app.SeleccionarButton_2.Position = [519 453 100 22];
app.SeleccionarButton_2.Text = 'Seleccionar';
end
end

methods (Access = public)

% Construct app
function app = procesarmedidas

% Create and configure components
createComponents(app)

% Register the app with App Designer
registerApp(app, app.AplicacinparaprocesarmedidasUIFigure)

if nargin == 0
clear app
end
end

% Code that executes before app deletion
function delete(app)

% Delete UIFigure when app is deleted
delete(app.AplicacinparaprocesarmedidasUIFigure)
end
end
end
```