



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE ESTUDIOS SUPERIORES ARAGÓN

**PROCESAMIENTO DE SEÑALES PARA
UNA SONDA DE ULTRASONIDO
DOPPLER TRANSCRANEAL**

TESIS

Que para obtener el título de

**Ingeniero Eléctrico-
Electrónico**

P R E S E N T A

Sergio Zaleta Andrade

DIRECTOR DE TESIS

Fernando Xavier Vázquez Martínez





Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

A la Facultad de Estudios Superiores Aragón y a la Universidad, por la formación que
me han dado.

Reconocimientos

A mi madre, mi hermano y, en especial, a mi padre por haberme apoyado de manera incondicional, en todos los aspectos, durante mi carrera en la Universidad, incluso en situaciones adversas. Sin ellos, nada de esto hubiese sido posible. Gracias.

Al Dr. Andrea Lecchini Visintini por haberme aceptado como asistente de investigación visitante en la Universidad de Leicester. A la Dra. Emma Chung por haberme dado la oportunidad de colaborar en su proyecto de investigación y por su paciencia y disponibilidad para enseñarme los fundamentos del ultrasonido Doppler TCD. A la estudiante de doctorado y amiga Poppy Turner por haber supervisado mi trabajo durante mi estancia en la Universidad de Leicester, así como de manera remota y siempre resolver mis dudas y estar al pendiente del desarrollo de este trabajo de tesis.

Al Dr. Donato Valdez Pérez por sus consejos y ayudarme a sentirme con confianza para seguir adelante en el desarrollo de este trabajo de tesis y mis estudios académicos, así como su incondicional apoyo para resolver mis dudas al escribir algunos capítulos de este proyecto.

A mi amigo y hermano Luis Sastre, y a la Universidad Politécnica de Valencia – su Universidad - por haber colaborado con la mayoría de los documentos científicos en los que este trabajo de tesis se sustenta.

Al Mtro. Fernando Xavier Vázquez-Martínez por ser mi asesor de tesis y por haber sido un apoyo durante el último año de mi carrera.

A mis amigos por haber estado conmigo siempre, motivándome, y recibir mis llamadas siempre que me sentía con fatiga por haber estado escribiendo durante varias horas capítulos de esta tesis.

Declaración de autenticidad

Por la presente declaro que el contenido de este trabajo de tesis es original y no se ha presentado de forma total o parcial para su consideración en cualquier otro título o grado en esta o cualquier otra Universidad del mundo, a menos que se haga referencia específica al trabajo de otras personas. Este trabajo de tesis es el resultado de mi investigación y no incluye nada que sea el resultado de algún trabajo realizado en colaboración, salvo que se especifique lo contrario en el texto.

Sergio Zaleta Andrade. Estado de México, 2020.

Índice general

Índice de figuras	VIII
Índice de tablas	XII
1. Resumen	2
2. Tecnología de los Sistemas de Ultrasonido Doppler Transcraneal	4
2.1 Transductores Ultrasónicos	5
La Idoneidad y Desempeño de los Materiales en los Transductores	5
Estructura de los Transductores Básicos	7
2.2 Rayos y Campos Ultrasónicos	9
Generación de Rayos y Campos Ultrasónicos	9
Rayos y Campos Ultrasónicos de Onda Continua – CW	9
Rayos y Campos Ultrasónicos de Onda Pulsada – PW	11
2.3 Reconstrucción de la Sección Transversal de un Rayo Ultrasónico Obtenido Experimentalmente Utilizando Matlab®	13
Equipo de Laboratorio a Utilizar: Descripción general	13
Proceso de Captura de Información: Ejecución del Experimento	15
Estructura de Datos de Archivos del Experimento: a Detalle	16
2.4 Estructura del Código de Reconstrucción de la Sección Transversal de un Rayo Ultrasónico Obtenido Experimentalmente Utilizando Matlab®	18
Lectura de Archivos *.tdt y *.txt con Matlab®	18
Gráfica de la Sección Transversal del Haz Ultrasónico en Dos Dimensiones: Última Parte del Proceso de Refactorización de Datos	22
Interfaz Gráfica de Usuario (GUI) del Código de reconstrucción de la Sección Transversal de un Rayo Ultrasónico Obtenido Experimentalmente Utilizando Matlab®	23
2.5 Frecuencia Doppler en Sistemas CW	25
2.6 Sistemas Doppler de Onda Continua – CW	26
2.7 Sistemas Doppler de Onda Pulsada Multibarrera – PW	28
Tipos de Sistemas Doppler PW	28
Procesamiento de Señales en Serie en Sistemas Doppler PW Multibarrera	30
Proceso de Obtención de Información Doppler en sistemas Doppler PW de una Barrera	31
La Demodulación Analógica de la Frecuencia Doppler de un sistema PW de una Barrera	32
Asegurar el Funcionamiento Adecuado del Detector del Cruce por Cero para Evitar la Atenuación de las Señales	34
Consideraciones a la Salida del Detector de Fase para Asegurar el Funcionamiento Adecuado del Sistema PW	35

Descripción Técnica de un sistema Doppler PW Multibarrera con Procesamiento de Datos Seriales	35
3. Intensidad del Eco de una Sonda de Ultrasonido Doppler Transcraneal	37
3.1 Dispositivos de Ultrasonido Transcraneales No Invasivos	37
Sistema Spencer	37
Posición de la Sonda en el Sistema Spencer	38
Sistema Nihon Kohden	39
3.2 Interfaz Gráfica de Usuario (GUI) en Matlab® para el Cálculo de la Intensidad de la Sonda de Ultrasonido Transcraneal de Equipos de Tecnologías Spencer y Nihon Kohden	41
Propósito de la Interfaz Gráfica de Usuario	41
Estructura del Código de la GUI	42
Descripción de los Códigos Incorporados en la GUI	43
Descripción de la GUI	44
3.3 Lectura de Archivos de las Señales de Ultrasonido Doppler de los Equipos de Tecnologías Spencer y Nihon Kohden	45
Archivos .mat del Sistema Spencer	46
Archivos .iq del Sistema NK	46
Algoritmo de Extracción de Datos de Archivos del Sistema Spencer y NK en la GUI	48
3.4 Aproximación Inicial: Señal de Ultrasonido Doppler Transcraneal Generadora del Eco	49
Parámetros del Sistema Spencer y NK	49
Algoritmo de Aproximación de la Señal de Ultrasonido Doppler Transcraneal Generadora del Eco en la GUI	54
3.5 Compresión de las Señales de Ultrasonido Doppler Transcraneales	57
Algoritmo de Compresión de Señales de Ultrasonido Doppler Usando la GUI	59
4. Temas Selectos de Procesamiento de Señales de Ultrasonido Doppler Transcraneal	60
4.1 Sobremuestreo de Alta Definición	60
4.2 Tecnología ST3 Doppler Transcraneal de Modo M de Potencia (PMM)	63
4.3 Interfaz Gráfica de Usuario para Analizar Señales de Ultrasonido Doppler Transcraneales	64
5. Conclusiones y Trabajo Futuro	66
5.1 Conclusiones	66
5.2 Discusión	67
Aproximación Inicial: Señal de Ultrasonido Doppler Transcraneal Generadora del Eco	67
Reducción de la Tasa de Muestreo o Compresión de las Señales	68
Control de Potencia de Transmisión en el Programa de Adquisición de Datos del Sistema NK	69
Exportación de Datos entre Aplicaciones de Equipos de Ultrasonido TCD	69
5.3 Trabajo Futuro	70
Investigación Futura	70
Algoritmos	70

Apéndice A – Acrónimos	71
Apéndice B – Códigos del Cálculo de la Intensidad de Señales de Ultrasonido Doppler Transcraneales	73
Interfaz Gráfica de Usuario – gui_Echo.m	73
Lectura de Archivos de equipos Spencer y NK – readfile.m	79
Generación de las Amplitudes de las Señales de Ultrasonido Doppler – beam.m	83
Compresión de las Señales TCD – avData.m	83
Normalización a Uno de las Amplitudes de las Señales Comprimidas – norm2one.m	84
Método de Aproximación de la Intensidad Inicial de las Señales de Ultrasonido TCD – amplitude.m	84
Cálculo de la Amplitud máxima Entre Todas las Señales de un Archivo de los Equipos Spencer o NK – peaksfinder.m	85
Apéndice C – Códigos para la Reconstrucción de la Sección Transversal de un Rayo Ultrasónico Obtenido Experimentalmente Utilizando Matlab®	88
Interfaz Gráfica para Reconstruir la Sección Transversal de un Rayo Ultrasónico – beampattern_GUI.m	88
Lectura de Archivos *.tdt y *.txt – filereader.m	92
Código Incorporado en filereader.m para poder llevar a cabo la lectura de archivos *.tdt – tdtread.m	92
Código Incorporado en filereader.m para poder llevar a cabo la lectura de archivos *.txt – txtread.m	93
Extracción de los Valores RMS de los Encabezados de los 37 Archivos que Contienen la Información de la Sección Transversal del Haz del Ultrasonido – main2.m	94

Índice de figuras

- 2.1. Transductor amortiguado de un solo elemento para la generación y detección de ultrasonidos de onda pulsada. Figura obtenida de Evans *et al.* 2000. Adaptada y traducida por Zaleta-Andrade para este trabajo de tesis. 7
- 2.2. Fluctuación a lo largo del eje de un campo ultrasónico de onda continua. Figura obtenida de Evans *et al.* 2000. Adaptada y traducida por Zaleta- Andrade para este trabajo de tesis. 10
- 2.3. Variaciones de la intensidad de un campo de ondas continuas perpendicular al eje en los puntos A y B, representado por la escala de grises. Figura obtenida de Evans *et al.* 2000. Adaptada y traducida por Zaleta-Andrade para este trabajo de tesis. 11
- 2.4. Variación de la intensidad de un campo de ondas pulsadas a lo largo de su eje, correspondiente a tres ciclos. Figura obtenida de Evans *et al.* 2000. Adaptada y traducida por Zaleta-Andrade para este trabajo de tesis. 12
- 2.5. Foto de una configuración similar al experimento de Justyna. Figura obtenida de University of Erlangen (2000). Adaptada por Zaleta-Andrade para este trabajo de tesis. 14
- 2.6. Diagrama de una configuración similar al experimento de Justyna. Figura obtenida de University of Erlangen (2000). Adaptada y traducida por Zaleta- Andrade para este trabajo de tesis. 14
- 2.7. Parte de un archivo *.tdt con nombre “RmsZ-20-0mm.tdt” con datos obtenidos del experimento de Justyna. 17
- 2.8. Parte de un archivo *.tdt con nombre “PmZ-23-0mm.tdt” con datos obtenidos del experimento de Justyna. 17
- 2.9. Parte de un archivo *.txt con nombre “RF-Z-44-0mm-Y8-X33.txt” con datos obtenidos del experimento de Justyna. 18
- 2.10. Interfaz gráfica de usuario (GUI) en Matlab® para la reconstrucción de la sección transversal de un rayo ultrasónico. 23

2.11. Sección transversal de un rayo ultrasónico en la posición Z-20-0mm. Resultado de los eventos “START” y “Select .mat file”.	25
2.12. Relación de los tiempos de muestreo y las amplitudes de cada muestra del haz de ultrasonido.	25
2.13. Ultrasonido Doppler de onda continua siendo insonado en un vaso sanguíneo. Figura obtenida de Atkinson <i>et al.</i> 1977. Adaptada y traducida por Zaleta-Andrade para este trabajo de tesis.	26
2.14. Diagrama a bloques de un sistema típico Doppler CW. Figura obtenida de Evans <i>et al.</i> 2000. Adaptada y traducida por Zaleta-Andrade para este trabajo de tesis.	28
2.15. Diagrama a bloques de un medidor de flujo de un sistema Doppler CW. Figura obtenida de Brody <i>et al.</i> 1974. Adaptada y traducida por Zaleta-Andrade para este trabajo de tesis.	28
2.16. Formas de onda de las velocidades instantáneas registradas simultáneamente en varios lugares del camino del ultrasonido en una arteria carótida común de un sujeto sano. Figura obtenida de Reneman <i>et al.</i> 1986. Adaptada y traducida por Zaleta-Andrade para este trabajo de tesis.	29
2.17. Los perfiles de velocidad axial en intervalos de tiempo discretos durante el ciclo cardiaco. Figura obtenida de Reneman <i>et al.</i> 1986. Adaptada y traducida por Zaleta-Andrade para este trabajo de tesis.	30
2.18. Representación esquemática del procesamiento de datos en paralelo y serie en dispositivos Doppler multibarrera. Los números representan los canales procesados (o barreras). ep es pulso de emisión (emission pulse en idioma inglés). Figura obtenida de Reneman <i>et al.</i> 1986. Adaptada y traducida por Zaleta-Andrade para este trabajo de tesis.	31
2.19. Diagrama a bloques de un sistema Doppler PW de una sola barrera. Las señales Ø1 y Ø2 representan entradas de referencia en cuadratura del detector de fase (que en realidad contiene dos detectores de fase). Figura obtenida de Hoeks, Reneman, y Peronneau 1981. Adaptada y traducida por Zaleta-Andrade para este trabajo de tesis.	33
2.20. Diagrama a bloques de un sistema Doppler PW multibarrera con procesamiento de datos en serie. Figura obtenida de Hoeks <i>et al.</i> 1981. Adaptada y traducida por Zaleta-Andrade para este trabajo de tesis.	36

2.21. Diagrama a bloques de un digitalizador o ADC. Figura obtenida de Hoeks, Reneman, y Peronneau 1981. Adaptada y traducida por Zaleta-Andrade para este trabajo de tesis.	36
3.1. ST3 Sistema de Ultrasonido Transcraneal de Tecnologías Spencer. Figura obtenida de Spencer Technologies 2016.	38
3.2. Hueso temporal.	39
3.3. Línea temporal (línea punteada); tragus (círculo). Figura obtenida de Wikimedia Commons Contributors 2010. Adaptada por Zaleta-Andrade para este trabajo de tesis.	39
3.4. Interfaz de adquisición de datos del dispositivo BrainTV. Figura obtenida de Chung, Banahan, Minhas, y Lecchini-Visintini 2018. Adaptada y traducida por Zaleta-Andrade para este trabajo de tesis.	40
3.5. Equipo de integración (superior izquierda), posición de la sonda (superior derecha), y puertos (inferior derecha) del equipo BrainTV. Figura obtenida de Chung <i>et al.</i> 2018. Adaptada por Zaleta-Andrade para este trabajo de tesis.	41
3.6. Salida de la GUI: intensidad de los valores RMS de las señales de cada una de las barreras (gates) en dB, utilizando un archivo *.iq2 (dos canales).	42
3.7. Flujo de las salidas de cada función incorporada en el código de la GUI.	42
3.8. Parte del código de lectura de archivos *.iq1, *.iq2, y *.mat llamado readfile.m. Figura obtenida del código original gui_Echo.m.	43
3.9. Parte del código que normaliza, a uno, todos los valores devueltos por avData.m: el valor absoluto de las componentes en fase y en cuadratura comprimidas de las señales de ultrasonido Doppler. Figura obtenida del código original gui_Echo.m.	44
3.10. Matlab® GUI que muestra la intensidad de señales de ultrasonido Doppler transcraneal de archivos *.mat, *.iq1, y *.iq2.	45
3.11. Cálculo de la intensidad de las señales de ultrasonido Doppler de 33 profundidades distintas, utilizando la interfaz gráfica de Matlab®.	45
3.12. Estructura interna de un archivo *.mat del sistema Spencer. Figura obtenida de un archivo de la carpeta privada de la investigación de la Universidad de Leicester.	46

3.13.	Encabezado de un archivo *.iq1. Figura obtenida de un archivo contenido en la carpeta privada de la investigación de la Universidad de Leicester.	47
3.14.	Componentes en fase y cuadratura de un archivo *.iq1 de señales de ultrasonido Doppler producidas por la insonación de un rayo ultrasónico transcraneal utilizando el sistema NK. Figura obtenida de un archivo contenido en la carpeta privada de la investigación de la Universidad de Leicester.	48
3.15.	Comandos de obtención de información de guardado del archivo a utilizar para su lectura. Figura obtenida del código original <i>gui_Echo.m</i> .	49
3.16.	Amplitud máxima de la señal de ultrasonido retrodispersada (eco) de la barrera número 10 de un archivo NK de un canal. Las magnitudes del tiempo de muestreo y amplitud del eco son desconocidas para el autor de este trabajo de tesis. Figura obtenida utilizando Matlab®.	50
3.17.	Parte del código de la función <i>peaksfinder.m</i> . Figura obtenida del código original <i>gui_Echo.m</i> .	54
3.18.	Código de la función <i>amplitude.m</i> . Figura obtenida del código original <i>gui_Echo.m</i> .	55
3.19.	Código de la función <i>beam.m</i> . Figura obtenida del código original <i>gui_Echo.m</i> .	55
3.20.	archivo <i>peaks.mat</i> construido con bases de datos de estudios Doppler transcraneales privados de 2019, propiedad de la Universidad de Leicester. Figura obtenida utilizando Matlab®.	56
3.21.	Señal creada aleatoriamente en Matlab® para representar una señal de ultrasonido Doppler con 300 muestras.	57
3.22.	Señal comprimida por un factor de 50 de la señal de ultrasonido Doppler de 300 muestras (creada aleatoriamente con Matlab®)	58
3.23.	Representación simbólica de comprimir una señal por un factor de 50, haciendo referencia al proceso de reducción de la resolución de las señales de ultrasonido Doppler en la función <i>avData.m</i> en la GUI.	58
3.24.	Parte del código de disminución de la resolución de las señales de ultrasonido Doppler transcraneales <i>readfile.m</i> . Figura obtenida del código original <i>gui_Echo.m</i> .	59

4.1. Señales TCD normales en pantalla PMM: arteria cerebral media (MCA) (rojo: 40 – 60 mm), arteria cerebral anterior (ACA) (azul: 60 – 70 mm), ACA contralateral (rojo: 70 – 80 mm), y MCA contralateral (azul: 80 – 90 mm). Figura obtenida de Hakimi <i>et al.</i> (2020).	64
4.2. Interfaz gráfica de usuario BrainTV en Matlab®. Figura obtenida del manual de usuario de la GUI, propiedad privada de la Universidad de Leicester (2019). Adaptada y traducida por Zaleta-Andrade para este trabajo de tesis.	65
5.1. Cálculo de la intensidad de las señales de ultrasonido Doppler de 33 profundidades distintas. A la izquierda se utiliza el valor de referencia inicial con el método de aproximación propuesto en el capítulo 3 de este trabajo de tesis. A la derecha se desprecia la intensidad de referencia (valor casi cero). Figura obtenida de la GUI del cálculo de la intensidad en Matlab®.	67

Índice de tablas

Tabla 2.1. Factores importantes para determinar la idoneidad de algunos materiales piezoeléctricos. Tabla obtenida de Evans <i>et al</i> (2000). Traducida por Zaleta- Andrade para este trabajo de tesis.	5
Tabla 2.2. Propiedades de los materiales piezoeléctricos más comunes. ρ , densidad; c , velocidad del sonido; d , constante de transmisión; g , constante de recepción; k , constante de acoplamiento electro-mecánico; ϵ^T , constante dieléctrica libre. Tabla obtenida de Evans <i>et al</i> (2000).	6
Tabla 2.3. Tabla de las matrices que contienen los datos de la refactorización de los archivos *.tdt o *.txt.	19
Tabla 2.4. Descripción general de las funciones de Matlab® más importantes utilizadas en filereader.m. Información obtenida del centro de ayuda de MathWorks® para América Latina.	19
Tabla 2.5. Descripción general de las funciones de Matlab® más importantes, utilizadas en tdtread.m y txtread.m. Información obtenida del centro de ayuda de MathWorks® para América Latina.	21

Tabla 2.6. Descripción general de las funciones de Matlab® más importantes utilizadas en <i>main2.m</i> . Información obtenida del centro de ayuda de MathWorks® para América Latina.	22
Tabla 2.7. Descripción de la funcionalidad de cada uno de los botones de la GUI para la reconstrucción de la sección transversal de un rayo ultrasónico.	23
Tabla 3.1. Máximas amplitudes de distintos archivos *.iq1 (NK), calculadas utilizando la función <i>peaksfinder.m</i> . Datos obtenidos de archivos contenidos en la carpeta privada de la investigación de la Universidad de Leicester.	51
Tabla 3.2. Máximas amplitudes de distintos archivos *.mat (Spencer), calculadas utilizando la función <i>peaksfinder.m</i> . Datos obtenidos de archivos contenidos en la carpeta privada de la investigación de la Universidad de Leicester.	52

Capítulo 1

Resumen

El estudio del movimiento cerebral en humanos es de particular interés para la medicina de diagnóstico debido a sus alcances en la detección no invasiva del daño cerebral. El uso de equipos de ultrasonido transcraneales Doppler (TCD) permite iterar frecuentemente en el desarrollo de mejores dispositivos que detecten de forma más precisa el daño cerebral de forma no invasiva con datos en tiempo real. Una de las posibles aplicaciones del estudio del movimiento cerebral y del desarrollo de estos dispositivos de ultrasonido TCD se basa en encontrar correlaciones con otros sistemas fisiológicos, como los movimientos del corazón, al igual que para desarrollar equipos de ultrasonido TCD no invasivos de emergencia que puedan detectar daño cerebral, en tiempo real, en una ambulancia. La Universidad de Leicester y el Hospital Universitario de Leicester NHS Trust del Reino Unido han abierto las puertas al autor de este trabajo de tesis para el estudio de las tecnologías de punta de equipos de ultrasonido TCD vanguardistas, como el dispositivo BrainTV desarrollado en colaboración con la compañía japonesa Nihon Kohden, y el programa computacional BrainTV hecho en Matlab® por investigadores del grupo de ingeniería aeroespacial y computacional y el departamento de ciencias cardiovasculares de la Universidad de Leicester.

El capítulo 2 de este trabajo de tesis contiene los fundamentos teóricos necesarios para entender las tecnologías que los sistemas de ultrasonido TCD implementan. El papel fundamental de los osciladores y transductores en la generación de los rayos y campos ultrasónicos de onda continua o pulsante. La forma de generar las mejores posibilidades de detectar el movimiento cerebral y otros parámetros como su velocidad o profundidad. Incluso, se explica a detalle la parte experimental de la caracterización digital de un rayo ultrasónico, hecho por la radióloga Justyna Janus – que a partir de ahora se le hará referencia simplemente como RJ – en colaboración con el autor de este trabajo de tesis.

El capítulo 3 muestra el orden de los sub-capítulos de acuerdo al flujo que el autor de este trabajo de tesis siguió para desarrollar el cálculo en Matlab® de la intensidad del eco de una sonda de ultrasonido Doppler transcraneal desde cero. Para realizar los estudios del movimiento cerebral en el Reino Unido, se tiene que ir al Hospital Universitario de Leicester NHS Trust, ya que es el único lugar en ese país que cuenta con equipos de ultrasonido TCD de última generación. Ahí se encuentran dos equipos TCD de dos compañías diferentes; Tecnologías Spencer y Nihon Kohden. Las tecnologías que estos dispositivos utilizan son de vanguardia, siendo el prototipo TCD de NK el más utilizado por los investigadores de Leicester en la recolección de los datos de campo en pacientes y voluntarios debido a que es un prototipo desarrollado exclusivamente para el estudio del movimiento cerebral en la Universidad de Leicester y el Hospital Universitario de Leicester NHS Trust. Para calcular la intensidad de las señales Doppler, el autor de este trabajo de tesis desarrolló una interfaz de Matlab® que solo necesita un archivo proveniente de un dispositivo NK o Spencer, para mostrar las gráficas de la intensidad deseadas. Los algoritmos de refactorización de datos y del cálculo de las intensidades de las señales Doppler se describen a detalle. También se resuelve el problema de la intensidad inicial mediante un método de aproximación desarrollado por el autor de este trabajo de tesis.

Las tecnologías comerciales de los sistemas TCD NK y Spencer están basadas en conocimiento científico que es accesible en varias universidades del mundo. En el capítulo 4 se describen a detalle los fundamentos de dos tecnologías que hacen posible los estudios del movimiento cerebral. También se hace una revisión general del programa de uso privado BrainTV hecho en Matlab® por la Universidad de Leicester.

Capítulo 2

Tecnología de los Sistemas de Ultrasonido Doppler Transcraneal

Los elementos activos son importantes para la operación de los transductores, en especial su efecto piezoeléctrico. Determinar las características de los elementos activos de los transductores es importante para la medicina de diagnóstico, ya que determinan el desempeño del transductor. Los sistemas TCD utilizan un rango de RF que solo ciertos transductores pueden generar, así que es sumamente importante que el fabricante del oscilador del transductor y del sistema TCD escojan el material piezoeléctrico con las propiedades adecuadas que permitan generar las frecuencias deseadas. El diseño de la estructura de los transductores también es relevante para su desempeño, especialmente en la transmisión de energía acústica a través de su interfaz.

La medición de las profundidades del tejido necesita el uso de ultrasonido de onda pulsada para obtener información Doppler desde la cabeza del transductor hasta el final de un rango fijo o incluso hasta el final de un grupo de varios rangos fijos. Los sistemas TCD utilizan osciladores que generan este tipo de ondas pulsantes en lugar de ondas ultrasónicas continuas.

La correcta calibración de los transductores utilizados para medir las propiedades de los tejidos humanos requiere una descripción completa de las características de los campos ultrasónicos generados. Existen técnicas basadas convencionalmente en mediciones punto a punto del campo acústico que, con ayuda de una computadora, logran predecir el campo ultrasónico a través de cálculos. En este capítulo se muestra el desarrollo experimental y el cálculo de predicción de un campo ultrasónico para representarlo de forma digital utilizando Matlab®, donde este último es uno de los resultados del trabajo del autor de esta tesis.

Los circuitos necesarios para procesar las señales Doppler dependen del tipo de ultrasonido insonado. Una señal de ultrasonido TCD de onda pulsada utiliza un solo transductor para generar y detectar las señales retrodispersadas, en seguida de un

circuito detector de fase, filtros, y un contador de cruce por ceros. El resultado es una señal de audio y una señal de velocidad, conteniendo la distancia de la cabeza del transductor al punto de colisión del ultrasonido con un objetivo. La señal de velocidad también contiene el tiempo de dicha colisión.

2.1 Transductores Ultrasonicos

La idoneidad y desempeño de los materiales en los transductores

En la medicina de diagnóstico, los elementos activos de los transductores dependen de su efecto piezoeléctrico para operar. Es decir, de la habilidad de los elementos activos de generar cargas eléctricas en respuesta a un estrés mecánico aplicado. Actualmente, los materiales de uso más común para dichos fines son los cerámicos piezoeléctricos, pero existen otros como los polímeros piezoeléctricos cuyas propiedades acústicas son muy cercanas a las del tejido. Esto hace que el paso del ultrasonido a lo largo de la interfaz transductor/tejido sea más eficiente.

El desempeño de los transductores no es siempre el mismo, ya que pequeñas variaciones en las características de los materiales o en su estructura pueden causar un gran impacto en el desempeño final. Por ejemplo, los transductores desarrollados exclusivamente para proyectos especiales – como el dispositivo BrainTV mencionado más adelante – pueden tener un rendimiento un poco decepcionante. Esto debido a que estos dispositivos no se han visto beneficiados por las mejoras que los transductores de equipos comerciales pueden adquirir debido al periodo de años de desarrollo a los que se han visto sometidos.

La idoneidad de los materiales depende de un número de factores, en particular, de la presión piezoeléctrica constante y la distorsión del campo eléctrico constante comúnmente denotados por la letra g y d , respectivamente (Shung y Zippuro 1996).

Denotación	Nombre	Definición	Unidades (SI)
------------	--------	------------	------------------

g	Constante de presión	Campo eléctrico de un circuito abierto por unidad de estrés aplicada.	$\frac{\text{volts} \cdot \text{metro}}{\text{Newtons}}$
d	Constante de transmisión	Deformación producida por unidad de estrés aplicado cuando la tensión externa es cero.	$\frac{\text{metro}}{\text{volt}}$

Tabla 2.1: Factores importantes para determinar la idoneidad de algunos materiales piezoeléctricos. Tabla obtenida de Evans *et al* (2000). Traducida por Zaleta-Andrade para este trabajo de tesis.

Evans y McDicken (2000), con permiso de Shung y Zipparo (1966), muestran en la tabla 2.2 las propiedades de los materiales piezoeléctricos más comunes. Además, Evans *et al.* analizan los datos de la tabla, diciendo que los cerámicos PZT son superiores al cuarzo cuando el rendimiento combinado de transmisión/recepción es considerado. También, mencionan que el polímero PVDF tiene un desempeño intermedio y que la elección entre el elemento PZT recae en otras consideraciones prácticas. Por ejemplo, los sistemas de ultrasonido TCD NK y Spencer – mencionados más adelante – no pueden utilizar un material PVDF aunque este muestre una impedancia acústica mejor adaptada al tejido que otros materiales. La razón es porque usar un material PVDF en los sistemas ultrasónicos como Spencer o NK rebasaría los parámetros de frecuencia adecuados para estudios transcraneales. Un material PVDF está diseñado para funcionar a 15 MHz, pero los sistemas NK y Spencer utilizados en ultrasonidos transcraneales solo a 2 MHz.

Propiedad	PVDF	Cuarzo (corte X)	PZT- 5H
d (10^{-12} m/V)	15	2.31	583
g (10^{-2} Vm/N)	14	5.78	1.91

k (sin unidades)	0.11	0.14	0.55
ϵ^T (10 ⁻¹¹ F/m)	9.7	3.98	3010
c (m/s)	2070	5740	3970
ρ (kg/m ³)	1760	2650	7450
Temperatura de Curie (°C)	100	573	190

Tabla 2.2: Propiedades de los materiales piezoeléctricos más comunes. ρ , densidad; c , velocidad del sonido; d , constante de transmisión; g , constante de recepción; k , constante de acoplamiento electro-mecánico; ϵ^T , constante dieléctrica libre. Tabla obtenida de Evans *et al* (2000).

Estructura de los transductores básicos

Es informativo conocer el modelo y estructura básica de un transductor como el de la figura 2.1, aún cuando los equipos de ultrasonido Doppler utilicen transductores más complejos que los que se describen en este sub capítulo.

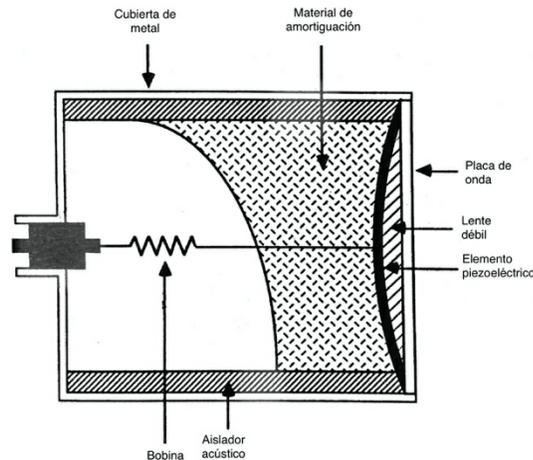


Figura 2.1: Transductor amortiguado de un solo elemento para la generación y detección de ultrasonidos de onda pulsada. Figura obtenida de Evans *et al.* 2000. Adaptada y traducida por Zaleta-Andrade para este trabajo de tesis.

Cada cara del elemento piezoeléctrico es revestido con una capa metálica delgada que actúa como un electrodo. Una conexión por cable a estos electrodos permite que se

aplique una señal de excitación y que la señal del eco pase al amplificador del receptor. En el caso de usar un elemento cerámico, de una o varias capas, este se une a la cara frontal del transductor – llamada placa de onda – con el propósito de reducir la diferencia de la impedancia acústica en la interfaz transductor/tejido. De esta manera lograr una mayor transferencia de energía acústica a través de la interfaz, *i.e.*, reducir las pérdidas de energía del ultrasonido durante el proceso de insonación del ultrasonido en el tejido.

La adaptación de la placa de onda es más efectiva para los transductores CW, ya que el grosor de la placa puede ser igual a un cuarto de la longitud de onda del ultrasonido (Evans *et al.* 2000). Para el caso de los transductores de sistemas PW – que se explican más adelante – el pulso de ultrasonido tiene un determinado ancho de banda en frecuencia y, por lo tanto, una extensión de anchos de banda que la placa de onda no puede acomodar de manera exacta. Sin embargo, Kossoff (1966) y Kino (1987) dicen que el uso de placas de onda continua siguen siendo una característica valiosa de los transductores en sistemas PW (Kossoff 1966). Para reducir las vibraciones del elemento activo después que la excitación del pulso ha cesado, un material de amortiguación – como la resina epoxi cargada de metal – deber ser unido a la superficie superior e inferior del transductor.

El elemento piezoeléctrico y sus componentes están soportados en el extremo de un tubo de aislante acústico que está alojado en un cilindro de metal. Este cilindro de metal se encuentra aterrizado para funcionar como una pantalla que previene la captación de señales electromagnéticas. Esto es muy importante debido a lo débil que pueden ser las señales de eco detectadas por el transductor. El electrodo frontal también está aterrizado para completar la pantalla alrededor del elemento. El electrodo trasero está conectado al transmisor y al receptor del instrumento. En ocasiones se coloca una bobina de inducción a través de la capacitancia del elemento junto a una red de adaptación eléctrica que optimiza la transferencia de potencia durante la transmisión y recepción. Para mantener una buena relación señal a ruido (SNR) es necesario colocar un chip pre-amplificador muy cerca del elemento piezoeléctrico (*ibid*). Finalmente, todo el ensamblado debe ser montado en un estuche de plástico que debe estar sellado para mantener fuera el agua o líquido de acoplamiento.

2.2 Rayos y Campos Ultrasónicos

Generación de rayos y campos ultrasónicos

Para producir campos y rayos ultrasónicos altamente direccionales es necesario que el elemento activo del transductor tenga dimensiones de 1 cm o menos. Esto ya que la longitud de onda del ultrasonido en tejidos blandos para todas las frecuencias utilizadas comercialmente es menor a 1mm (Evans *et al.* 2000). El pequeño tamaño de los transductores es muy conveniente y versátil para el diagnóstico ultrasónico –y su alta capacidad de direccionalidad de suma importancia– ya que prácticamente todas las técnicas de imagen Doppler dependen de ella.

Healey, Leeman, y Weight (1996) hacen hincapié en que la descripción completa de los campos ultrasónicos en las aplicaciones médicas es muy importante cuando se miden las propiedades de ultrasonido en los tejidos humanos. Una descripción completa de los campos ultrasónicos es generalmente requerida en la caracterización y/o calibración de los transductores utilizados en estos estudios (Healey *et al.* 1996). Las técnicas de medición del campo ultrasónico se basan convencionalmente en mediciones punto a punto del campo acústico, utilizando un hidrófono de punto, que son casi invariables en un tanque de agua. Utilizar esta técnica de medición del campo ultrasónico resulta en la obtención de la dependencia tridimensional del espacio y el tiempo del campo de presión en forma de un mapa tetra-dimensional. La cantidad de datos del mapa es tan substancial que Healey *et al.* consideran apropiado especificar el campo con un conjunto de sus datos mucho más pequeños, a partir del cual se pueda predecir el campo ultrasónico a través de un cálculo.

Rayos y campos ultrasónicos de onda continua –CW

Evans *et al.* mencionan que existe una amplia documentación de la transmisión de campos desde elementos de disco plano que oscilan uniformemente a una frecuencia bien definida –como los sistemas CW descritos más adelante– en un modo de vibración delgada. Kinsler, Frey, Coppens, y Sander (1982) han desarrollado soluciones analíticas

completas para la distribución de amplitud de presión a lo largo del eje del transductor y fuera del eje en el campo lejano (Kinsler *et al.* 1962, y Evans *et al.* 2000). La amplitud de presión a lo largo del eje del transductor está dada por:

$$P(z) = P_0 \text{Sin} \left[\frac{\pi}{\lambda} (a^2 + z^2)^{\frac{1}{2}} - z \right] \quad \text{Ec.(2.1)}$$

P_0 : es la amplitud de presión máxima;
 Z : es la distancia a lo largo del eje del transductor;
 a : es el radio del disco.

La figura 2.2 muestra una de las características significativas de las fluctuaciones del transductor al generar un rayo ultrasónico. Se puede observar que existen fluctuaciones rápidas a una distancia menor que a^2/λ con respecto a la cara del transductor: esta región se conoce como campo cercano o campo de Fresnel. El último máximo axial – o el máximo más distante del transductor (Duck, Baker, y Starritt 1998) – ocurre al final del campo cercano: a esta región se le conoce como campo lejano o de Fraunhofer (Evans *et al.* 2000).

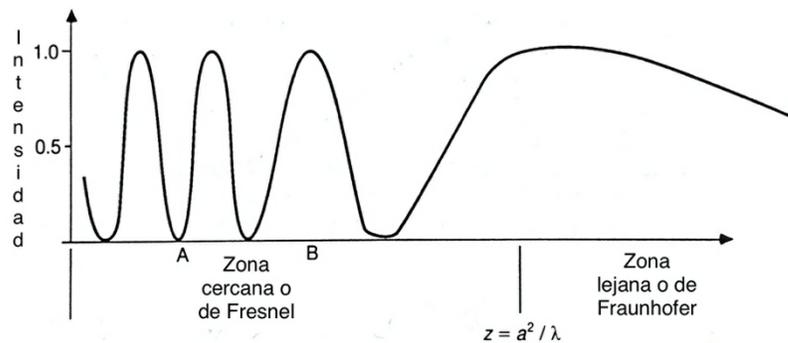


Figura 2.2: Fluctuación a lo largo del eje de un campo ultrasónico de onda continua. Figura obtenida de Evans *et al.* 2000. Adaptada y traducida por Zaleta-Andrade para este trabajo de tesis.

También existen fluctuaciones en la dirección perpendicular al eje del campo de ondas continuas (*ibid*). En la figura 2.3 se puede observar como las fluctuaciones del

transductor que genera el rayo ultrasónico son rápidas en la zona cercana y lentas en la zona lejana.

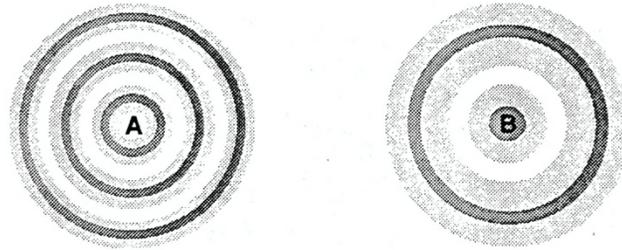


Figura 2.3: Variaciones de la intensidad de un campo de ondas continuas perpendicular al eje en los puntos A y B, representado por la escala de grises. Figura obtenida de Evans *et al.* 2000. Adaptada y traducida por Zaleta-Andrade para este trabajo de tesis.

La zona de recepción, de acuerdo con Evans *et al.* (2000) es idéntica en forma al campo de transmisión, ya que en transductores sin enfoque electrónico la recepción es lo contrario de la transmisión.

Rayos y campos ultrasónicos de onda pulsada – PW

Sustituir las técnicas experimentales por cálculos teóricos es posible (Duck 1981), ya que existe un buen consenso entre varios de los resultados calculados y los experimentales (Beaver 1974) para la simulación de la descripción completa de los campos ultrasónicos generados por transductores para aplicaciones médicas. Estos cálculos han sido desarrollados para elementos oscilatorios de varias formas (Cahill y Baker 1997, Duerinckx 1981, Jensen y Svendsen 1992, y Baker, Berb, Sahin, y Tijotta 1995). Evans *et al* (2000) provee un ejemplo de la variación de la intensidad del campo ultrasónico de ondas pulsadas a lo largo de su eje utilizando un cálculo numérico. Esto se muestra en la figura 2.4.

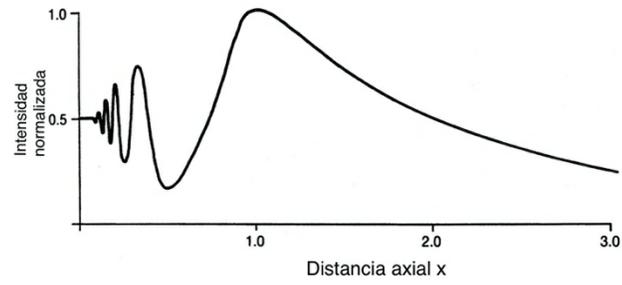


Figura 2.4: Variación de la intensidad de un campo de ondas pulsadas a lo largo de su eje, correspondiente a tres ciclos. Figura obtenida de Evans *et al.* 2000. Adaptada y traducida por Zaleta-Andrade para este trabajo de tesis.

2.3 Reconstrucción de la sección transversal de un rayo ultrasónico obtenido experimentalmente utilizando Matlab®

El resultado del proceso de caracterización del rayo ultrasónico de este sub-capítulo es producto de la experimentación llevada a cabo por RJ el Hospital Universitario de Leicester NHS Trust (Reino Unido) en 2019, en colaboración con la Universidad de Leicester. La experimentación se describe de forma general debido a que la divulgación de la caracterización de los sistemas utilizados, y otros de sus detalles, no han sido autorizados por el Hospital Universitario de Leicester NHS Trust ni por la Universidad de Leicester.

Equipo de laboratorio a utilizar: descripción general

En el experimento se colocó un hidrófono y un transductor, separados a una cierta distancia el uno del otro, alineados con ayuda de un sistema de posicionamiento con precisión aparente de 3 mm dentro de un tanque tipo pecera lleno de agua. La razón por la que esta técnica se hace en agua es porque la atenuación del ultrasonido en este medio acuoso es muy mínima, ya que el coeficiente de atenuación es de $0.002 \text{ dB} \cdot \text{cm}^{-1}$ (*ibid*). A continuación, se muestra una fotografía y un diagrama que se apega a la configuración del experimento de la radióloga (figura 2.5 y 2.6) (University of Erlangen 2014).

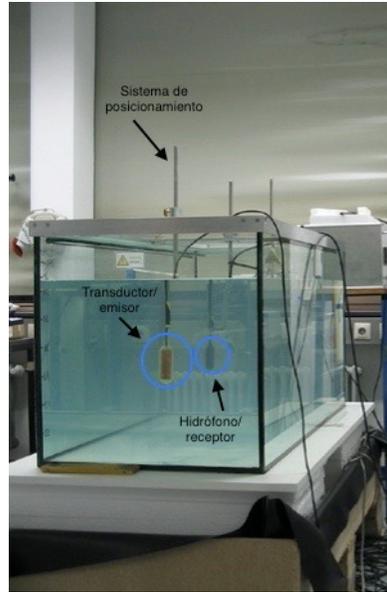


Figura 2.5: Foto de una configuración similar al experimento de RJ. Figura obtenida de University of Erlangen (2000). Adaptada por Zaleta-Andrade para este trabajo de tesis.

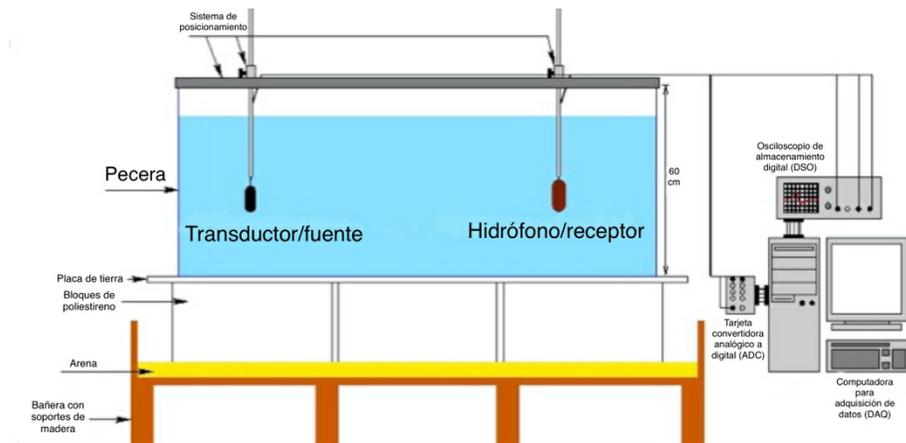


Figura 2.6: Diagrama de una configuración similar al experimento de RJ. Figura obtenida de University of Erlangen (2000). Adaptada y traducida por Zaleta-Andrade para este trabajo de tesis.

Proceso de captura de información: ejecución del experimento

En el experimento de RJ, el transductor se mantiene fijo y el único que utiliza el sistema de posicionamiento es el hidrófono. La razón es porque la sección transversal del rayo ultrasónico emitido por el transductor es mayor que la sección transversal receptora del hidrófono, por lo que el experimento consiste en mover el hidrófono de tal manera que se logre capturar la totalidad de la sección transversal del haz de ultrasonido.

El sistema de posicionamiento permite mover al hidrófono a lo largo y ancho de los ejes tridimensionales. La precisión de los pasos de cada movimiento del sistema de posicionamiento del experimento de RJ es de 3 mm. La posición inicial Z del hidrófono, de acuerdo a la colección de datos del experimento, es en 20-0 mm. El número 0 – a la derecha – de esta notación es el punto de referencia del eje Z, el cual puede estar tan cercano a las paredes de la pecera como el experimentador lo requiera. El número de la izquierda indica la posición actual del hidrófono. Esta notación se vuelve más compleja cuando se añaden las posiciones iniciales X y Y, las cuales tienen su referencia justo en el centro del sistema de posicionamiento y son denotadas como X1 y Y1. Por lo tanto, debido a que el experimento de Justyna tiene una posición inicial en el eje Z de 20-0 mm, X de -9 mm, y Y de -9 mm, ambas respecto al centro del sistema de posicionamiento, la notación para describir la posición inicial del hidrófono es: Z20-Y1-X1.

La ejecución del experimento de Justyna consiste en posicionar al hidrófono en el punto inicial Z20-Y1-X1, para después emitir el rayo ultrasónico y que el hidrófono detecte las perturbaciones del medio acuoso generadas por el ultrasonido para guardar estos datos en la computadora DAQ – proceso conocido como *adquisición de datos*. En seguida, el hidrófono da un paso en el eje X, describiendo ahora una posición: Z20-Y1-X2. Este proceso de adquisición de datos se repite a conveniencia del experimentador. De esta manera, el experimento de Justyna logra capturar toda la información necesaria para caracterizar un rayo ultrasónico de forma digital.

Estructura de datos de archivos del experimento: a detalle

La computadora DAQ almacena tres archivos diferentes: dos en formato *.tdt y otro en *.txt. Los archivos *.tdt contienen información relevante del sistema de posicionamiento que mueve al hidrófono: los valores iniciales de posicionamiento de cada uno de los ejes tridimensionales, la precisión del paso de los movimientos del hidrófono, el número total de pasos del hidrófono, y su posición actual en el eje Z. Estos valores están dados en milímetros y al conjunto de toda esta información se le denomina “encabezado” o “header”. Los archivos *.txt contienen las posiciones exactas del hidrófono en todos los ejes (en milímetros), así como el mayor valor pico del rayo ultrasónico detectado y su valor RMS (raíz media cuadrada por sus siglas en idioma inglés). Estos valores están dados en mili-voltios.

Los archivos *.tdt contienen una malla de datos de campo libre que no fueron relevantes para los resultados que el experimento de Justyna requería del autor de este trabajo de tesis. Entonces, desde que las dos versiones de archivos contienen los mismos datos de encabezado y solamente difieren en la malla de datos de campo libre, y debido a que estos datos no se utilizan, la extracción puede llevarse a cabo despreciando uno de los tipos de archivos *.tdt. Es posible que uno de los archivos *.tdt contenga los valores picos de las magnitudes de las perturbaciones ocasionadas por el rayo ultrasónico debido al nombre que reciben. *E.g.*, “RmsZ-20-0mm”.

Los archivos *.txt contienen la información necesaria para recrear el rayo ultrasónico mediante la relación tiempo y magnitud: la frecuencia de muestreo (en MHz), y la resolución (en bits) de la tarjeta ADC, así como el número de muestras obtenidas junto a cada una de sus amplitudes (en mili-volts). Sin embargo, estos datos no fueron relevantes para los resultados que el experimento de Justyna requería del autor de este trabajo de tesis.

Las figuras 2.7, 2.8, y 2.9 muestran una parte de cómo es que el procesador de palabras y editor de texto TextEdit® (incluido en el sistema operativo macOS) muestran la información de los archivos *.tdt y *.txt antes mencionados.

```

RmsZ-20-0mm.tdt
PmMagnetic_In_In_w
V_Start(m)= -9.000000
V_End(m)= 9.000000
Z_Start(m)= 20.000000
Z_End(m)= 20.000000
V_Step(m)= 37
Z_Step(m)= 37
X_Start(m)= 0.500000
X_End(m)= 0.500000
Z_Step(m)= 1.000000
Z_CurrentPost(m)= 20.000000
79.857939 76.451823 78.236743 59.218478 65.952819 65.659939 43.856258 59.656158 50.537979 65.975445 43.184416
59.724080 64.813726 53.224742 58.227274 78.812159 79.487987 81.346395 86.954480 79.238619 74.589714 75.080834
64.667177 59.998542 68.599578 68.935567 64.488196 66.971433 31.634386 66.924631 51.839325 51.481325 67.334241
51.794154 61.845556 68.780526 67.482266
77.648748 71.491488 57.471732 67.827955 63.978452 49.249042 49.815112 64.378544 58.181344 79.744142 53.446673
59.294797 81.849059 89.200234 92.717158 88.288329 188.837737 188.728411 188.746323 86.886644 95.360889 81.205935
88.338448 88.276436 58.876858 64.797665 65.282441 58.835496 59.547848 68.861446 68.845172 68.885813 68.799596
88.688987 67.261920 63.814539 67.426756
79.984889 57.281542 69.733376 58.834338 63.866978 57.945891 64.811962 64.858883 53.174887 68.823711 84.965968
84.161882 89.281516 186.493728 189.449739 113.488934 113.488925 114.888323 112.283448 113.896647 111.718922 188.985404
184.811388 99.746682 85.324754 86.158458 62.293434 67.226622 42.480277 58.197952 65.437983 58.758749 59.885485
58.953288 59.283372 65.797373 65.641272
63.163998 71.218643 66.588653 62.867891 63.134727 63.281868 57.965883 58.815135 84.443269 93.828368 181.863897
187.647818 115.242744 164.181958 164.997386 168.818078 169.959643 167.497843 167.431668 166.282351 163.765211 139.461977
115.854929 113.863809 188.835153 183.878382 94.159788 83.635575 56.344393 66.419924 58.349187 65.166118 86.743438
69.428644 65.818652 82.214749 58.892254
74.521977 62.476521 49.133468 56.779335 63.159858 51.888333 61.841223 88.286572 99.358222 187.381852 136.582698
187.647818 115.242744 164.181958 164.997386 168.818078 169.959643 167.497843 167.431668 166.282351 163.765211 139.461977
166.868487 161.888252 162.858873 139.994435 114.743246 118.537391 108.376548 87.854171 56.878544 63.632535 65.293137
57.214682 57.283372 63.892738 57.79445
54.782895 49.868994 58.853316 58.613486 63.976482 92.776479 186.331613 113.842553 142.179464 178.697499 173.886523
77.782187 185.816483 158.839943 148.729758 248.488883 113.818649 248.947795 228.923292 228.288884 129.588822 138.287823
166.331314 168.577438 156.699932 168.378688 137.448854 115.712828 111.918888 181.889978 84.588888 55.545477 65.293137
69.489784 59.283372 59.731643 59.442124
64.179483 57.858367 58.389795 63.481317 99.769156 187.288428 117.332287 146.678528 171.467988 171.488662 187.883668
71.545898 187.888882 148.822884 114.957451 248.488883 113.818649 248.947795 228.923292 228.288884 129.588822 138.287823
115.256882 159.288886 142.293255 148.838581 188.122115 156.798375 114.588444 187.833525 94.831726 68.417932 63.632535
69.489784 69.489784 59.218478 89.898924 186.733469 116.268861 122.671946 172.877226 169.193953 162.379234 151.189575
64.129878 63.958958 59.218384 69.898924 186.733469 116.268861 122.671946 172.877226 169.193953 162.379234 151.189575
71.545898 187.888882 148.822884 114.957451 248.488883 113.818649 248.947795 228.923292 228.288884 129.588822 138.287823
127.892423 146.825138 117.855977 133.855778 148.939661 156.193958 158.372514 114.678597 186.143815 91.881212 97.212847
69.489784 69.489784 59.218478 89.898924 186.733469 116.268861 122.671946 172.877226 169.193953 162.379234 151.189575
65.492276 59.331586 86.978499 189.388735 117.584813 123.888726 172.621687 171.648831 168.826119 143.466583 122.289156
71.545898 187.888882 148.822884 114.957451 248.488883 113.818649 248.947795 228.923292 228.288884 129.588822 138.287823
178.741643 116.778488 136.888283 115.672139 137.549218 151.157889 168.169763 155.925319 111.978466 99.778826 83.995918
69.489784 69.489784 59.218478 89.898924 186.733469 116.268861 122.671946 172.877226 169.193953 162.379234 151.189575
58.288737 63.777371 97.389881 111.618889 123.128885 127.178885 172.818886 166.338851 158.788853 127.288889 164.178818
186.331314 168.577438 156.699932 168.378688 137.448854 115.712828 111.918888 181.889978 84.588888 55.545477 65.293137
289.118563 198.218359 117.248827 133.787812 116.747484 148.153889 155.618731 168.294858 158.463339 189.223123 97.212847
78.211497 69.489784 59.218478 89.898924 186.733469 116.268861 122.671946 172.877226 169.193953 162.379234 151.189575
57.267828 74.358131 189.137385 123.858218 128.928218 174.888858 178.522387 154.418841 131.582828 159.242317 146.534222
186.331314 168.577438 156.699932 168.378688 137.448854 115.712828 111.918888 181.889978 84.588888 55.545477 65.293137
328.828848 222.812435 154.588838 118.228799 146.719577 127.979844 158.782887 161.189782 161.817983 113.861723 182.817983

```

Figura 2.7: Parte de un archivo *.tdt con nombre “RmsZ-20-0mm.tdt” con datos obtenidos del experimento de Justyna.

```

PmZ-23-0mm.tdt
PmMagnetic_In_In_w
V_Start(m)= -9.000000
V_End(m)= 9.000000
Z_Start(m)= 20.000000
Z_End(m)= 20.000000
V_Step(m)= 37
Z_Step(m)= 37
X_Start(m)= 0.500000
X_End(m)= 0.500000
Z_Step(m)= 1.000000
Z_CurrentPost(m)= 23.000000
211.000000 136.000000 75.800000 138.000000 130.000000 136.000000 131.000000 131.000000 144.200000 125.800000 125.000000
216.000000 256.200000 294.200000 154.000000 332.000000 343.000000 358.000000 349.000000 345.000000 334.000000 328.000000
227.000000 255.200000 243.000000 140.000000 332.000000 338.000000 339.200000 338.000000 349.000000 349.000000 331.000000
136.000000 136.000000 129.200000 141.000000
228.200000 136.000000 140.000000 131.000000 132.000000 138.000000 135.000000 285.000000 224.600000 284.000000 284.000000
318.000000 346.000000 369.000000 387.000000 482.000000 411.000000 418.000000 416.000000 411.200000 483.000000 388.000000
359.000000 363.000000 352.000000 362.000000 237.000000 139.000000 148.200000 139.000000 162.200000 137.600000 362.000000
139.000000 141.200000 134.000000 134.000000
132.000000 252.000000 232.000000 132.000000 131.000000 124.000000 138.000000 236.200000 278.000000 313.000000 349.000000
379.000000 482.200000 419.200000 432.200000 441.000000 446.000000 452.000000 455.000000 452.000000 443.000000 436.000000
71.545898 187.888882 148.822884 114.957451 248.488883 113.818649 248.947795 228.923292 228.288884 129.588822 138.287823
134.000000 134.200000 133.000000 133.000000
425.000000 429.000000 425.000000 389.000000 385.000000 361.000000 328.000000 268.000000 244.000000 274.000000 484.000000
445.000000 458.000000 488.000000 586.000000 521.200000 526.000000 531.000000 522.800000 519.000000 489.000000 463.000000
463.000000 493.000000 425.000000 389.000000 385.000000 361.000000 328.000000 268.000000 244.000000 274.000000 484.000000
171.200000 129.200000 127.200000 125.000000
409.000000 537.000000 503.000000 577.000000 586.000000 588.200000 583.200000 579.400000 569.000000 548.000000 528.000000
281.000000 602.000000 458.000000 413.000000 398.000000 374.000000 348.000000 293.200000 248.000000 163.000000 141.200000
134.000000 147.000000 134.000000 135.000000
575.000000 555.000000 682.200000 680.000000 587.000000 597.000000 565.000000 579.000000 557.000000 547.000000 535.000000
52.788212 113.000000 111.400000 126.000000
132.000000 132.000000 132.000000 131.000000 131.000000 124.000000 138.000000 236.200000 278.000000 313.000000 349.000000
368.000000 429.000000 425.000000 389.000000 385.000000 361.000000 328.000000 268.000000 244.000000 274.000000 484.000000
409.000000 537.000000 503.000000 577.000000 586.000000 588.200000 583.200000 579.400000 569.000000 548.000000 528.000000
463.000000 493.000000 425.000000 389.000000 385.000000 361.000000 328.000000 268.000000 244.000000 274.000000 484.000000
134.000000 147.000000 134.000000 135.000000
575.000000 555.000000 682.200000 680.000000 587.000000 597.000000 565.000000 579.000000 557.000000 547.000000 535.000000
52.788212 113.000000 111.400000 126.000000
132.000000 132.000000 132.000000 131.000000 131.000000 124.000000 138.000000 236.200000 278.000000 313.000000 349.000000
368.000000 429.000000 425.000000 389.000000 385.000000 361.000000 328.000000 268.000000 244.000000 274.000000 484.000000
409.000000 537.000000 503.000000 577.000000 586.000000 588.200000 583.200000 579.400000 569.000000 548.000000 528.000000
463.000000 493.000000 425.000000 389.000000 385.000000 361.000000 328.000000 268.000000 244.000000 274.000000 484.000000
134.000000 147.000000 134.000000 135.000000
575.000000 555.000000 682.200000 680.000000 587.000000 597.000000 565.000000 579.000000 557.000000 547.000000 535.000000
52.788212 113.000000 111.400000 126.000000
132.000000 132.000000 132.000000 131.000000 131.000000 124.000000 138.000000 236.200000 278.000000 313.000000 349.000000
368.000000 429.000000 425.000000 389.000000 385.000000 361.000000 328.000000 268.000000 244.000000 274.000000 484.000000
409.000000 537.000000 503.000000 577.000000 586.000000 588.200000 583.200000 579.400000 569.000000 548.000000 528.000000
463.000000 493.000000 425.000000 389.000000 385.000000 361.000000 328.000000 268.000000 244.000000 274.000000 484.000000
134.000000 147.000000 134.000000 135.000000
575.000000 555.000000 682.200000 680.000000 587.000000 597.000000 565.000000 579.000000 557.000000 547.000000 535.000000
52.788212 113.000000 111.400000 126.000000
132.000000 132.000000 132.000000 131.000000 131.000000 124.000000 138.000000 236.200000 278.000000 313.000000 349.000000
368.000000 429.000000 425.000000 389.000000 385.000000 361.000000 328.000000 268.000000 244.000000 274.000000 484.000000
409.000000 537.000000 503.000000 577.000000 586.000000 588.200000 583.200000 579.400000 569.000000 548.000000 528.000000
463.000000 493.000000 425.000000 389.000000 385.000000 361.000000 328.000000 268.000000 244.000000 274.000000 484.000000
134.000000 147.000000 134.000000 135.000000
575.000000 555.000000 682.200000 680.000000 587.000000 597.000000 565.000000 579.000000 557.000000 547.000000 535.000000
52.788212 113.000000 111.400000 126.000000
132.000000 132.000000 132.000000 131.000000 131.000000 124.000000 138.000000 236.200000 278.000000 313.000000 349.000000
368.000000 429.000000 425.000000 389.000000 385.000000 361.000000 328.000000 268.000000 244.000000 274.000000 484.000000
409.000000 537.000000 503.000000 577.000000 586.000000 588.200000 583.200000 579.400000 569.000000 548.000000 528.000000
463.000000 493.000000 425.000000 389.000000 385.000000 361.000000 328.000000 268.000000 244.000000 274.000000 484.000000
134.000000 147.000000 134.000000 135.000000
575.000000 555.000000 682.200000 680.000000 587.000000 597.000000 565.000000 579.000000 557.000000 547.000000 535.000000
52.788212 113.000000 111.400000 126.000000
132.000000 132.000000 132.000000 131.000000 131.000000 124.000000 138.000000 236.200000 278.000000 313.000000 349.000000
368.000000 429.000000 425.000000 389.000000 385.000000 361.000000 328.000000 268.000000 244.000000 274.000000 484.000000
409.000000 537.000000 503.000000 577.000000 586.000000 588.200000 583.200000 579.400000 569.000000 548.000000 528.000000
463.000000 493.000000 425.000000 389.000000 385.000000 361.000000 328.000000 268.000000 244.000000 274.000000 484.000000
134.000000 147.000000 134.000000 135.000000
575.000000 555.000000 682.200000 680.000000 587.000000 597.000000 565.000000 579.000000 557.000000 547.000000 535.000000
52.788212 113.000000 111.400000 126.000000
132.000000 132.000000 132.000000 131.000000 131.000000 124.000000 138.000000 236.200000 278.000000 313.000000 349.000000
368.000000 429.000000 425.000000 389.000000 385.000000 361.000000 328.000000 268.000000 244.000000 274.000000 484.000000
409.000000 537.000000 503.000000 577.000000 586.000000 588.200000 583.200000 579.400000 569.000000 548.000000 528.000000
463.000000 493.000000 425.000000 389.000000 385.000000 361.000000 328.000000 268.000000 244.000000 274.000000 484.000000
134.000000 147.000000 134.000000 135.000000
575.000000 555.000000 682.200000 680.000000 587.000000 597.000000 565.000000 579.000000 557.000000 547.000000 535.000000
52.788212 113.000000 111.400000 126.000000
132.000000 132.000000 132.000000 131.000000 131.000000 124.000000 138.000000 236.200000 278.000000 313.000000 349.000000
368.000000 429.000000 425.000000 389.000000 385.000000 361.000000 328.000000 268.000000 244.000000 274.000000 484.000000
409.000000 537.000000 503.000000 577.000000 586.000000 588.200000 583.200000 579.400000 569.000000 548.000000 528.000000
463.000000 493.000000 425.000000 389.000000 385.000000 361.000000 328.000000 268.000000 244.000000 2
```

```
Scan_Mode: Normal_3D_scan_mode
Signal_Magnitude_1v_1u_mv
X(mm)= 7.5
Y(mm)= -5.4
Z(mm)= 44.4
Detected_Peak_Mean_within_a_80ns_Window_(mv)= 378.400000
Detected_Root_Mean_Square_within_8.5_12_Micro_Second_Window_(mv)= 118.333346
Sampling_Frequency(MHz)= 62.5
No_of_Samples= 15500
225.000000
247.000000
234.000000
214.000000
127.000000
149.000000
181.000000
22.000000
-1.000000
-51.000000
-113.000000
-178.000000
-216.000000
-259.000000
-297.000000
-325.000000
-348.000000
-384.000000
-337.000000
-318.000000
-288.000000
-248.000000
-292.000000
-151.000000
-85.000000
-71.000000
42.000000
115.000000
181.000000
181.000000
297.000000
334.000000
361.000000
366.000000
329.000000
331.000000
321.000000
248.000000
182.000000
119.000000
51.000000
-15.000000
-79.000000
-143.000000
-196.000000
-249.000000
-283.000000
-325.000000
-358.000000
-361.000000
-355.000000
-314.000000
```

Figura 2.9: Parte de un archivo *.txt con nombre “RF-Z-44-0mm-Y8-X33.txt” con datos obtenidos del experimento de RJ.

2.4 Estructura del código de reconstrucción de la sección transversal de un rayo ultrasónico obtenido experimentalmente usando Matlab®

La estructura del código de reconstrucción del haz de ultrasonido se divide en dos partes fundamentales: la lectura de archivos y la reconstrucción del haz ultrasónico. La parte de lectura de archivos reconoce los de tipo *.tdt y *.txt, los identifica y les extrae la información relevante de sus encabezados e ignora datos irrelevantes para el experimento de Justyna. Por ejemplo, el código detecta e ignora los valores de las magnitudes, o datos de campo libres, que provienen del muestreo de la señal de ultrasonido.

La parte de reconstrucción del haz ultrasónico lleva a cabo una refactorización de los datos de los archivos *.tdt y *.txt para ordenarlos matricialmente, con el propósito de mejorar su procesamiento en Matlab®. Este procesamiento permite reconstruir el haz ultrasónico en dos dimensiones, siendo de principal interés mostrar una correlación de los valores pico del haz ultrasónico entre los ejes X y Y.

Lectura de archivos *.tdt y *.txt con Matlab®

Para que el código de reconstrucción logre reconocer los archivos *.tdt y *.txt, así como identificarlos, se construyen dos funciones llamadas *tdtread.m* y *txtread.m*. Estas dos funciones se encuentran integradas en otra función llamada *filereader.m*. Esta última función es la encargada de llevar a cabo la refactorización de datos de un archivo a la vez, ya sea *.tdt o *.txt, utilizando tres matrices que se describen a detalle en la tabla 2.3.

Nombre de la matriz	Descripción de la matriz
header	Salva la información de los encabezados.
data	Salva la información de los datos de campo.
extension	Salva la extensión del archivo como un identificador.

Tabla 2.3: Tabla de las matrices que contienen los datos de la refactorización de los archivos *.tdt o *.txt.

El centro de ayuda de MathWorks® para América Latina contiene la documentación completa y actualizada de todas las funcionalidades de Matlab®. Cada una de las funciones presentadas en este trabajo de tesis, nativas de Matlab®, contienen más argumentos de los presentados. Estos argumentos se pueden encontrar de forma extensa en la documentación del centro de ayuda de MathWorks® en: <https://la.mathworks.com/help/matlab/index.html>. Las funciones específicas y más importantes de la función *filereader.m* son descritas de forma general en la tabla 2.4.

Nombre de la función	Descripción general de la función

<code>f=fullfile(partearch1,...,partearchN)</code>	Construye nombre completo de archivo a partir de partes.
<code>tf=isnumeric(A)</code>	Determina si la entrada es una matriz numérica.

Tabla 2.4: Descripción general de las funciones de Matlab® más importantes utilizadas en `filereader.m`. Información obtenida del centro de ayuda de MathWorks® para América Latina.

La función `tdtread.m` abre un archivo `*.tdt` para su lectura. Extrae y guarda la información del encabezado y discrimina la que no es relevante mediante una serie de iteraciones entre las líneas de texto de este archivo. De la misma manera, extrae y guarda la información de los datos de campo que, si bien no son relevantes para lo que el experimento de Justyna requería del autor de este trabajo de tesis, presentan el potencial de mostrar la relación *tiempo de muestreo* y *magnitud de la muestra* en las futuras versiones de la GUI del programa de reconstrucción del rayo ultrasónico.

La función `txtread.m` parece más compleja en comparación con la función `tdtread.m` debido al número más extenso de líneas de este código, consecuencia de los caracteres de cada línea de los archivos `*.tdt`. Sin embargo, estas dos funciones – *txtread.m* y *tdtread.m* – llevan a cabo el mismo objetivo: extraer y guardar la información del encabezado y datos de campo de los archivos y discriminar su información irrelevante. Las funciones específicas más importantes que las funciones `tdtread.m` y `txtread.m` implementan son descritas de forma general en la tabla 2.5.

Nombre de la función	Descripción general de la función
fopen(nombreach)	Abre el archivo, nombreach, para acceso de lectura binaria.
zeros(n)	Crea una matriz de ceros.
fgetl(IDarchivo)	Lee líneas de archivos previamente abiertos e identificados por fopen(nombreach), eliminando caracteres de nuevas líneas.
strfind(cadena,patrón)	Encuentra cadenas dentro de otras cadenas.
dlmread(nombre_archivo)	Lee archivos delimitados por ASCII (por ejemplo, números reales)
fclose(IDarchivo)	Cierra uno o todos los archivos abiertos.
fscanf(IDarchivo, formato)	Lee datos desde un archivo de texto de acuerdo al formato especificado por el argumento de la función "formato"

Tabla 2.5: Descripción general de las funciones de Matlab® más importantes, utilizadas en tdtread.m y txtread.m. Información obtenida del centro de ayuda de MathWorks® para América Latina.

Gráfica de la sección transversal del haz ultrasónico en dos dimensiones: última parte del proceso de refactorización de datos

La función *main2.m* lee un archivo del formato tipo “RF-Z-44-0mm-Y8-X33.txt” y extrae solamente los valores RMS del encabezado de cada uno de los 37 archivos en total que contienen información de una sola posición en el eje Z. Estos archivos deben estar en una carpeta llamada “RF” – por recomendación – y el primer archivo que se seleccione debe corresponder a la posición X0 y Y0. En otras palabras, el formato tipo debe llamarse, por ejemplo, “RF-Z-44-0mm-Y0-X0.txt”. Un archivo *.mat con una matriz de datos de 37 x 37 (para el caso del experimento de Justyna) es el resultado de este proceso y debe ser guardado en un lugar de acceso conocido por el usuario, ya que este archivo *.mat contiene parte de la información necesaria para la gráfica de la sección transversal del haz ultrasónico en dos dimensiones. Las funciones específicas y más importantes que la función *main2.m* implementa son descritas de forma general en la tabla 2.6.

Nombre de la función	Descripción general de la función
zeros(n)	Crea una matriz de ceros.
cell()	Crea una estructura de datos del tipo matriz de celdas.
strfind(cadena,patron)	Encuentra cadenas dentro de otras cadenas.
strcat(s1,...,sN)	Concatena cadenas horizontalmente.
save(nombre_archivo)	Guarda variables del espacio de trabajo en un archivo *.mat.

Tabla 2.6: Descripción general de las funciones de Matlab® más importantes utilizadas en *main2.m*. Información obtenida del centro de ayuda de MathWorks® para América Latina.

Interfaz gráfica de usuario (GUI) del código de reconstrucción de la sección transversal de un rayo ultrasónico obtenido experimentalmente utilizando Matlab®

Las gráficas de sección transversal del haz de ultrasonido se obtienen utilizando la información de los archivos *.tdt, de acuerdo a lo que Justyna requería del autor de este trabajo de tesis. Se deja de lado el uso de datos de los archivos *.txt solamente para graficar la intensidad del rayo ultrasónico en el dominio del tiempo como una funcionalidad adicional a lo requerido por Justyna para la caracterización del rayo ultrasónico.

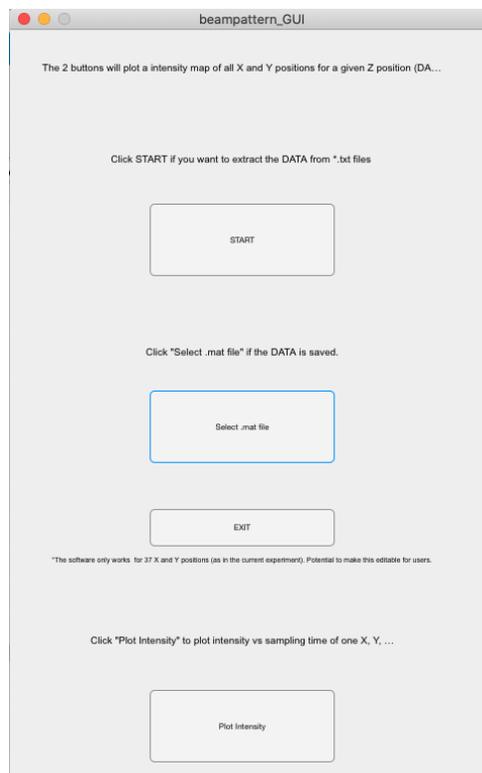


Figura 2.10: Interfaz gráfica de usuario (GUI) en Matlab® para la reconstrucción de la sección transversal de un rayo ultrasónico.

Nombre del botón	Funcionalidad
START	Implementa la funcionalidad de main2.m y crea un archivo *.mat con la información necesaria para reconstruir

	<p>la sección transversal del rayo ultrasónico mediante un mapa de intensidades. Este proceso toma 13 minutos y 49 segundos en una computadora comercial MacBook Air de inicios del 2015. El resultado se muestra en la figura 2.11.</p>
Select .mat file	<p>Pide insertar el archivo *.mat con la información necesaria para reconstruir la sección transversal del haz de ultrasonido. Esta funcionalidad tiene el potencial de que los 13 minutos y 19 segundos que el evento START toma para crear el archivo *.mat sean necesarios solamente una vez.</p>
EXIT	<p>Termina la ejecución de la GUI.</p>
Plot intensity	<p>Utiliza los datos de campo, mediante la función txtread.m, para mostrar la relación de los tiempos de muestreo del rayo ultrasónico y sus respectivas magnitudes.</p>

Tabla 2.7: Descripción de la funcionalidad de cada uno de los botones de la GUI para la reconstrucción de la sección transversal de un rayo ultrasónico.

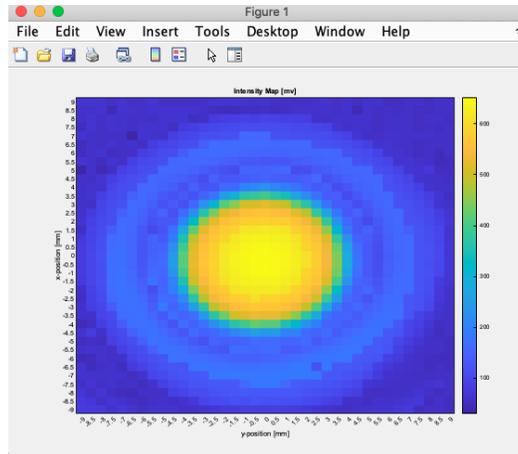


Figura 2.11: Sección transversal de un rayo ultrasónico en la posición Z-20-0mm. Resultado de los eventos “START” y “Select .mat file”.

La imagen anterior no muestra la posición Z del hidrófono, por lo que se convierte en una posible mejora para la GUI. Además, el programa en cuestión solo funciona para archivos con 37 posiciones, ajustándose a los requerimientos del experimento de Justyna. Sin embargo, tiene el potencial de mejorar el algoritmo de main2.m para reconstruir rayos ultrasónicos con infinitas posiciones.

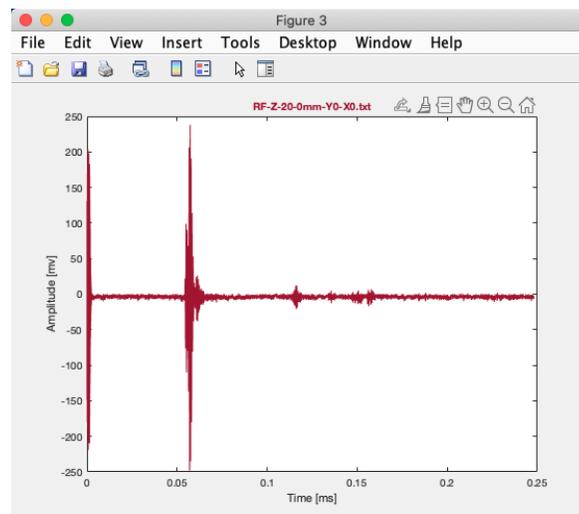


Figura 2.12: Relación de los tiempos de muestreo y las amplitudes de cada muestra del haz de ultrasonido.

2.5 Frecuencia Doppler en Sistemas CW

El principio del efecto Doppler establece que la frecuencia del eco reflejado desde un objetivo en movimiento es diferente a la frecuencia incidente (Atkinson y Wells 1977 y Toman 1984). Si un transductor emite un ultrasonido que viaje a través de los vasos sanguíneos, las células sanguíneas en movimiento retrodispersan la señal nuevamente hacia el transductor con una frecuencia diferente. Esto se puede apreciar en la figura 2.13.

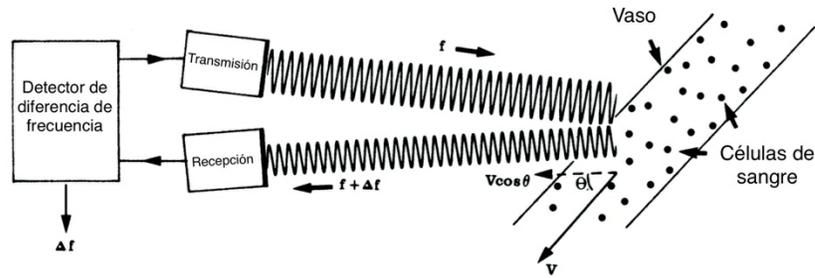


Figura 2.13: Ultrasonido Doppler de onda continua siendo insonado en un vaso sanguíneo. Figura obtenida de Atkinson *et al.* 1977. Adaptada y traducida por Zaleta-Andrade para este trabajo de tesis.

El cambio de frecuencia es detectado por la comparación de la frecuencia de transmisión y la frecuencia de recepción de la señal (Atkinson *et al.* 1977) El cambio de frecuencia – o frecuencia Doppler – se define como:

$$\Delta f = 2f \frac{V * \cos\theta}{c} \quad \text{Ec.(2.2)}$$

Δf es el cambio de frecuencia

f es la frecuencia incidente

$V * \cos\theta$ es la velocidad del objetivo en la dirección del transductor

c es la velocidad del ultrasonido

2.6 Sistemas Doppler de Onda Continua – CW

Un sistema Doppler CW emite un haz de ultrasonido continuo, normalmente enfocado a una profundidad específica. La sonda de este sistema usa dos transductores: uno para el proceso de insonación y otro transductor separado para recibir la retrodispersión del ultrasonido insonado (señal Doppler). La medición de las profundidades (o barreras) utilizando un sistema Doppler CW no es posible debido a las características del haz de ultrasonido continuo. Los sistemas CW tienen la ventaja de ser capaces de medir los desfases de la frecuencia Doppler sin interrupciones; a

diferencia de los sistemas PW, en los cuales, la medición de la frecuencia Doppler está restringida por la frecuencia de repetición del pulso (John-Keeton 1997) – PRF.

Un sistema típico Doppler CW comienza su operación mediante un oscilador que opera a una frecuencia constante y acciona un cristal transmisor de la sonda a través de un amplificador transmisor para lograr la insonación del ultrasonido. La retrodispersión, que contiene el eco de los objetivos móviles y estacionarios, alimenta al amplificador de radio frecuencia desde el cristal receptor (Evans *et al.* 2000). Este eco pasa a través de una etapa de detección para eliminar la señal portadora y dejar solo las componentes de desfase de la frecuencia Doppler. Posteriormente, el eco se somete a un contador de cruce por ceros – etapa de conversión de frecuencia a voltaje – para producir una señal de salida estimada que represente los desfases de la frecuencia Doppler para estimar la velocidad del flujo. Cualquier dispositivo que realice este proceso se conoce como medidor de flujo (Brody y Meindl 1974). En las figuras 2.14 y 2.15 se observan dos diagramas a bloques: un sistema típico Doppler CW y otro de un medidor de flujo de un sistema Doppler CW.

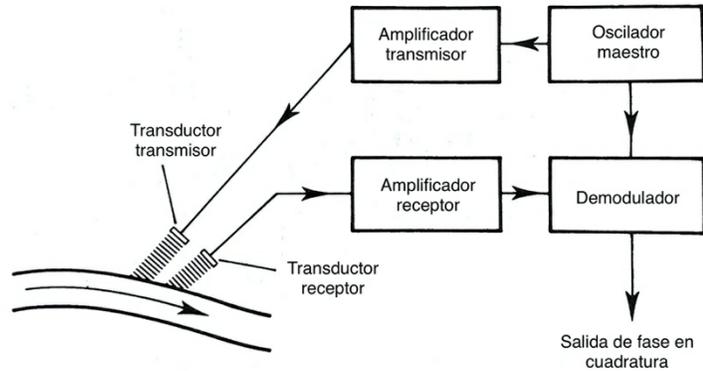


Figura 2.14: Diagrama a bloques de un sistema típico Doppler CW. Figura obtenida de Evans *et al.* 2000. Adaptada y traducida por Zaleta-Andrade para este trabajo de tesis.

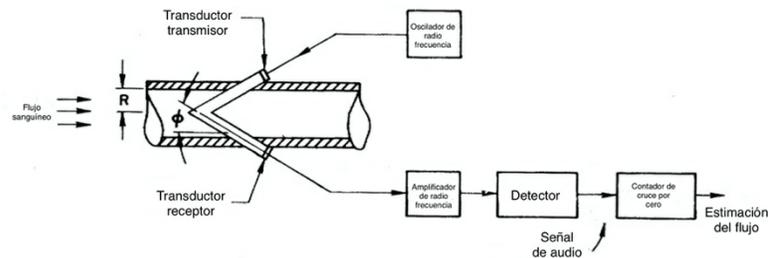


Figura 2.15: Diagrama a bloques de un medidor de flujo de un sistema Doppler CW. Figura obtenida de Brody *et al.* 1974. Adaptada y traducida por Zaleta-Andrade para este trabajo de tesis.

2.7 Sistemas Doppler de Onda Pulsada Multibarrera – PW

Los sistemas PW son diferentes a los CW principalmente porque el transductor del sistema PW se excita con una ráfaga de pulsos en lugar de estar continuamente excitado. Un sistema PW permite obtener información Doppler desde la cabeza del transductor hasta un rango determinado (Evans *et al.* 2000).

Tipos de sistemas Doppler PW

Existen dos tipos de sistemas Doppler PW: de una sola barrera y de multibarrera. Estos términos son comúnmente usados en idioma inglés como sistemas Doppler PW single-gate y multi-gate. Un sistema de una sola barrera permite seleccionar la dispersión deseada desde una determinada distancia del transductor. *I.e.*, permite seleccionar la profundidad desde la que se desea recopilar el eco del ultrasonido insonado; pero esto

presenta inconvenientes. Por ejemplo, los perfiles de velocidad durante un ciclo cardiaco pueden ser determinados solamente desde un solo lugar. Esto limita la aplicación de estos sistemas por errores encontrados en las estimaciones de estos perfiles (Reneman, van Merode, Hick, y Hoeks 1985, y Wille 1979) y dificultades de procesamiento de datos (Reneman, van Merode, Hick, y Hoeks 1986).

Los sistemas de multibarrera son sistemas Doppler más recientes y complejos, que han sido desarrollados para tener la habilidad de detectar de manera simultánea e instantánea las velocidades de un determinado rango de interés (Brandestini 1978, y Hoeks, Reneman, y Peronneau 1981). Con los sistemas multibarrera, los problemas de los perfiles de velocidad de los sistemas de una sola barrera se resuelven debido a que las velocidades pueden registrarse desde varios lugares a lo largo del camino del ultrasonido (Reneman *et al.* 1986). En la figura 2.16 y 2.17 se muestran, de manera respectiva, las formas de onda de las velocidades y perfiles de velocidad de un sistema multibarrera.

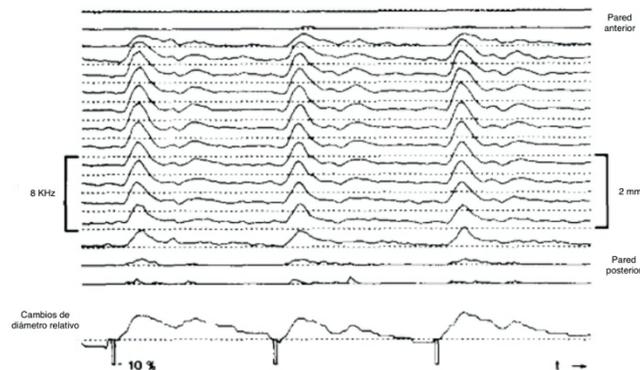


Figura 2.16: Formas de onda de las velocidades instantáneas registradas simultáneamente en varios lugares del camino del ultrasonido en una arteria carótida común de un sujeto sano. Figura obtenida de Reneman *et al.* 1986. Adaptada y traducida por Zaleta-Andrade para este trabajo de tesis.

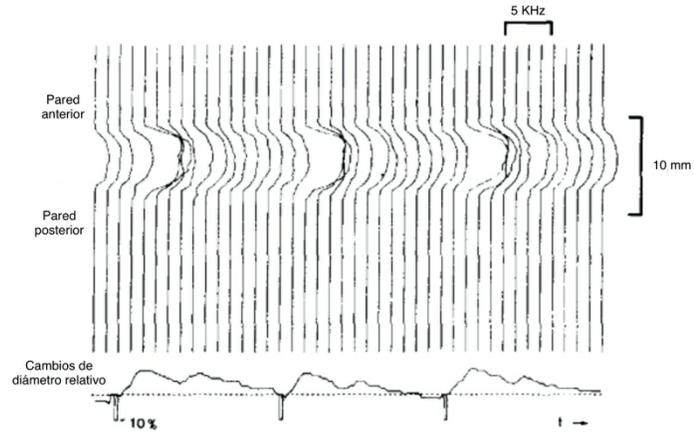


Figura 2.17: Los perfiles de velocidad axial en intervalos de tiempo discretos durante el ciclo cardiaco. Figura obtenida de Reneman *et al.* 1986. Adaptada y traducida por Zaleta-Andrade para este trabajo de tesis.

Procesamiento de señales en serie en sistemas Doppler PW multibarrera

Un sistema Doppler PW de una barrera puede convertirse en multibarrera duplicando los circuitos necesarios para el almacenamiento intermedio de las señales de fase, los filtros de señales Doppler y la detección de la velocidad promedio para cada uno de los canales adicionales.

Los sistemas PW multibarrera procesan el eco del ultrasonido en forma serial. Esto quiere decir que las señales son procesadas una tras otra entre las emisiones de los pulsos. El procesamiento de señales en serie evita el proceso tedioso del procesamiento en paralelo de comparar los diferentes canales entre sí – denominado sintonización de las barreras o frecuencias (Reneman *et al* 1986 y Hoeks 1982). En la figura 2.18 se muestra el principio del procesamiento de señales en serie y paralelo.

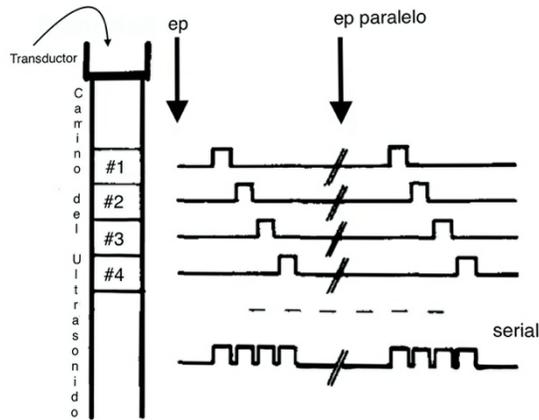


Figura 2.18: Representación esquemática del procesamiento de datos en paralelo y serie en dispositivos Doppler multibarrera. Los números representan los canales procesados (o barreras). ep es pulso de emisión (emission pulse en idioma inglés). Figura obtenida de Reneman *et al.* 1986. Adaptada y traducida por Zaleta-Andrade para este trabajo de tesis.

Proceso de obtención de información Doppler en sistemas Doppler PW de una barrera

El proceso de obtención de información Doppler comienza cuando el pulso RF es insonado para pasar a través del cuerpo. Esta señal de ultrasonido se desfasa respecto a la señal inicial debido a objetos a lo largo del camino del ultrasonido. Recordar el principio del efecto Doppler: “la frecuencia del eco reflejado desde un objetivo en movimiento será diferente de la frecuencia incidente” (Atkinson y Wells, 1977 y Toman 1984).

Los ecos que regresan tanto por el desfase de objetos en movimiento como estacionarios son recibidos por el mismo transductor. Este proceso es un ejemplo de procesamiento de señales en serie y se repite en cada pulso de ultrasonido (Evans *et al.* 2000). Posteriormente, la información Doppler tiene que extraerse del eco de cada una de las barreras. A este proceso se le conoce como demodulación y es igual para sistemas PW y CW. La señal resultante de este proceso se denomina “señal demodulada” y contiene la información rango-fase o frecuencia Doppler. *I.e.*, la diferencia de fase entre la señal de referencia y el eco recibido en el rango especificado.

La señal demodulada ahora se muestrea en puntos específicos de tiempo, relativos al inicio de pulso de transmisión. El tiempo de muestreo define la profundidad de interés y es elegido para corresponder al tiempo que toma el sonido en viajar de ida y vuelta del transductor hasta la profundidad deseada. Este proceso se logra a través de la actualización continua por pulso de un amplificador de muestreo y retención. El proceso final de filtrado emite la señal con la información de frecuencia de cambio Doppler del rango de interés (Reneman *et al.* 1986).

La demodulación analógica de la frecuencia Doppler de un sistema PW de una barrera

Una vez que el transductor detecta la señal, esta se amplifica en la sección del receptor. Esta amplificación depende del tiempo para corregir la atenuación creciente del ultrasonido reflejado y retrodispersado a profundidades mayores.

La diferencia de fase instantánea entre la señal detectada y la señal del oscilador en el detector de fase es derivada como una función continua de la profundidad – recordar que la derivada de la distancia respecto al tiempo es la velocidad. Esto se consigue filtrando de paso bajo el resultado de mezclar la señal detectada con la del oscilador. La duración efectiva de la respuesta impulso del filtro paso bajas (Gabor 1946) establece el ancho de la muestra de la barrera y los límites en combinación con la duración de la ráfaga emitida y la dimensión axial del volumen de muestreo. Mientras tanto, la dimensión lateral está dada por el ancho local de la señal de ultrasonido. Para lograr la mejor resolución axial y la mejor relación señal a ruido para una duración determinada de la emisión, el ancho de la muestra de la barrera debe coincidir con la duración de la emisión (Peronneau, Bournat, Bugon, Barbet, y Xhaard 1974).

En un sistema Doppler PW de una sola barrera, la salida del detector de fase de la señal Doppler, sin procesar y a una profundidad específica, se almacena en una memoria analógica después del retraso presente con respecto a la emisión. Al circuito encargado de este proceso se le conoce como circuito de muestreo y retención de las señales. Este circuito tiene una frecuencia de muestreo igual a la frecuencia de repetición de pulso (PRF) de la señal Doppler.

Para obtener una función continua en el tiempo, un filtro paso bajas debe ser empleado con una frecuencia de corte igual a la mitad de la máxima repetición de pulso. Las frecuencias de corte, y desde luego el tipo de filtro de un sistema Doppler, van a depender de los objetivos para los que el sistema haya sido pensado. Por ejemplo, el eco retrodispersado por objetos estacionarios o con poco movimiento pueden oscurecer las señales Doppler débiles provocadas por partículas de sangre. Para este caso, se necesita sustituir el filtro paso bajas por un paso bandas. El resultado es una señal Doppler filtrada en forma de una señal de audio.

Para obtener la frecuencia promedio de la señal Doppler, la cual es proporcional a la velocidad promedio del volumen de muestreo, la señal Doppler filtrada alimenta al contador de cruce por cero – que cuenta el número de veces que una señal cruza la línea de referencia 0. La duración efectiva del conteo depende de la frecuencia característica del filtro paso bajas que suaviza los pulsos producidos por el detector de cruce por cero.

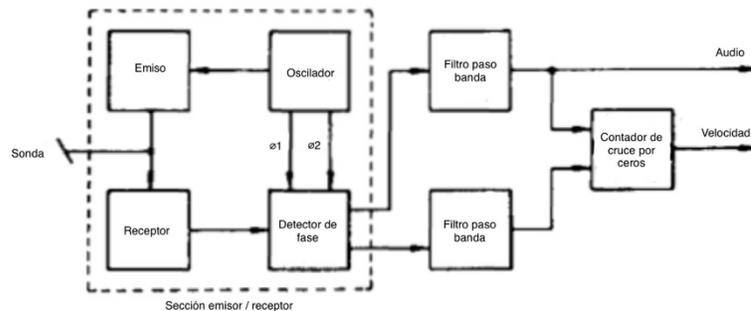


Figura 2.19: Diagrama a bloques de un sistema Doppler PW de una sola barrera. Las señales $\varnothing 1$ y $\varnothing 2$ representan entradas de referencia en cuadratura del detector de fase (que en realidad contiene dos detectores de fase). Figura obtenida de Hoeks, Reneman, y Peronneau 1981. Adaptada y traducida por Zaleta-Andrade para este trabajo de tesis.

La detección de fase se interpreta en cuadratura para preservar la información de la dirección del flujo (McLeod 1967), y por ende del movimiento, mediante la mezcla de las señales recibidas con dos señales de oscilador ($\varnothing 1$ y $\varnothing 2$), desfasadas 90 grados una respecto a la otra. Esto requiere de dos detectores de fase y de dos filtros. En el detector de cruce de ceros, la dirección del flujo se retiene teniendo en cuenta la salida de ambos canales de procesamiento.

En sistemas Doppler PW con procesamiento de datos en serie, las muestras de fase se toman desde un número adyacente de volúmenes de muestras y son procesadas a través de un solo circuito. Esto solo es posible si la tasa de rendimiento del sistema de procesamiento excede la frecuencia de muestreo de fase (Hoeks *et al.* 1981).

Asegurar el funcionamiento adecuado del detector del cruce por cero para evitar la atenuación de las señales

Para facilitar el correcto funcionamiento del detector de cruce por cero, la relación señal a ruido en su entrada no debe superar los 20 dB (*ibid*). Sin embargo, a la salida del detector de fase, las señales Doppler pueden estar oscurecidas por señales de baja frecuencia que pueden superar las señales Doppler verdaderas. Por ejemplo, una señal de ultrasonido retrodispersada por partículas de sangre en movimiento se ve oscurecida por señales que se originan en las interfaces de los tejidos. Estas últimas señales pueden superar las señales Doppler de partículas de sangre, y de otros objetivos de interés, en unos 30dB. Sin embargo, este número puede disminuir si se aumenta la frecuencia de emisión por las propiedades de dispersión dependientes de la frecuencia de las partículas de sangre. Esto ocasiona que el sistema Doppler PW sea más sencillo de operar, ya que no requiere de un ajuste para la atenuación individualmente variable de la señal de ultrasonido.

Consideraciones a la salida del detector de fase para asegurar el funcionamiento adecuado del sistema PW

Los niveles de las señales de ultrasonido varían a lo largo del ciclo cardiaco. Para acomodar las variaciones de persona a persona, así como el nivel de señal cambiante durante el ciclo cardiaco, se requieren 20 dB adicionales. Por lo tanto, si se desea facilitar el correcto funcionamiento del detector de cruce por cero, en el caso que la señal de interés sea la retrodispersada por el movimiento de sangre y se quiera garantizar que el sistema Doppler PW funcione adecuadamente, el detector de fase necesita a la salida un rango dinámico de 70dB. La velocidad de procesamiento y el rango dinámico de la salida de los detectores de fase establecen los requisitos básicos para un sistema Doppler PW multibarrera con procesamiento de datos en serie (*ibid*).

Descripción técnica de un sistema Doppler PW multibarrera con procesamiento de datos seriales

Si bien el procesamiento en serie de datos de señales de ultrasonido Doppler se puede llevar a cabo con señales en forma analógica, se prefiere que el alto rango dinámico de estas señales se procese de manera digital.

La frecuencia de emisión y el PRF son características principales de los sistemas que dependen de las velocidades máximas esperadas en la región de interés y la profundidad a lo largo del haz ultrasónico (Peronneau 1974), donde las velocidades deben ser detectadas. La frecuencia de emisión y el PRF demandan características especiales debido a la relación señal a ruido requerida de las señales Doppler detectadas y a su resolución. Por ejemplo, las aplicaciones del sistema Doppler PW se pueden restringir a solamente detectar mediciones de velocidades de flujo superficial si se escoge una frecuencia de emisión de 6 MHz (Hoeks 1981).

Para extender la profundidad a la que se pueden detectar las velocidades, se tiene que retrasar el momento en que se toma la primera muestra de fase respecto al momento de emisión. Es importante considerar una relación señal a ruido lo suficientemente alta para activar el detector de cruce por cero adecuadamente. Sin embargo, este incremento en la relación señal a ruido es a expensas de la resolución axial (*ibid*). La figura 2.20 y

2.21 muestran dos diagramas a bloques simplificados: un sistema Doppler PW de multibarrera con procesamiento de datos en serie y su digitalizador (ADC).

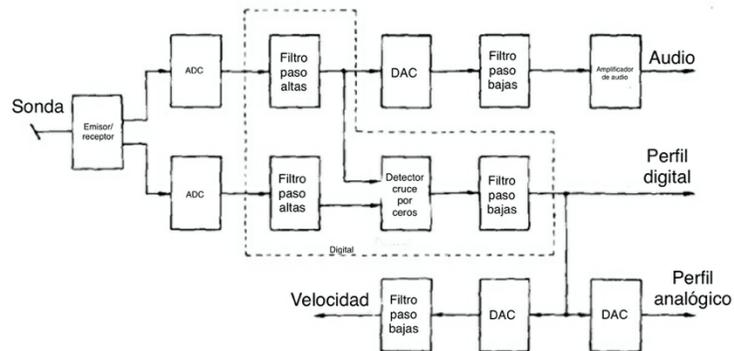


Figura 2.20: Diagrama a bloques de un sistema Doppler PW multibarrera con procesamiento de datos en serie. Figura obtenida de Hoeks *et al.* 1981. Adaptada y traducida por Zaleta-Andrade para este trabajo de tesis.

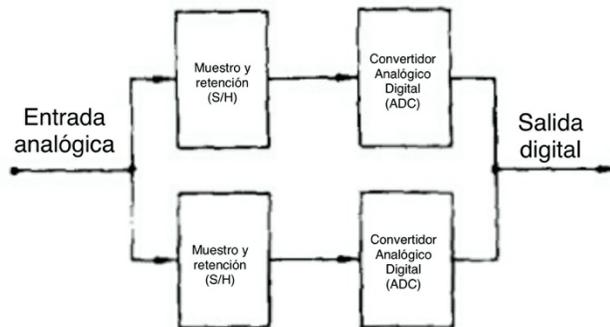


Figura 2.21: Diagrama a bloques de un digitalizador o ADC. Figura obtenida de Hoeks, Reneman, y Peronneau 1981. Adaptada y traducida por Zaleta-Andrade para este trabajo de tesis.

Capítulo 3

Intensidad del Eco de una Sonda de Ultrasonido Doppler Transcraneal

El análisis ultrasónico transcraneal no invasivo requiere de equipos TCD con tecnología de vanguardia que mejoren la precisión de las mediciones hechas por la sonda transcraneal y que tengan una interfaz de gestión que permita administrar los parámetros del equipo TCD y los datos de campo capturados por la sonda. La empresa Nihon Kohden y la Universidad de Leicester han desarrollado un prototipo TCD denominado BrainTV capaz de detectar señales de ultrasonido Doppler transcraneales y sincronizarlas con otras entradas fisiológicas. De esta manera buscar correlaciones entre el movimiento cerebral y otros sistemas fisiológicos, *e.g.*, los latidos del corazón.

El cálculo de la intensidad de las señales Doppler es un cálculo que no ha sido implementado en el equipo TCD de NK o Spencer, ni en la GUI de Matlab®, y que permite detectar el tipo de objetivo que ocasiona la señal Doppler – *e.g.*, un glóbulo rojo o tejido blando. Dicha intensidad es relativa entre una señal Doppler de referencia, a una profundidad específica, y su señal Doppler precedente. En este capítulo se muestra un algoritmo que realiza el cálculo de la intensidad de las Señales Doppler en Matlab®.

3.1 Dispositivos de Ultrasonido Transcraneales No Invasivos

Sistema Spencer

El dispositivo Doppler transcraneal Spencer implementa dos tecnologías patentadas por Tecnologías Spencer. La primera se llama sobremuestreo de alta definición y la segunda ST3 Doppler Transcraneal de modo M de potencia (Spencer Technologies 2016). A partir de ahora estas dos tecnologías serán llamadas HDO y PMM, respectivamente. Estas dos tecnologías se explican a detalle en el capítulo 4 “Temas Selectos de Ultrasonido Doppler”.

Tecnologías Spencer menciona que los exámenes neurovasculares y de monitoreo embólico con ultrasonido transcraneal se pueden realizar fácilmente con su tecnología de vanguardia. También, destaca que la tecnología PMM es un método extremadamente preciso para la medición de las velocidades del flujo sanguíneo cerebral. Esta es la motivación por la que este dispositivo es altamente relevante para este trabajo de tesis. Las capacidades del dispositivo ST3 Doppler Transcraneal de modo M de potencia se enlistan a continuación:

- Procesamiento digital HDO un modo-M de alta fidelidad;
- Monitoreo embólico altamente sensitivo mediante el patentado PMM;
- Mediciones de alta velocidad mediante alta frecuencia de repetición de pulso (PRF);
- Múltiples formatos de reporte; y
- Configuraciones del sistema personalizadas.



Figura 3.1: ST3 Sistema de Ultrasonido Transcraneal de Tecnologías Spencer. Figura obtenida de Spencer Technologies 2016.

Posición de la sonda en el sistema Spencer

El hueso temporal es la porción más delgada del cráneo humano y está localizado inmediatamente arriba del tragus. En esta posición las arterias intracraneales son transmitidas con una sonda TCD de 2 MHz para producir las formas de onda espectrales.

Las distancias se miden desde el punto de referencia “profundidad cero”: superficie de la piel donde la sonda hace contacto con la cabeza. Para la mayoría de adultos la línea media se encuentra entre 70 y 80 mm aproximadamente (Hakimi, Alexandrov, y Garami 2020).

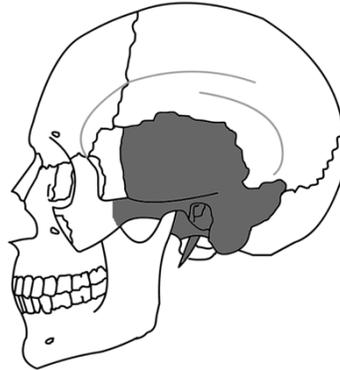


Figura 3.2: Hueso temporal.



Figura 3.3: Línea temporal (línea punteada); tragus (círculo). Figura obtenida de Wikimedia Commons Contributors 2010. Adaptada por Zaleta-Andrade para este trabajo de tesis.

Sistema Nihon Kohden

Nihon Kohden es uno de los principales desarrolladores y fabricantes de sistemas de medición fisiológica-médica en todo el mundo (Chung 2019). Su filosofía se concentra en contribuir al combate de las enfermedades y mejorar la asistencia sanitaria con ayuda de tecnología avanzada (Nihon Kohden 2019). La visión de esta compañía es la creación y evaluación de la tecnología emergente de neuromonitorización y monitoreo fisiológico para la atención aguda hospitalaria.

El velocímetro de tejido cerebral – comúnmente llamado BrainTV o Sistema NK – es un prototipo desarrollado en asociación con la Universidad de Leicester y el Servicio Nacional de Salud del Reino Unido (NHS por sus siglas en idioma inglés) (Chung 2018). Por consecuencia, la información de las tecnologías utilizadas en el dispositivo BrainTV no es de dominio público. Sin embargo, si es conocido que el dispositivo implementa la tecnología TCD de sistemas Spencer.

El dispositivo BrainTV es una unidad mejorada más ligera, portable, y con conexiones USB 2.0. Cuenta con dos canales de transmisión de datos utilizados junto a dos sondas transcraneales de 2 MHz, que se puede usar en conjunto o solamente una de ellas. Las sondas transcraneales tienen una frecuencia de repetición de pulso de 8 KHz (modificables). Este dispositivo sincroniza las señales de ultrasonido Doppler a un electrocardiograma (ECG) y otras entradas fisiológicas como la presión sanguínea (BP) o la presión intracraneal (ICP). Nihon Kohden ha dotado al sistema NK con una interfaz de adquisición de datos capaz de hacer una grabación continua (figura 3.4), sin mayor límite que el del tamaño de la unidad de memoria donde se guarda la grabación.

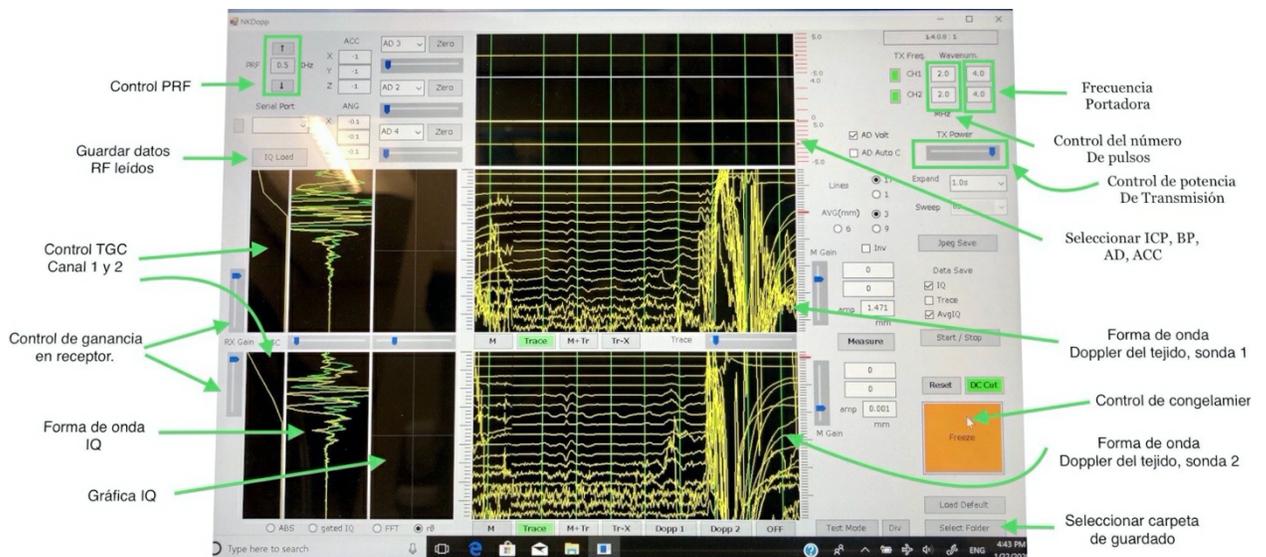


Figura 3.4: Interfaz de adquisición de datos del dispositivo BrainTV. Figura obtenida de Chung, Banahan, Minhas, y Lecchini-Visintini 2018. Adaptada y traducida por Zaleta-Andrade para este trabajo de tesis.



Figura 3.5: Equipo de integración (superior izquierda), posición de la sonda (superior derecha), y puertos (inferior derecha) del equipo BrainTV. Figura obtenida de Chung *et al.* 2018. Adaptada por Zaleta-Andrade para este trabajo de tesis.

3.2 Interfaz gráfica de usuario (GUI) en Matlab® para el cálculo de la intensidad de la sonda de ultrasonido transcraneal de equipos de Tecnologías Spencer y Nihon Kohden

Propósito de la interfaz gráfica de usuario

Todos los códigos incorporados en esta GUI de Matlab® trabajan en conjunto para calcular la intensidad de las señales de ultrasonido Doppler transcraneal, leyendo archivos con extensiones *.mat, *.iq1, y *.iq2 para mostrar la intensidad de salida de cada señal RMS de cada barrera con base en la siguiente fórmula de Evans *et al.* 2000:

$$\text{Intensidad de salida en decibeles} = 10 \cdot 10 \log_{10} \frac{I}{I_0} \quad \text{Ec.(3.1)}$$

I_0 = es la intensidad del rayo ultrasónico
 I = es la intensidad de una señal RMS Doppler

Tanto I_0 como I son detectadas por el mismo transductor. El resultado final del cálculo de las intensidades de las señales Doppler transcraneales se muestra en la figura 3.6

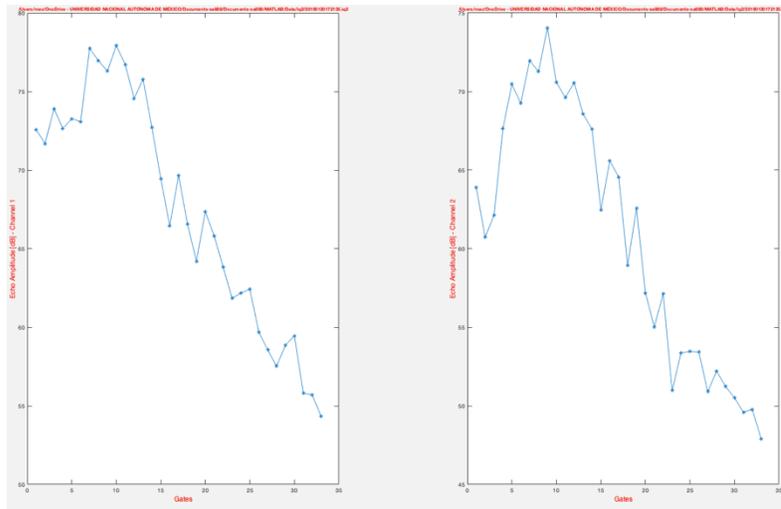


Figura 3.6: Salida de la GUI: intensidad de los valores RMS de las señales de cada una de las barreras (gates) en dB, utilizando un archivo *.iq2 (dos canales).

Estructura del código de la GUI

La GUI incorpora cuatro funciones creadas exclusivamente para llevar a cabo tareas específicas como leer archivos, disminuir la resolución de las señales y normalizar los datos de campo en archivos específicos, etc. Cada función proporciona resultados que siguen el flujo que se muestra en la figura 3.7.

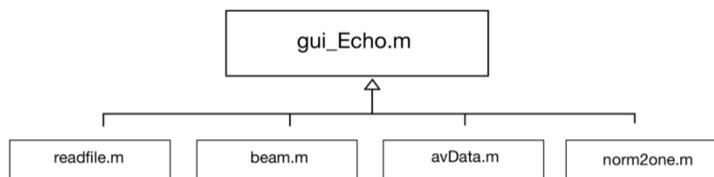
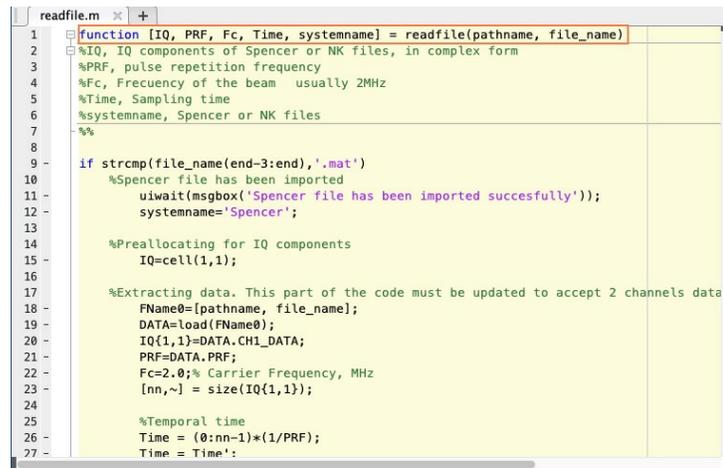


Figura 3.7: Flujo de las slidas de cada función incorporada en el código de la GUI.

Descripción de los códigos incorporados en la GUI

readfile.m

Esta función necesita, como entrada, el nombre de ruta y el nombre de archivo de los archivos *.iq1, *.iq2, o *.mat. *readfile.m* devuelve la siguiente información de las señales de ultrasonido Doppler: las componentes IQ de las señales, su PRF, la frecuencia de la señal de ultrasonido generada por el transductor (generalmente de 2 MHz) y un identificador.



```
1 function [IQ, PRF, Fc, Time, systemname] = readfile(pathname, file_name)
2 %IQ, IQ components of Spencer or NK files, in complex form
3 %PRF, pulse repetition frequency
4 %Fc, Frequency of the beam usually 2MHz
5 %Time, Sampling time
6 %systemname, Spencer or NK files
7 %%
8
9 if strcmp(file_name(end-3:end),'.mat')
10 %Spencer file has been imported
11 uiwait(msgbox('Spencer file has been imported succesfully'));
12 systemname='Spencer';
13
14 %Preallocating for IQ components
15 IQ=cell(1,1);
16
17 %Extracting data. This part of the code must be updated to accept 2 channels data
18 FName0=[pathname, file_name];
19 DATA=load(FName0);
20 IQ{1,1}=DATA.CH1_DATA;
21 PRF=DATA.PRF;
22 Fc=2.0;% Carrier Frequency, MHz
23 [nn,~] = size(IQ{1,1});
24
25 %Temporal time
26 Time = (0:nn-1)*(1/PRF);
27 Time = Time';
```

Figura 3.8: Parte del código de lectura de archivos *.iq1, *.iq2, y *.mat llamado readfile.m. Figura obtenida del código original gui_Echo.m.

avData.m

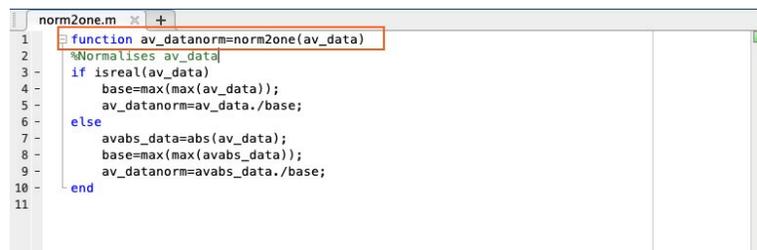
El código BrainTV disminuye la resolución de las señales de entrada en algún punto dentro de sus códigos. El mismo proceso se implementa en la función *avData.m*, donde las componentes IQ de las señales Doppler son reducidas en tiempo. Este proceso se explica a detalle en el sub-capítulo “Compresión de las Señales de Ultrasonido Doppler Transcraneales” más adelante en este mismo capítulo.

La función *avData.m* necesita las componentes en fase y cuadratura de las señales de ultrasonido Doppler con sus respectivos tiempos de muestreo y el número de muestras para reducir la resolución de las señales de entrada. Esta función devuelve el valor absoluto (magnitud) de las componentes en fase y en cuadratura comprimidas,

junto a sus respectivos tiempos de muestreo. También devuelve el valor absoluto de las componentes en fase y en cuadratura no comprimidas.

norm2one.m

Esta función es muy simple y solo normaliza, a uno, todos los valores devueltos por la función avData.m.



```
norm2one.m x +
1 function av_datanorm=norm2one(av_data)
2 %Normalises av_data
3 if isreal(av_data)
4     base=max(max(av_data));
5     av_datanorm=av_data./base;
6 else
7     avabs_data=abs(av_data);
8     base=max(max(avabs_data));
9     av_datanorm=avabs_data./base;
10 end
11
```

Figura 3.9: Parte del código que normaliza, a uno, todos los valores devueltos por avData.m: el valor absoluto de las componentes en fase y en cuadratura comprimidas de las señales de ultrasonido Doppler. Figura obtenida del código original gui_Echo.m.

Descripción de la GUI

La interacción del usuario con la GUI es sencilla, ya que solo se necesita hacer clic en el botón “Import file” para introducir los archivos *.iq1, *.iq2, o el archivo *.mat, que contienen datos de campo de las señales de ultrasonido Doppler transcraneal. También se necesita añadir el archivo “peaks.mat” (explicado a detalle más adelante en este capítulo) que contiene la intensidad inicial para el cálculo de la intensidad de las señales Doppler. La figura 3.10 muestra las diferentes partes de la interfaz gráfica de usuario y la figura 3.11 el resultado final de la GUI, es decir, el cálculo de la intensidad de las señales de ultrasonido Doppler.

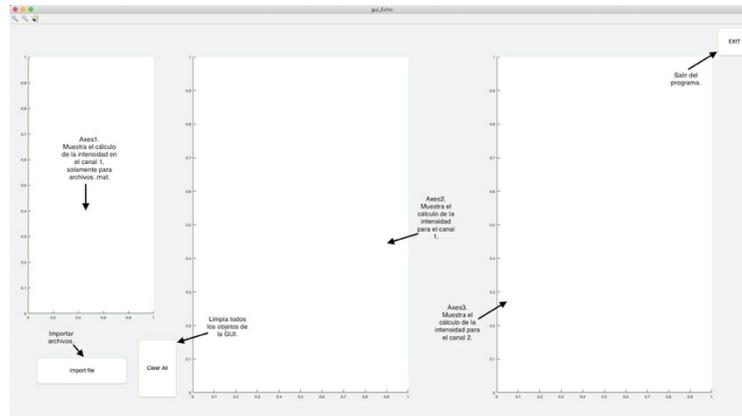


Figura 3.10: Matlab® GUI que muestra la intensidad de señales de ultrasonido Doppler transcraneal de archivos *.mat, *.iq1, y *.iq2.

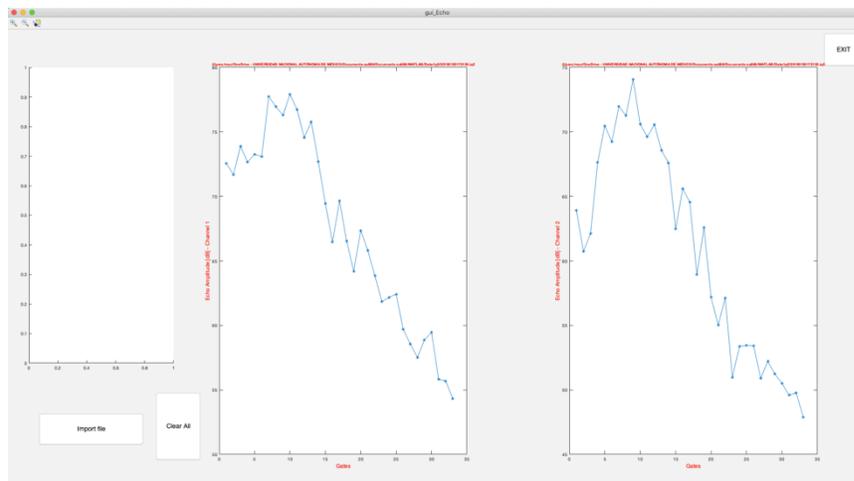


Figura 3.11: Cálculo de la intensidad de las señales de ultrasonido Doppler de 33 profundidades distintas, utilizando la interfaz gráfica de Matlab®.

3.3 Lectura de archivos de las señales de ultrasonido Doppler de los equipos de Tecnologías Spencer y Nihon Kohden

Tanto el sistema Spencer como NK guardan todas las lecturas de los datos de campo dentro de tipos de archivos especiales. Los archivos del sistema NK están en formato binario y contienen un mayor número de datos distintos en comparación con los archivos del sistema Spencer. Dicha información se encuentra en los encabezados o “headers” de los archivos. Esto hace que sea necesario una refactorización de datos mucho más compleja que la de los archivos Spencer, los cuales, son capaces de ser leídos en Matlab® como un dato de tipo estructura (struct). El proceso de lectura de

archivos está a cargo de la función `pb_importfile_Callback(...)` en la GUI, principalmente mediante la implementación de `readfile.m`. Parte de la lectura consta en extraer los datos de campo de las señales codificadas en fase y cuadratura (Componentes IQ).

Archivos `.mat` del sistema Spencer

Los archivos `.mat` del sistema Spencer muestran un nombre alfanumérico que incluye fecha y datos relevantes del estudio transcraneal. Esto es muy importante porque permite a los programadores conocer información relevante del archivo incluso sin necesidad de abrirlo. E.g., la extensión `*.iq1` quiere decir que el archivo contiene información de un solo canal, lo que quiere decir que la recopilación de datos se hizo mediante una sola sonda en lugar de dos. La estructura interna de los archivos `*.mat` muestra sus datos como en la figura 3.12.

```
Recording: '20171116100351.iq1'
System: 'NKL'
Study: ''
ProbePos: []
Subject: []
Sex: ''
Age: []
Posture: []
RawVelocity: {[89998*34 double]}
Time: [89998*1 double]
RR: [223*1 double]
BadRR: []
Bp: {[[] [] [] [] []] [90000*1 double]}
Co2: [90000*1 double]
Gyr: []
Acc: []
Flags: []
Notes: ''
BPRecorded: [120 80]
time_IQavg: [90000*1 double]
time_IQ: [90000*1 double]
HT: [90000*1 double]
```

Figura 3.12: Estructura interna de un archivo `*.mat` del sistema Spencer. Figura obtenida de un archivo de la carpeta privada de la investigación de la Universidad de Leicester.

La figura 3.12 muestra datos de interés para quien monitorea el TCD (equipo de ultrasonido), entre ellos destacan algunos tiempos de muestreo y la velocidad del movimiento del tejido cerebral (`time_IQavg`, `time_IQ`, y `RawVelocity`, respectivamente). Los algoritmos que extraen y refactorizan estos datos se explican más adelante en este capítulo.

Archivos `.iq` del Sistema NK

A diferencia de un archivo del sistema Spencer, tal como se mencionó anteriormente, los archivos.iq del sistema NK están en formato binario. Contienen un nombre de archivo que muestra directamente información del estudio. La extensión *.iq también es importante porque dice si el archivo contiene datos de dos canales o solo uno. Por ejemplo, *P0000410S4120.iq2* contiene información de dos canales o *P0000410S4120.iq1* que contiene información solo de un canal. La cabecera contiene un arreglo de 8 datos, de los cuales solo 2 son relevantes para el cálculo de la intensidad del ultrasonido retrodispersado. La posición 4 del arreglo contiene la frecuencia de la señal portadora (en MHz) y la posición 8 la frecuencia de repetición de pulso (en kHz). De manera similar, en el sistema Spencer las señales de ultrasonido están codificadas en componentes en cuadratura y fase mediante números complejos. Los algoritmos que extraen y refactorizan estos datos se explican más adelante. Un ejemplo de encabezado de un archivo *.iq1 y una parte de los datos de campo de este archivo se muestran en la figura 3.13 y 3.14, respectivamente.

Encabezado

header	
	1
1	3.2709e+21
2	7.5110e-43
3	48000000
4	2
5	4
6	2
7	4
8	0.5000

Frecuencia portadora (MHz)

PRF (kHz)

Figura 3.13: Encabezado de un archivo *.iq1. Figura obtenida de un archivo contenido en la carpeta privada de la investigación de la Universidad de Leicester.

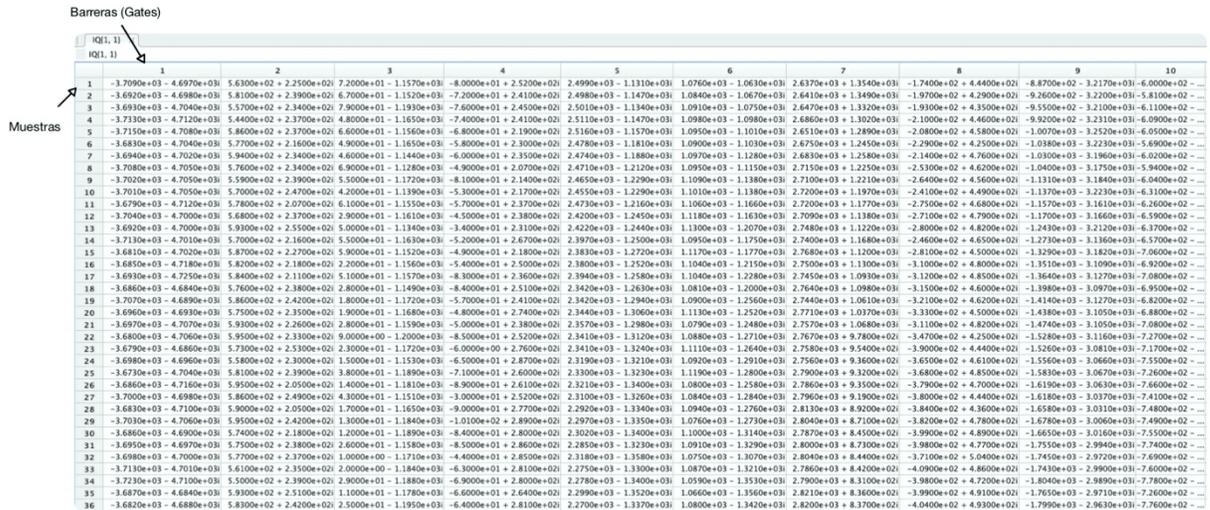


Figura 3.14: Componentes en fase y cuadratura de un archivo *.iq1 de señales de ultrasonido Doppler producidas por la insonación de un rayo ultrasónico transcraneal utilizando el sistema NK. Figura obtenida de un archivo contenido en la carpeta privada de la investigación de la Universidad de Leicester.

Algoritmo de extracción de datos de archivos del sistema Spencer y NK en la GUI

Matlab® permite la lectura y reescritura de datos provenientes de otras aplicaciones. Por ejemplo, los datos extraídos por el sistema Spencer o NK pueden someterse a procesos de refactorización de datos que no son posibles de realizar directamente en los equipos biomédicos. Es decir, para que los procesos de compresión de señales fueran posibles, los fabricantes como Nihon Kohden tendrían que desarrollar soluciones muy específicas para cada usuario.

Afortunadamente, parte del código modular de extracción del eco vuelve el proceso de extracción de datos algo muy sencillo. Basta con insertar el comando `“uigetfile(...)”` para tener la información de guardado del archivo y posteriormente leerlo utilizando `“readfile(...)”`, tal como se muestra en la figura 3.15.

```

81 --- *.mat file open
82 [file_name, pathname, ~] = uigetfile({'*.iql;*.iq2;*.mat',...
83 'Models (*.iql;*.iq2) or MATLAB files (*.mat)'});
84
85 setappdata(handles.figure1,'fileName', file_name);
86 cd(CFd);
87
88 if isnumeric(pathname)
89     %Avoiding error due to cancellation
90     uiwait(msgbox('Import Data Action Cancelled'));
91 else
92     %setting app data
93     fullfile_name=fullfile(pathname,file_name);
94     setappdata(handles.figure1,'file_name',fullfile_name);
95
96     %extracting IQ data and other parameters
97     %IQdata, IQ components in a complex form
98     %PRF, Hz
99     %Fc, Mhz
100    %systemname, Spencer or NK
101    %Time, s
102    avsample=100; %For averaging
103
104    [IQdata, PRF, Fc, Time, systemname]=readfile(pathname, file_name);%PRF, KHz - Fc, Mhz
105    [IQdatabeam timebeam]=beam(fcbeam, time); %IQdatabeam is not a cell.
106    IQdata(1,1)=[IQdatabeam IQdata(1,1)]; %IQdatabeam is concatenated in the first collumn of IQdata
107    [nn,mm]=size(IQdata(1,1)); %mm is num of col -gates-, nn num of rows
108    %Checking whether IQdata has 1 or 2 channels

```

Figura 3.15: Comandos de obtención de información de guardado del archivo a utilizar para su lectura. Figura obtenida del código original *gui_Echo.m*.

La función *readfile(...)* retorna las componentes en fase y cuadratura de las 33 señales Doppler correspondientes a distintas profundidades cerebrales; la frecuencia de repetición de pulso; la frecuencia portadora; los tiempos de cada muestra; y un identificador del sistema Spencer o NK.

3.4 Aproximación inicial: señal de ultrasonido Doppler transcraneal generadora del eco

Parámetros del Sistema Spencer y NK

La intensidad del eco de una señal de ultrasonido la define Evans *et al.* (2000) como:

$$EI = 10 \cdot \log_{10} \left(\frac{I}{I_0} \right) \tag{Ec.(3.2)}$$

EI es la intensidad del eco [dB]
I es la intensidad de la señal recibida
I₀ es la intensidad de referencia

La intensidad o amplitud de la señal recibida proviene de la retrodispersión de la insonación transcraneal del ultrasonido Doppler. *I.e.*, son las señales muestreadas codificadas en componentes en fase y cuadratura. La intensidad de referencia es aquella

señal de insonación Doppler transcraneal (también llamada señal portadora), con una frecuencia de 2 MHz (Evans *et al.* 2000) y una amplitud desconocida – a menos que se utilice un instrumento de medición directo sobre la sonda que mida su potencia de emisión. En el apartado de conclusiones de este trabajo de tesis esto se discute a detalle.

El proceso de insonación en sistemas PW permite que la señal de ultrasonido TCD llegue a 33 profundidades distintas del cerebro (llamadas barreras o gates), donde la profundidad 33 es regularmente la más débil debido a la pérdida de potencia de la señal ocasionada por la distancia (Kucewicz 2004). El eco producido en la barrera más cercana al transductor, por lo tanto, es la señal más parecida a la señal portadora. La figura 3.16 muestra la forma de onda de una señal Doppler con un punto de amplitud máxima- I.e., un punto de mayor intensidad del eco.

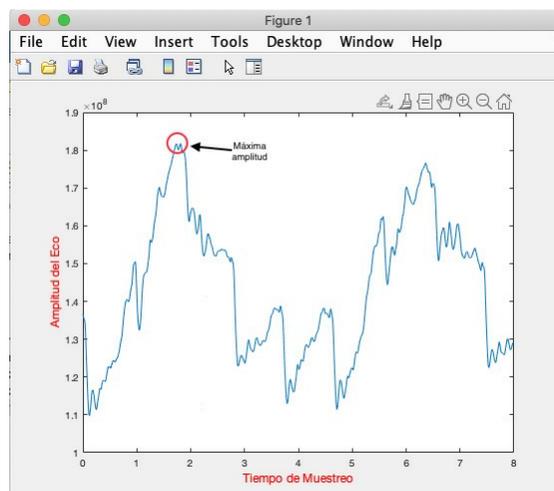


Figura 3.16: Amplitud máxima de la señal de ultrasonido retrodispersada (eco) de la barrera número 10 de un archivo NK de un canal. Las magnitudes del tiempo de muestreo y amplitud del eco son desconocidas para el autor de este trabajo de tesis. Figura obtenida utilizando Matlab®.

Existe la posibilidad que la intensidad de las señales no sea inversamente proporcional a la profundidad de la barrera, ya que una señal Doppler a una mayor profundidad puede ser más intensa que una señal Doppler a menor profundidad. La razón es que el eco de la señal menos profundo pudo haber sido reflejado por un vaso sanguíneo en vez de tejido blando. El coeficiente de atenuación de la sangre es de $0.2 \text{ dB} \cdot \text{cm}^{-1}$ y el del tejido blando de $0.7 \text{ dB} \cdot \text{cm}^{-1}$ (Evans *et al.* 2000). De hecho, detectar

estos fenómenos es una de las aplicaciones potenciales del cálculo de la intensidad del eco – objeto de estudio de este trabajo de tesis –, ya que permite decidir si una señal retrodispersada es producto del movimiento del tejido cerebral o del flujo de un vaso sanguíneo. Por lo anterior, y asumiendo que la señal con el punto de mayor amplitud es la más aproximada a la señal portadora (o de insonación), se puede lograr una aproximación si se encuentra el punto de mayor amplitud entre las 33 señales de eco. Esto es preferible después de aplicar la técnica de disminución de la tasa de muestreo (Tan y Jiang 2013).

La amplitud máxima que resulta del proceso de aproximación es diferente entre archivos del sistema NK y Spencer, así como entre archivos del mismo sistema, NK o Spencer. Las tablas 3.1 y 3.2 muestran las amplitudes máximas de archivos de estudios reales, propiedad de la Universidad de Leicester.

Datos del archivo	Máxima amplitud de la señal de ultrasonido Doppler
Estudio: Georgina	8.0147e03
Sistema: NK	
Número de estudio: P001	
Descripción del estudio: Leg_raise_F	
Tipo: .iq1	
Estudio: Georgina	8.5922e3
Sistema: NK	
Número de estudio: P005	
Descripción del estudio: Stand_T	
Tipo: .iq1	

Estudio: Georgina	7.3742e03
Sistema: NK	
Número de estudio: P010	
Descripción del estudio: Maths_T	
Tipo: .iq1	
Estudio: Georgina	2.44e4
Sistema: NK	
Número de estudio: P015	
Descripción del estudio: Leg_raise_T	
Tipo: .iq1	

Tabla 3.1: Máximas amplitudes de distintos archivos *.iq1 (NK), calculadas utilizando la función *peaksfinder.m*. Datos obtenidos de archivos contenidos en la carpeta privada de la investigación de la Universidad de Leicester.

Datos del archivo	Máxima amplitud de la señal de ultrasonido Doppler
Estudio: Poppy	1.2e9
Sistema: Spencer	
Número de estudio: 00392	
Tipo: .mat	
Estudio: Poppy	1.0372e9
Sistema: Spencer	
Número de estudio: 00398	
Tipo: .mat	

Estudio: Poppy	1.1023e9
Sistema: Spencer	
Número de estudio: 00408	
Tipo: .mat	
Estudio: Poppy	1.414e9
Sistema: Spencer	
Número de estudio: 00464	
Tipo: .mat	

Tabla 3.2: Máximas amplitudes de distintos archivos *.mat (Spencer), calculadas utilizando la función peaksfinder.m. Datos obtenidos de archivos contenidos en la carpeta privada de la investigación de la Universidad de Leicester.

Por los resultados que se muestran en las figuras 3.17 y 3.18, la aproximación es que depende de una base de datos de distintas amplitudes de distintos archivos del mismo sistema, provenientes de estudios con protocolos muy similares. Dicha base de datos muestra una desviación entre cada una de las amplitudes ordenadas de menor a mayor. El resultado promedio de estas desviaciones entre amplitudes propone ser la desviación aproximada entre el punto de mayor amplitud registrado y la señal portadora. Entonces, se dice que la intensidad de referencia (de la señal portadora) se puede definir de la siguiente manera:

$$I+ = A_{\max} + \Delta I \quad \text{Ec.(3.3)}$$

I es la intensidad de referencia
A_{max} es la amplitud máxima de varias amplitudes
ΔI es la desviación promedio entre varias amplitudes máximas

Algoritmo de aproximación de la señal de ultrasonido Doppler transcraneal generadora del eco en la GUI

El código *beam.m*, junto con el código *amplitude.m*, son funciones que reconstruyen la señal de ultrasonido Doppler RMS con mayor intensidad entre un grupo de señales del mismo tipo. Su objetivo es resolver el problema de no tener una referencia inicial I_0 para el cálculo de la intensidad de salida. El código *peaksfinder.m* puede calcular la amplitud máxima entre todas las señales de un archivo *.iq1, *.iq2, *.mat – que contienen las componentes IQ del haz de ultrasonido Doppler – y guardarla en un archivo llamado *peaks.mat*. *peaksfinder.m* puede repetir este proceso infinitamente para quizás detectar amplitudes más altas en otros archivos *.mat, *.iq1, o *.iq2 y conseguir una mejor aproximación. Cuanto mayor sea la amplitud encontrada, mejor será la aproximación debido a que la señal más intensa siempre será el haz de salida y a que no puede haber amplificación en ninguna de las señales Doppler. El pico más intenso encontrado por *peaksfinder.m* será la señal más aproximada a la señal de ultrasonido transcraneal insonada.

```
1 function peaksfinder
2 %The aim of this code is to generate a 'peaks.mat' file that will help to
3 %approximate the amplitude of a reference signal
4
5 %The interaction is with the Command Window.
6 % -0 for stop running
7 % -1 up to 9 to continue
8
9 %NOTES:
10 % - 'peaks.mat' is created automatically, even if it has not been created
11 % yet.
12 % -a the max peak of all the signals within the file will be saved into
13 % 'peaks.mat'
14
15 Cfd=pwd;
16 userin=1;
17 while userin==0
18     userin=input('Inset 0 to stop running or other number to continue: ');
19     if userin == 0
20         continue;
21     end
22 %Initialising
23     Cfd = pwd;
24     avsample=100;
25
26 %---- *.mat file open
27 uiwait(msgbox('Select a Spencer or NK file'));
28 [file_name, pathname, ~] = uigetfile( '*.iq1;*.iq2;*.mat',...
29     ' Models (*.iq1;*.iq2) or MATLAB files (*.mat)');
30
31 if isnumeric(pathname)
32     uiwait(msgbox('Import Data Action Cancelled'));
33     userin=0;
34 else
35     fullfile_name=fullfile(pathname, file_name);
36     [IQdata, ~, ~, Time, systemname]=readfile(pathname, file_name);
37     %Assesing whether IQdata has 1 or 2 channels
38     [numchannels, ~]=size(IQdata);
39     %Assesing whether IQdata is Spencer or NK
40     switch systemname
41         case 'Spencer'
42             if numchannels==1 %1 channel
43                 [avdata, ~, Time]=avData(IQdata(1,1), avsample, Time);
44                 if isreal(avdata)
45                     base=max(max(avdata));
46                 else
47                     avabs_data=abs(avdata);
48                     base=max(max(avabs_data));
```

Figura 3.17: Parte del código de la función *peaksfinder.m*. Figura obtenida del código original *gui_Echo.m*.

```

1 function [reference, deltagv]=amplitude(pathname, filename)
2 %The aim of this code is to approximate the amplitude of the reference
3 %signal used to plot the ultrasound echo.
4 %It uses the peaks calculated using 'peaksfinder.m'
5
6 %reference = Max peak of a *.mat file database
7 %deltavg = the average of the deviation between all sorted peaks in
8 %ascending order
9
10 FName=[pathname filename];
11 if isnumeric(pathname)
12     uiwait(msgbox('Import Data Action Cancelled'));
13 else
14     DATA=load(FName);
15     peaks=DATA.peaks;
16     sortpeaks=sort(peaks);%<-- Maybe move this line
17     maxpeak=max(peaks); %Calculating maximum peak of all data
18     peaksl=length(sortpeaks);
19     norepeaks=0;
20
21     %Avoiding repetition in values
22     for i=1:1:peaksl-1
23         if sortpeaks(i+1)~=sortpeaks(i) && norepeaks(end)~=sortpeaks(i)
24             norepeaks=[norepeaks sortpeaks(i) sortpeaks(i+1)]; %#ok<AGROW>
25         end
26     end
27     norepeaks=norepeaks(2:end); %Getting rid of the 1st 0
28     peaksl=length(norepeaks);% New length with no repetead elem
29     %Preallocating
30     delta=zeros(1,peaksl-1);
31
32     %Calculating de deviation of each sorted peak
33     for i=1:1:peaksl-1
34         delta(i)=abs(norepeaks(i+1)-norepeaks(i));
35     end
36     deltagv=mean(delta); %Average deviation
37
38     reference = maxpeak+deltavg;
39 end

```

Figura 3.18: Código de la función *amplitude.m*. Figura obtenida del código original *gui_Echo.m*.

```

1 function [beam, Time]=beam(Fc, Time)
2 %The aim of this code is to generate the amplitudes for the signal of the
3 %ultrasound beam.
4
5 %NOTES:
6 %1/PRF is used as Ts. I haven't understood why, Fc=2Mhz for Dopp Ultrasound
7 %Fc, Hz
8 %
9 %Instructions for the user
10 uiwait(msgbox('select peaks.mat file'));
11 %Opening *.mat file
12 [filename, pathname]=uigetfile({'*.mat', 'MATLAB files(*.mat)'});
13
14 %Calculating Amplitude reference for creating the beam
15 [reference, ~]=amplitude(pathname, filename);
16 %reference=0.000001; %Uncomment this line if the initial reference I0 wants
17 %to be personalised.
18
19 time=length(Time);
20
21 %Preallocating
22 beam=zeros(time,1);
23
24 for i=1:1:time
25     beam(i,1)=reference*sin(2*pi*Fc*Time(i)); %
26 end
27
28 beam=complex(beam); %If complex values are not required, eliminate this line.
29

```

Figura 3.19: Código de la función *beam.m*. Figura obtenida del código original *gui_Echo.m*.

El código *peaksfinder.m* puede ser ejecutado de forma independiente, pues no está integrado en la GUI. Este código no necesita ninguna variable de entrada, sino un archivo *.iq1, *.iq2, o *.mat que contenga las componentes IQ de las señales de ultrasonido Doppler. El código *peaksfinder.m* crea – si no existe ya – un archivo llamado *peaks.mat*

en la misma ruta que *peaksfinder.m*. Las amplitudes máximas de las señales de ultrasonido Doppler procesadas se irán guardando en *peaks.mat* en forma de una pila horizontal. El código *peaksfinder.m* puede ejecutarse de forma infinita para seguir insertando datos en *peaks.mat* y así lograr que la referencia de intensidad inicial se aproxime más al rayo de ultrasonido transcraneal insonado.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	9.2029e...	9.2029e...	9.1123e...	7.8366e...	7.8366e...	8.0147e...	6.0271e...	4.9232e...	8.2352e...	8.2352e...	8.0137e...	5.5934e...	5.5934e...	4.7453e...
2														

Figura 3.20: archivo *peaks.mat* construido con bases de datos de estudios Doppler transcraneales privados de 2019, propiedad de la Universidad de Leicester. Figura obtenida utilizando Matlab®.

El código *amplitude.m* necesita el nombre y ruta del archivo *peaks.mat* para retornar el pico o punto de mayor amplitud, así como la desviación promedio entre los picos. De estas dos variables retornadas, solo el pico – la variable *reference* dentro del código *amplitude.m* – es utilizado en *beam.m* y la desviación promedio entre los picos puede ser utilizada en mejoras posteriores del método de aproximación planteado en este trabajo de tesis o en otras aplicaciones.

Para retornar las magnitudes y tiempos de muestreo de la aproximación del rayo ultrasónico de salida, *beam.m* necesita de su frecuencia portadora así como de los tiempos de muestreo de las señales de ultrasonido Doppler. El código *beam.m* retorna una señal de salida aproximada con el mismo número de muestras que las señales de ultrasonido Doppler, codificadas en fase y cuadratura.

3.5 Compresión de las señales de ultrasonido Doppler transcraneales

Sea F_m la frecuencia cíclica del componente de frecuencia más alta de una señal continua $x(t)$, el teorema de muestreo (Nyquist) establece que $x(t)$ puede reconstruirse exactamente a partir de una versión muestreada $x(n)$, siempre que la frecuencia de muestreo $F_s=1/T_s$ sea mayor que el doble de F_m . Debido a las características no lineales de los dispositivos físicos, en la práctica se suelen utilizar frecuencias de muestreo algo mayores que la mínima teórica utilizada (Sundararajan 2016). En el caso de las señales de ultrasonido Doppler, estas se encuentran en el espectro electromagnético de radiofrecuencia y, por lo tanto, la frecuencia de muestreo de estas señales $F_s=1/T_s$ debe ser $4F_m$ (Hoeks, Brands, Arts, y Reneman 1994).

De acuerdo con la información del software BrainTV, se supone que el factor de escala es 50 para la compresión del tiempo – o disminución de la resolución de las señales de ultrasonido Doppler. Si esta señal de ultrasonido se denotara como:

$$x(n): x(0), x(1), x(2), x(3), \dots$$

Entonces, una señal de ultrasonido de 300 muestras puede verse como en la figura 3.21:

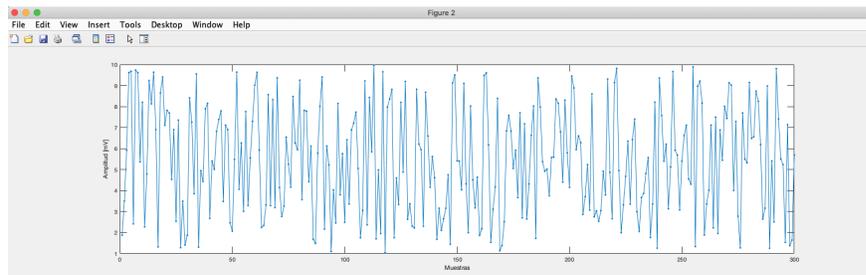


Figura 3.21: Señal creada aleatoriamente en Matlab® para representar una señal de ultrasonido Doppler con 300 muestras.

Para reducir la resolución de la señal $x(n)$, se sustituye la variable n por $50n$. Las muestras de la señal comprimida de $x(n)$ se denotan entonces como:

$$x(50n): x(0), x(50), x(100), x(150) \dots$$

La señal comprimida por un factor de 50 de la señal de ultrasonido Doppler con 300 muestras se muestra en la figura 3.22.

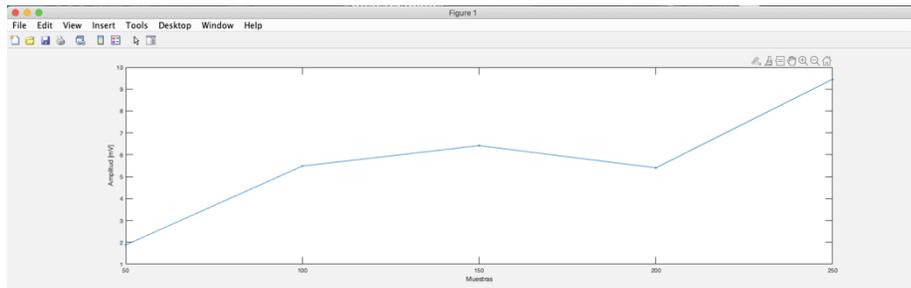


Figura 3.22: Señal comprimida por un factor de 50 de la señal de ultrasonido Doppler de 300 muestras (creada aleatoriamente con Matlab®).

La señal de ultrasonido Doppler comprimida, $x(50n)$, ahora está compuesta por las muestras múltiplos de 50 de $x(n)$. Las muestras que no son múltiplos de 50 se pierden en el proceso de compresión. La figura 3.23 muestra la representación simbólica de comprimir una señal de ultrasonido Doppler, llevado a cabo por la función `avData.m` en la GUI para este trabajo de tesis.

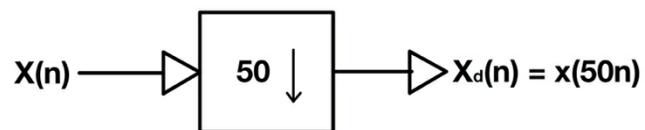


Figura 3.23: Representación simbólica de comprimir una señal por un factor de 50, haciendo referencia al proceso de reducción de la resolución de las señales de ultrasonido Doppler en la función `avData.m` en la GUI.

Algoritmo de compresión de señales de ultrasonido Doppler usando la GUI

El algoritmo de disminución de la resolución es sumamente sencillo. Se necesitan los datos de campo y sus respectivos tiempos de muestreo, así como el factor de disminución de resolución. La función `avData.m` toma las muestras de las señales de ultrasonido Doppler moduladas en fase y cuadratura y calcula su magnitud a la vez que también comprime la señal Doppler. El código `avData.m` retorna las magnitudes de las muestras de las señales de ultrasonido Doppler comprimidas por un factor dado por la variable `Av_sample`, sus respectivos tiempos de muestreo, y las magnitudes de las señales de ultrasonido Doppler sin comprimir.

```
1 function [av_data, IQdata, t] = avData(IQDATA, Av_sample, Time)
2 %Average IQDATA over a number given by Av_sample
3 %Returns The abs value of complex data
4 % The IQDATA but averaged
5 % Sampling time as a vector
6
7 %Number of rows; samples.
8 [row,col]=size(IQDATA);
9 x=1:Av_sample:row;
10
11 %Preallocating size for av_data
12 av_data=zeros(length(x)-1,col);
13 IQdata=zeros(length(x)-1,col);
14 t=zeros(length(x)-1,1);
15 for i = 1:col
16     for k=length(x)-1
17         av_data(k,i) = sqrt(real(IQDATA(x(k),i))^2 + imag(IQDATA(x(k),i))^2);
18         IQdata(k,i)=IQDATA(x(k),i);
19         if i==1
20             t(k,i)=Time(k,1)*100;
21         end
22     end
23 end
24 end
25
26
```

Figura 3.24: Parte del código de disminución de la resolución de las señales de ultrasonido Doppler transcraneales `readfile.m`. Figura obtenida del código original `gui_Echo.m`.

Capítulo 4

Temas Selectos de Procesamiento de Señales de Ultrasonido Doppler Transcraneal

Es importante motivar al lector de este trabajo de tesis con temas que revelen información detallada y específica de los sistemas TCD y la GUI *BrainTV*. Tecnologías Spencer y su tecnología HDO, mediante cálculos más precisos, disminuyen la potencia del ruido en banda de las señales de ultrasonido sin afectar la potencia de la señal. La tecnología PMM (de Tecnologías Spencer) utiliza sonógrafos capaces de identificar no solo las formas de ondas espectrales de las señales Doppler, sino otros parámetros como la profundidad de los objetivos – o barreras. La GUI *BrainTV* (programa de uso privado de la Universidad de Leicester) utiliza los datos de campo de los sistemas TCD de Nihon Kohden y Tecnologías Spencer para extraer la mayor cantidad de datos tanto de los objetivos perturbados ultrasónicamente, como de las señales Doppler resultantes de la perturbación de estos objetivos.

4.1 Sobremuestreo de Alta Definición – HDO

Muchas aplicaciones requieren tomar mediciones usando un convertidor analógico-digital (ADC). Dichas aplicaciones tienen requerimientos basados en el rango dinámico de la señal, el cambio más pequeño de los parámetros a medir, y la relación señal a ruido (SNR). Por esta razón, para mejorar el SNR e incrementar la resolución efectiva de las mediciones, muchos sistemas utilizan la técnica de promediado de señales y sobremuestreo en sistemas ADC (Sillicon Labs 2013).

De acuerdo con el teorema de Nyquist, una frecuencia de muestreo F_s , permite que una señal de interés se reconstruya a $\frac{1}{2}$ de la frecuencia de muestreo. Esto quiere decir que si la frecuencia de muestreo es 1000 kHz, las señales que estén por debajo de los 500 kHz pueden ser reconstruidas y analizadas correctamente (*ibid*). Sin embargo, junto con la señal de entrada hay una señal de ruido – denominada *ruido blanco* – que causa un empalme en las frecuencias menores a $\frac{1}{2} F_s$.

$$E(f) = e_{rms} \cdot \left(\frac{2}{F_s}\right)^{\frac{1}{2}} \quad \text{Ec.(4.1)}$$

e_{rms} es la potencia de ruido promedio
F_s es la frecuencia de muestreo
E(f) es la densidad espectral de energía en banda (ESD)

E(f) muestra que ESD disminuye a medida que la frecuencia de muestreo aumenta (Candy y Temes 1987).

Si se tiene una potencia de ruido fija, generada por el ruido de cuantización, es posible calcular la cantidad de sobremuestreo necesaria para aumentar la resolución efectiva de los datos. Para esto, vale la pena mencionar que el ruido de cuantización es la distancia adyacente entre códigos ADC o niveles de cuantización.

$$\Delta = \frac{V_{ref}}{2^N} \quad \text{Ec.(4.2)}$$

N es el número de bits en el código ADC
V_{ref} es el voltaje de referencia
Δ es el ruido de cuantización

El ruido de cuantización cumple la siguiente condición:

$$e_q \leq \frac{\Delta}{2} \quad \text{Ec.(4.3)}$$

e_q = ruido de cuantización

Si se asume que el ruido se aproxima al ruido blanco, la variable aleatoria representando al ruido tiene una distribución uniforme con una media de cero entre los códigos ADC (Candy *et al.* 1987).

$$\int_{-\frac{\Delta}{2}}^{\frac{\Delta}{2}} \left(\frac{e_q^2}{\Delta}\right) de = \frac{\Delta^2}{12} \quad \text{Ec.(4.4)}$$

La relación entre la frecuencia de muestreo y la frecuencia de Nyquist se denomina *relación de sobremuestreo (OSR)*.

$$OSR = \frac{F_s}{2 \cdot F_m} \quad \text{Ec.(4.5)}$$

F_s es la frecuencia de muestreo

F_m es la cmponente de frecuencia mayor de la señal de entrada

Debido a que se sigue asumiendo que el ruido es *ruido blanco*, se puede hacer que la potencia de ruido en banda sea una función de OSR.

$$n_0 = \frac{e_{rms}^2}{OSR} \quad \text{Ec.(4.6)}$$

n_0 es la potencia del ruido de salida

Por lo tanto, la n_0 muestra que es posible disminuir la potencia del ruido en banda simplemente al aumentar el OSR. El sobremuestreo y promedio de las señales no afecta la potencia de la señal (Oppenheim y Schaffer 1998). Entonces, se aumenta el SNR porque el sobremuestreo disminuye la potencia del ruido y no afecta la potencia de la señal (Sillicon Labs 2013).

En términos prácticos, una señal limitada en banda se mide a menos de $\frac{1}{2}$ de la frecuencia de muestreo y después se sobremuestra con ayuda de la relación de sobremuestreo (OSR). Las muestras resultantes se promedian para los datos de salidas resultantes. Se suele sobremuestrear por un factor de cuatro para cada bit adicional de resolución o 6 dB de reducción de ruido (*ibid*).

$$f_{os} = 4^w \cdot f_s \quad \text{Ec.(4.7)}$$

w es el número adicional de bit de la resolución deseada

f_s es el requisito de frecuencia de muestreo original

f_{os} es la frecuencia de sobremuestreo

4.2 Tecnología ST3 Doppler Transcraneal de Modo M de Potencia – PMM

Tecnologías Spencer menciona que en el libro *Neuroimaging, An Issue of Neurologic Clinics* de Laszlo Mechtler se encuentra la información más reciente del uso de Doppler Transcraneal (TCD). En una parte de la página web de tecnologías Spencer¹ se hace una cita textual donde se menciona que el mayor avance técnico, desde la introducción del TCD, es el uso del Doppler Modo Movimiento de Potencia (PMD), lo que quiere decir que la tecnología ST3 Doppler Transcraneal de modo M de potencia (PMM) está basada en la tecnología PMD.

¹ <https://www.spencertechnologies.com>

El dispositivo TCD o sistema Spencer muestra formas de onda espectrales que representan la profundidad, dirección, e intensidad del flujo sanguíneo a través de una vasculatura intracraneal (vasos sanguíneos dentro del cráneo). El dispositivo de Tecnologías Spencer no solo mide el flujo sanguíneo directamente, sino que calcula la correlación de otros parámetros basándose en el flujo sanguíneo cerebral (CBF).

Hakimi, Alexandrov, y Garami (2020) mencionan que en el pasado, las máquinas TCD solamente mostraban las formas de onda espectrales. Esto ocasionaba que el operador tuviera que decidir cuál vaso había sido insonado para intentar obtener la misma forma de onda a diferentes profundidades – utilizando técnicas de inversión de la señal con diferentes enfoques o ventanas. Con la incorporación del PMD, los sonógrafos ahora son capaces de obtener la forma de onda espectral e identificar la profundidad de los vasos sanguíneos insonados, la dirección del flujo con respecto a la sonda, y la intensidad de la señal.

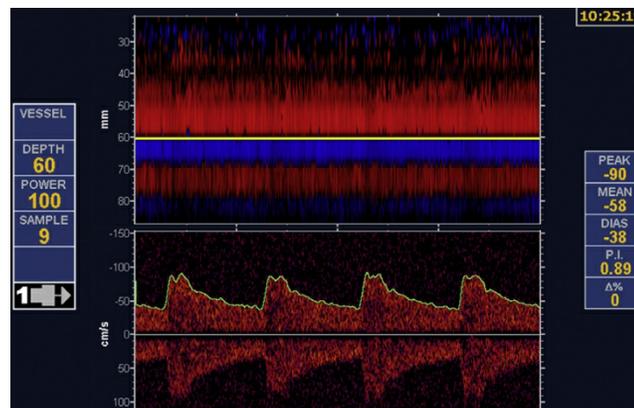


Figura 4.1: Señales TCD normales en pantalla PMM: arteria cerebral media (MCA) (rojo: 40 – 60 mm), arteria cerebral anterior (ACA) (azul: 60 – 70 mm), ACA contralateral (rojo: 70 – 80 mm), y MCA contralateral (azul: 80 – 90 mm). Figura obtenida de Hakimi *et al.* (2020).

4.3 Interfaz Gráfica de Usuario para Analizar Señales de Ultrasonido Doppler Transcraneales

El grupo de ingeniería aeroespacial y computacional, en colaboración con el departamento de ciencias cardiovasculares de la Universidad de Leicester, han desarrollado una GUI en Matlab® – denominada BrainTV – capaz de leer, analizar y mostrar los datos extraídos de los sistemas TCD Spencer y NK. El desarrollo de este

programa es la principal motivación del capítulo 3 de este trabajo de tesis, ya que la intensidad del eco de las señales Doppler no se muestra en el programa BrainTV. Los algoritmos presentados en el capítulo 3 serán evaluados y probablemente mejorados por el grupo de investigación encargado del desarrollo del programa BrainTV para su implementación y uso, aproximadamente, a partir de la publicación de este trabajo de tesis. La figura 4.2 muestra la GUI BrainTV en su versión de finales del 2019.

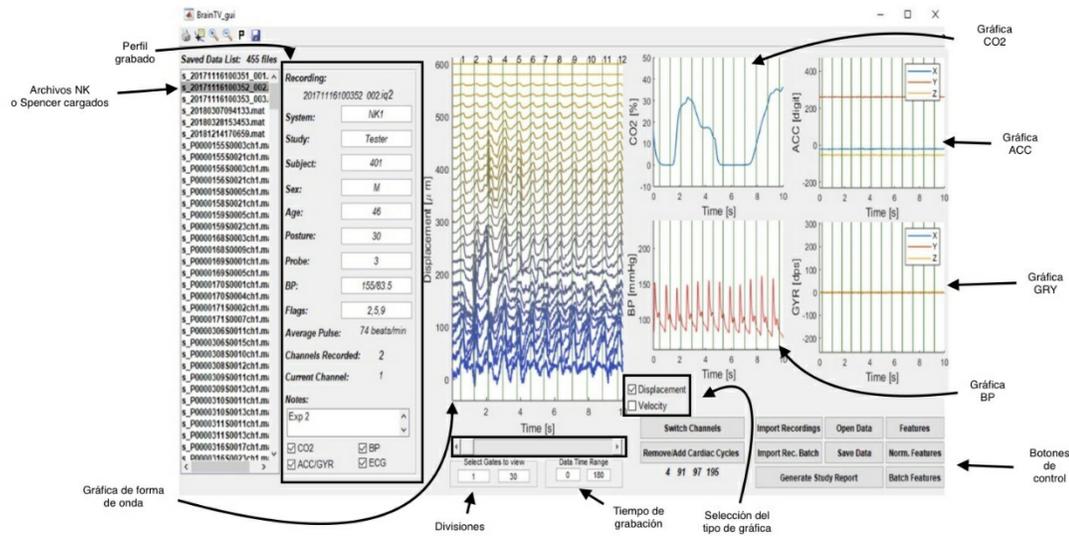


Figura 4.2: Interfaz gráfica de usuario BrainTV en Matlab®. Figura obtenida del manual de usuario de la GUI, propiedad privada de la Universidad de Leicester (2019). Adaptada y traducida por Zaleta-Andrade para este trabajo de tesis.

Capítulo 5

Conclusiones y trabajo futuro

5.1 Conclusiones

El estudio de la generación de los rayos y campos ultrasónicos permite comprender la razón de usar señales de RF pulsantes en los dispositivos de ultrasonido TCD en lugar de señales continuas. Los transductores juegan un papel importante en el estudio del movimiento cerebral, ya que son los encargados no solo de generar las señales ultrasónicas, sino de capturarlas junto con las perturbaciones recibidas a través de su viaje por el cerebro. El procesamiento de estas señales Doppler ha requerido años de estudio para encontrar la manera de extraer la mayor cantidad de información posible de las señales Doppler y correlacionarla con otros sistemas fisiológicos. El cálculo de la intensidad del eco de una sonda de ultrasonido Doppler transcraneal es un dato innovador para el grupo de investigación de la Universidad de Leicester, ya que hasta antes de la publicación de este trabajo de tesis el cálculo de la intensidad no formaba parte del procesamiento de señales Doppler efectuado por el programa BrainTV. Dicha implementación de la GUI que calcula las intensidades de las señales Doppler será evaluada por investigadores de la Universidad de Leicester para su implementación futura en el programa BrainTV. Dicho programa permite a este grupo de investigación procesar las señales Doppler obtenidas por los equipos de ultrasonido TCD para interpretar el movimiento cerebral con otros parámetros fisiológicos como el CO₂ del cuerpo, la presión arterial y, ahora, la intensidad con la que esas señales ultrasónicas retornan después de su viaje a través del cerebro.

El experimento hecho por RJ en el capítulo 2 pretende explicar la técnica reconstrucción de un rayo ultrasónico desde su parte experimental hasta el procesamiento de los datos con el fin de digitalizar un haz de ultrasonido. El objetivo final de la digitalización es lograr caracterizar el rayo ultrasónico para determinar su factibilidad en estudios médicos y comprobar si tendrá los resultados deseados por los

investigadores o médicos. El uso específico de la caracterización del rayo ultrasónico presentado en este trabajo de tesis es desconocido por el autor de este trabajo de tesis.

Matlab® es un lenguaje de programación que permite procesar datos digitales a muy alto nivel. La capacidad e innovación tecnológica de los equipos de ultrasonido TCD de NK y Tecnologías Spencer de procesar las señales Doppler, tal como se explica en el capítulo 2, permite que toda la información de las señales se contenga en un solo archivo digital. De esta manera, Matlab® y un computador pueden procesar las señales y extraer la mayor cantidad de información utilizando algoritmos como el de compresión de datos o el de aproximación de la intensidad inicial de las señales de ultrasonido Doppler.

5.2 Discusión

Aproximación inicial: señal de ultrasonido Doppler transcraneal generadora del eco

Por una parte, se puede argumentar que las formas de onda de la intensidad del eco no cambian, aunque en la fórmula de la intensidad del eco de una señal de ultrasonido TCD se utilice o no una referencia aproximada. De hecho, si la intensidad de referencia se desprecia, la forma de onda permanecerá idéntica. Esto se muestra en la figura 5.1.

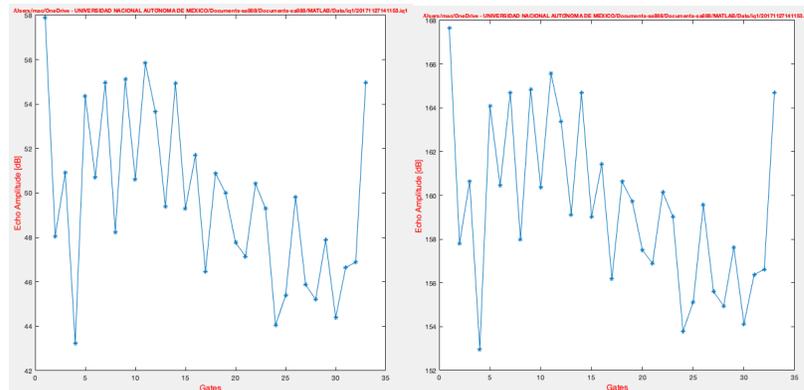


Figura 5.1: Cálculo de la intensidad de las señales de ultrasonido Doppler de 33 profundidades distintas. A la izquierda se utiliza el valor de referencia inicial con el método de aproximación propuesto en el capítulo 3 de este trabajo de tesis. A la derecha se desprecia la intensidad de referencia (valor casi cero). Figura obtenida de la GUI del cálculo de la intensidad en Matlab®.

Reducción de la tasa de muestreo o compresión de las Señales

La técnica de reducción de la tasa de muestreo también reduce el número de amplitudes analizadas durante la aproximación. Sin embargo, dicha reducción aumenta el periodo de muestreo por un factor M (Tan *et al.* 2013) y por lo tanto la nueva frecuencia de muestreo se vuelve:

$$\frac{f_{sM}}{2} = \frac{1}{M \cdot T} = \frac{f_s}{M} \quad \text{Ec.(5.1)}$$

$\frac{f_{sM}}{2}$ es la frecuencia de Nyquist;
 $M \cdot T$ es el nuevo periodo de muestreo;
 f_s es la frecuencia de muestreo original; y
 M es el factor entero de reducción de la tasa de muestreo.

Lo anterior quiere decir que la frecuencia de Nyquist, después de la reducción de la tasa de muestreo, disminuye M veces. Si la señal a la que se le reduce la tasa de muestreo tiene componentes más grandes que la nueva frecuencia de Nyquist, entonces se introduce solapamiento en la nueva señal. *I.e.*, existe aliasing en la señal con menor tasa de muestreo si:

$$f > \frac{f_s}{2 \cdot M} \quad \text{Ec.(5.2)}$$

f es una componente en frecuencia de la señal con reducción en su tasa de muestreo;
 $\frac{f_s}{2 \cdot M}$ es la frecuencia de Nyquist de la señal con reducción en su tasa de muestreo.

Control de potencia de transmisión en el programa de adquisición de datos del sistema NK

El programa de adquisición de datos muestra una opción de control de potencia de transmisión. La intensidad está directamente relacionada con la potencia de la señal. Entonces, la intensidad de la señal de insonación puede que esté relacionada con la potencia de transmisión. *I.e.*, es posible que la intensidad de referencia se pueda recuperar desde el programa de adquisición de datos del prototipo BrainTV de NK. También puede ser que la intensidad de referencia se pueda recuperar directamente desde la hoja de datos técnicos del sistema NK.

Los transductores de los sistemas de ultrasonido TCD NK y Spencer no pueden utilizar osciladores hechos de un material PVDF debido a que su uso es para frecuencias superiores a los 15 MHz. Esto rebasa los parámetros de 2 MHz de la señal portadora del transductor de los equipos TCD Spencer y NK. Por lo tanto, es probable que los sistemas de Nihon Kohden y Tecnologías Spencer utilicen transductores hechos a base de material PZT-5H, según las propiedades mostradas en la tabla 2.2 de Evans et al. (2000).

Exportación de datos entre aplicaciones de equipos de ultrasonido TCD

Si la exportación de datos entre los distintos equipos de ultrasonido TCD no fuera posible, los fabricantes fueran los únicos capaces de desarrollar soluciones específicas para las necesidades de cada usuario. Por ejemplo, el desarrollo del programa BrainTV no hubiese sido posible a no ser que el fabricante del equipo TCD lo hubiera desarrollado él mismo. Esta situación ocasionaría que tanto los equipos de ultrasonido TCD como los programas fueran más costosos. La producción de un solo producto acabaría dependiendo del número de usuarios con necesidades distintas. La cantidad de oferta y demanda disminuiría y con ella la motivación de continuar con el desarrollo tecnológico del producto y de sus aplicaciones.

5.3 Trabajo Futuro

Investigación futura

Después de una investigación previa del contenido relacionado con el ultrasonido Doppler en la biblioteca que la UNAM pone a disposición de alumnos de pregrado, los resultados encontrados con contenido relacionado al tema fue limitado. Además, la mayoría de este contenido está en idioma inglés, por lo que la información relacionada al ultrasonido Doppler se vuelve más limitada aún. Esto reduce la capacidad de alumnos de pregrado de hacer investigación en el tema de ultrasonido Doppler.

Esta tesis busca motivar a futuros alumnos de licenciatura de la Universidad Nacional Autónoma de México a introducirse en el tema del ultrasonido Doppler Transcraneal para producir trabajos de investigación similares a este trabajo de tesis y promover así investigación de posgrado en la detección no invasiva del daño cerebral en humanos desde un punto de vista de ingeniería. Si el lector de este trabajo de tesis entiende el idioma inglés, podrá iniciar su propia investigación basándose en la notable lista de referencias relacionadas con el ultrasonido Doppler que se presentan en la bibliografía.

Algoritmos

La minuciosa descripción de los algoritmos presentados en este trabajo de tesis, en el capítulo 2 y 3, sientan una base para la mejora de los mismos o para el desarrollo de otros algoritmos para que logren extraer datos que puedan ser interesantes para la detección no invasiva del daño cerebral en humanos. Es importante seguir iterando con más estudios en el área del ultrasonido TCD para que alumnos de Licenciatura de la UNAM puedan ir al siguiente nivel y hacer investigación de posgrado con investigadores de otras partes del mundo. De esta manera, aumentar la tasa de desarrollo de un sistema que sea capaz de detectar el daño cerebral de forma no invasiva, y en tiempo real, en casos de emergencia, por ejemplo, en ambulancias.

Apéndice A

Acrónimos

Acrónimo	Definición (Español)	Definición (Inglés)
ADC	Convertidor analógico a digital	Analogue-to-Digital Converter
BP	Presión Sanguínea	Blood Pressure
BrainTV	Velocímetro de Tejido Cerebral	Brain Tissue Velocimetry
CBF	Flujo Sanguíneo Cerebral	Cerebral Blood Flow
CO2	Dióxido de Carbono	Carbon Dioxide
DAC	Convertidor digital a analógico	Digital-to-Analogue Converter
DAQ	Adquisición de datos	Data Acquisition
DSO	Osciloscopio de Almacenamiento Digital	Digital Storage Oscilloscope
ECG	Electrocardiograma	Electrocardiogram
EI	Intensidad o Amplitud del Eco	Eco Intensity or Amplitude
ENOB	Número Efectivo de Bits	Effective Number of Bits
EQ	Ruido de Cuantización	Quantisation Error
ESD	Densidad Espectral de Energía	Energy Spectral Density
GRY	Giroscopio	Gyroscope
GUI	Interfaz Gráfica de Usuario	Graphic User Interface
ICP	Presión Intracraneal	Intracranial Pressure
IQ	En Fase y Cuadratura	In Phase Quadrature
NHS	Servicio Nacional de Salud (del Reino Unido)	National Health Service (in the UK)
NK	Nihon Kohden	Nihon Kohden

OSR	Relación de sobremuestreo	Oversampling Ratio
PMD	Doppler modo movimiento de Potencia	Power Motion-mode Doppler
PMM	ST3 Doppler Transcraneal de modo M de potencia	ST3 Power-M-mode transcranial Doppler
PRF	Frecuencia de Repetición de Pulso	Pulse Repetition Frequency
PW	Sistemas de Onda Pulsada	Pulsed Wave System
RF	Radio Frecuencia	Radio Frequency
RMS	Raíz Media Cuadrada	Root Mean Square
S/H	Muestreo y Retención	Sample and Hold
SNR	Relación Señal a Ruido	Signal-to-Noise Ratio
TCD	Doppler Transcraneal o Ultrasonografía Doppler	Transcranial Doppler or Transcranial Doppler Ultrasonography/Ultrasound
TGC	Compensación de Ganancia de Tiempo	Time Gain Compensation

Apéndice B

Códigos del Cálculo de la Intensidad de señales de Ultrasonido Doppler Transcraneales

Los códigos y comentarios de los códigos que se muestran a continuación fueron escritos en Matlab® R2019b durante la estancia de investigación en la Universidad de Leicester del autor de este trabajo de tesis. Estos códigos operan en conjunto siguiendo el flujo de la figura 3.7 del capítulo 3 y tienen el propósito de llevar a cabo el cálculo de la intensidad de las señales de ultrasonido TCD a través de la GUI que se muestra en la figura 3.11 del capítulo 3. El único código que opera por separado es `peaksfinder.m`. Se puede encontrar la descripción de esta función y de todas las demás funciones de este apéndice en el capítulo 3.

Interfaz Gráfica de Usuario - `gui_Echo.m`

```
function varargout = gui_Echo(varargin)
%%Código modular de extracción del eco de una sonda de ultrasonido Doppler
%%transcraneal con MATLAB.
%%Before running the code, ensure the Signal Processing Toolbox is
%%installed.

%Author: Sergio Zaleta-Andrade, Estudiante de Lic.
%%
% GUI_ECHO MATLAB code for gui_Echo.fig
%   GUI_ECHO, by itself, creates a new GUI_ECHO or raises the existing
%   singleton*.
%
%   H = GUI_ECHO returns the handle to a new GUI_ECHO or the handle to
%   the existing singleton*.
%
%   GUI_ECHO('CALLBACK',hObject,eventData,handles,...) calls the local
%   function named CALLBACK in GUI_ECHO.M with the given input arguments.
%
%   GUI_ECHO('Property','Value',...) creates a new GUI_ECHO or raises the
%   existing singleton*. Starting from the left, property value pairs are
%   applied to the GUI before gui_Echo_OpeningFcn gets called. An
%   unrecognized property name or invalid value makes property application
%   stop. All inputs are passed to gui_Echo_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES
```

```

% Edit the above text to modify the response to help gui_Echo

% Last Modified by GUIDE v2.5 28-May-2020 10:47:38

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @gui_Echo_OpeningFcn, ...
                  'gui_OutputFcn',  @gui_Echo_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before gui_Echo is made visible.
function gui_Echo_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to gui_Echo (see VARARGIN)

% Choose default command line output for gui_Echo
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes gui_Echo wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = gui_Echo_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in pb_importfile.

```

```

function pb_importfile_Callback(hObject, eventdata, handles)
%Cleaning both axes
cla(handles.axes1);
cla(handles.axes2);
%Initialising
    CFd = pwd;
%-----
----
% --- *.mat file open
    [file_name, pathname, ~] = uigetfile({ '*.iq1;*.iq2;*.mat',...
        ' Models (*.iq1,*.iq2) or MATLAB files (*.mat)'});
%-----
----
setappdata(handles.figure1,'fileName', file_name);
cd(CFd);

if isnumeric(pathname)
    %Avoiding error due to cancellation
        uiwait(msgbox('Import Data Action Cancelled'));
else
    %setting app data
        fullfile_name=fullfile(pathname,file_name);
        setappdata(handles.figure1,'file_name',fullfile_name);

    %extracting IQ data and other parameters
    %IQdata, IQ components in a complex form
    %PRF, Hz
    %Fc, Mhz
    %systemname, Spencer or NK
    %Time, s
    avsample=100; %For averaging
    Fcbeam=2e6; %Hz
    [IQdata, PRF, Fc, Time, systemname]=readfile(pathname, file_name);%PRF,
    KHz - Fc, Mhz
    [IQdatabeam Timebeam]=beam(Fcbeam, Time); %IQdatabeam is not a cell.
    %Assesing whether IQdata has 1 or 2 channels
    [nn,mm]=size(IQdata{1,1}); %mm is num of col -ngates-n, nn num of rows
    [numchannels,~]=size(IQdata);
    if numchannels == 1
        IQdata{1,1}=[IQdatabeam IQdata{1,1}]; %IQdatabeam is concatenated in the
        first collumn of IQdata
    else
        IQdata{1,1}=[IQdatabeam(1:end-8160) IQdata{1,1}]; %IQdatabeam is
        concatenated in the first collumn of IQdata
        IQdata{2,1}=[IQdatabeam(8161:end) IQdata{2,1}]; %
    end

    %Assesing whether IQdata is Spencer or NK
    switch systemname
        case 'Spencer'
            %[row,~]=size(IQdata(1,1)); uncomment this line when
            %IMPLEMENTING VSDopp_v2... or VSDoppExt...

            if numchannels==1 %1 channel
                [avdata,~, Time]=avData(IQdata{1,1},avsample, Time);
                avdatanorm=norm2one(avdata);

```

```

        echo=zeros(1,mm-1);%Preallocating for saving echo
        rmsref=rms(avdatanorm(:,1));
        for i=1:1:mm-1
            %Calculating RMS value of the reference and all
            %the signals (from the 34 gates) to calculate their
            %echo

echo(1,i)=10*log10(rms(avdatanorm(:,i+1))/rms(avdatanorm(:,1)));
        end
        %Plot time
    else %2 channels
        %Not tested yet because no 2 channel Spencer file was
        %found in the R drive
        Time=Time/2;%divide time by 2

        [avdata1,~,Time1]=avData(IQdata{1,1},avsample, Time);
        [avdata2,~,~]=avData(IQdata{2,1},avsample, Time);
        avdatanorm1=norm2one(avdata1);
        avdatanorm2=norm2one(avdata2);
        rmsref1=rms(avdatanorm1(:,1));
        rmsref2=rms(avdatanorm2(:,1));
        echo1=zeros(1,mm-1);%Preallocating for saving echo
        echo2=zeros(1,mm-1);
        for i=1:1:mm-1
            %Calculating RMS value of the reference and all
            %the signals (from the 34 gates) to calculate their
            %echo

echo1(1,i)=10*log10(rms(avdatanorm1(:,i+1))/rms(avdatanorm1(:,1)));

echo2(1,i)=10*log10(rms(avdatanorm2(:,i+1))/rms(avdatanorm2(:,1)));
        end
        %Implement here VSDopp_v2_BrainTV(...) for raw velocity

        %Implement here VSDoppExt_BrainTV(...) for displacement or
        %velocity

    case 'NK'
        if numchannels==1 %1 channel
            [avdata,~,Time]=avData(IQdata{1,1},avsample, Time);
            avdatanorm=norm2one(avdata);
            echo=zeros(1,mm-1);%Preallocating for saving echo
            rmsref=rms(avdatanorm(:,1));
            for i=1:1:mm-1
                %Calculating RMS value of the reference and all
                %the signals (from the 34 gates) to calculate their
                %echo

echo(1,i)=10*log10(rms(avdatanorm(:,i+1))/rms(avdatanorm(:,1)));
            end

        else %2 channels
            Time=Time(1:end/2); %Time divided by 2

            [avdata1,~,Time1]=avData(IQdata{1,1},avsample, Time);
            [avdata2,~,~]=avData(IQdata{2,1},avsample, Time);

```

```

        avdatanorm1=norm2one(avdata1);
        avdatanorm2=norm2one(avdata2);
        rmsref1=rms(avdatanorm1(:,1));
        rmsref2=rms(avdatanorm2(:,1));
        echo1=zeros(1,mm-1);%Preallocating for saving echo
        echo2=zeros(1,mm-1);
        for i=1:1:mm-1
            %Calculating RMS value of the reference and all
            %the signals (from the 34 gates) to calculate their
            %echo

echo1(1,i)=10*log10(rms(avdatanorm1(:,i+1))/rms(avdatanorm1(:,1)));

echo2(1,i)=10*log10(rms(avdatanorm2(:,i+1))/rms(avdatanorm2(:,1)));
            end
        end
    otherwise
        uiwait(msgbox('An error has occurred'));
    end

%%
%Left side plot
yticks('auto');
if numchannels==1
    %%
    %Plotting process
    echol=length(echo(1,:));
    %%
    %Trying to plot DB values
    axes(handles.axes1);

    %Preallocating for saving each gate with its corresponding echo
    gates=zeros(1,echol);
    decibels=zeros(1,echol);
    aux=500; %It helps separating the plots

    for i=1:1:echol
        plot(1,echo(1,i)+aux,'LineStyle','none','Marker','o');
        gates(1,i)=i; decibels(1,i)=echo(1,i);
        %plot(1,maxlow(i,1),'LineStyle','none','Marker','.');
        text(1.2,aux,num2str(i));
        text(0.5,aux,num2str(echo(1,i)),'FontSize',10);
        %text(1.2,aux,num2str(maxlow(i,1)),'FontSize',5);
        hold on;
        aux=aux+500;
    end
    ylabel('Echo Amplitud [dB]','FontSize',12,'FontWeight','bold');

    %Plotting echo vs gates
    axes(handles.axes2);
    plot(gates,decibels,...
        'Marker','*'); %Implement 'MarkerIndices' to see markers in each
symbol
    title(fullfile_name,'FontSize',8,'FontWeight','bold','Color','red');
    ylabel('Echo Amplitude [dB]','FontSize',12,'Color','red');
    xlabel('Gates','FontSize',12,'Color','red');
else

```

```

%implement algorithm for two channels.
%A plot like axes1 won't be displayed.
%Preallocating for saving each gate with its corresponding echo
echol=length(echol(1,:));
gates=zeros(1,echol); %both channels contains the same number of gates
decibels=zeros(2,echol);
echol=length(echol(1,:)); %echo1 and echo2 will always be the same
length.
for i=1:1:echol
    gates(1,i)=i; decibels(1,i)=echo1(1,i); decibels(2,i)=echo2(1,i);
end

    %Plotting echo vs gates
axes(handles.axes2);
plot(gates,decibels(1,:),...
    'Marker','*'); %Implement 'MarkerIndices' to see markers in each
symbol
title(fullfile_name,'FontSize',8,'FontWeight','bold','Color','red');
ylabel('Echo Amplitude [dB] - Channel 1','FontSize',12,'Color','red');
xlabel('Gates','FontSize',12,'Color','red');

axes(handles.axes3);
plot(gates,decibels(2,:),...
    'Marker','*'); %Implement 'MarkerIndices' to see markers in each
symbol
title(fullfile_name,'FontSize',8,'FontWeight','bold','Color','red');
ylabel('Echo Amplitude [dB] - Channel 2','FontSize',12,'Color','red');
xlabel('Gates','FontSize',12,'Color','red');
end

%%
%plotting on axes2 and axes3.

%Displaying in GUI RMS value of the reference signal
% if numchannels>1
%     %Implement for two channels
%
% else
%     set(handles.st_rmsref,'String',num2str(rmsref));
% end
end

% --- Executes on button press in pb_exit.
function pb_exit_Callback(hObject, eventdata, handles)
% hObject    handle to pb_exit (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
clc;close;

% --- Executes on button press in pb_cleanall.
function pb_cleanall_Callback(hObject, eventdata, handles)

```

```

% hObject    handle to pb_cleanall (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
axes(handles.axes1);
cla(handles.axes1);
title('');
ylabel('');
xlabel('');
axes(handles.axes2);
cla(handles.axes2);
title('');
ylabel('');
xlabel('');
axes(handles.axes3);
cla(handles.axes3);
title('');
ylabel('');
xlabel('');
uiwait(msgbox('All data have been cleaned successfully.'));

```

Lectura de archivos de equipos Spencer y NK - readfile.m

```

function [IQ, PRF, Fc, Time, systemname] = readfile(pathname, file_name)
%IQ, IQ components of Spencer or NK files, in complex form
%PRF, pulse repetition frequency
%Fc, Frequency of the beam -usually 2MHz-
%Time, Sampling time
%systemname, Spencer or NK files
%%

if strcmp(file_name(end-3:end),'.mat')
    %Spencer file has been imported
    uiwait(msgbox('Spencer file has been imported succesfully'));
    systemname='Spencer';

    %Preallocating for IQ components
    IQ=cell(1,1);

    %Extracting data. This part of the code must be updated to accept 2
channels data
    FName0=[pathname, file_name];
    DATA=load(FName0);
    IQ{1,1}=DATA.CH1_DATA;
    PRF=DATA.PRF;
    Fc=2.0;% Carrier Frequency, MHz
    [nn,~] = size(IQ{1,1});

    %Temporal time
    Time = (0:nn-1)*(1/PRF);
    Time = Time';
else
    %NK file has been imported
    uiwait(msgbox('NK file has been imported succesfully'));
    systemname='NK';

    %*.iq1 file read

```

```

fullfile_name=fullfile(pathname,file_name);
fileID=fopen(fullfile_name);
header=fread(fileID,8,'float');
Fc=header(4); %Carrier Frequency, MHz
PRF = header(8)*1000; %KHz
%Extracting data
av_sample = PRF/500; %use in downsamplig, later in the code
idx1=strfind(fullfile_name, '.iq1');
if idx1 > 0
    %1 channel recorder
    file_nameav=strcat(fullfile_name(1:idx1-1),'.ai1');
else
    %2 channels recorded
    idx2=strfind(fullfile_name, '.iq2');
    file_nameav=strcat(fullfile_name(1:idx2-1),'.ai2');
end

%Reading IQ and AD data???
if PRF > 500 %downsampling process?
    if isempty(idx1) %2 channels
        while ~feof(fileID)%Reading file line by line
            IQdata = transpose(fread(fileID,...
                [117*2+2+32,PRF*10000],'short'));%Read as
word - 117 is the #IQ - PRF*10000 is the read length.
            end
            %separating ch1 and ch2
            [row,~]=size(IQdata);
            IQdatach1=IQdata(1:2:row,:);

            %downsampling to 500Hz
            avdata_ds =
downsample(IQdatach1(:,117*2+2+1:117*2+2+16),av_sample);

            clear IQdatach1;
            fclose(fileID);

            fileID_av=fopen(file_nameav);

            %Read IQ and averaged data for high PRF
            while ~feof(fileID_av)

IQavdata=transpose(fread(fileID_av,[34*2+17,PRF*10000],'float')); %Read as
float - 34 is the #IQ averaged - PRF*10000 is the read length
            end
            %separating ch1 and ch2
            [row,~]=size(IQdata);
            IQavdatach1=IQavdata(1:2:row,:);
            IQavdatach2=IQavdata(2:2:row,:);

            %downsampling to 500Hz
            avdata2_ds =
downsample(IQavdata(:,34*2+1:34*2+17),av_sample);%- 34 is the #IQ averaged

            %extracting IQ for channel 1 and 2
            Iavch1=IQavdatach1(:,1:2:34*2);
            Qavch1=IQavdatach1(:,2:2:34*2);
            Iavch2=IQavdatach2(:,1:2:34*2);

```

```

        Qavch2=IQavdatach2(:,2:2:34*2);
        IQ{1,1}=complex(Iavch1,Qavch1);
        IQ{2,1}=complex(Iavch2,Qavch2);

        clear Iavch1 Qavch1 Iavch2 Qavch2 IQavdatach1
IQavdatach2;

        %clear Iavch1 Qavch1 Iavch2 Qavch2 IQavdata;
        fclose(fileID);
    else %1 channel
        while ~feof(fileID)%Reading file line by line
            IQdata = transpose(fread(fileID,...
                [117*2+2+32,PRF*10000],'short'));%Read as word -
117 is the #IQ - PRF*10000 is the read length.
            end

            %downsampling to 500Hz
            avdata_ds =
downsample(IQdata(:,117*2+1:34*2+16),av_sample);%- 117 is the #IQ averaged

            clear IQdata;
            fclose(fileID);

            fileID_av=fopen(file_nameav);

            %Read IQ and averaged data for high PRF
            while ~feof(fileID_av)

IQavdata=transpose(fread(fileID_av,[34*2+17,PRF*10000],'float')); %Read as
float - 34 is the #IQ averaged - PRF*10000 is the read length
            end

            %downsampling to 500Hz
            avdata2_ds =
downsample(IQavdata(:,34*2+1:34*2+17),av_sample);%- 34 is the #IQ averaged

            %extracting IQ for channel 1
            Iav=IQavdata(:,1:2:34*2);
            Qav=IQavdata(:,2:2:34*2);
            IQ{1,1}=complex(Iav,Qav);

            clear Iav Qav IQavdata;
            fclose(fileID);
        end
    else
        if isempty(idx1)%2 channels
            %Reading file
            IQdata = transpose(fread(fileID,...
                [117*2+2+32,180.0*PRF],'short'));%Read as word -
117 is the #IQ - 180.0 is the record time per file in seconds
            %separating ch1 and ch2
            [row,~]=size(IQdata);
            IQdatach1=IQdata(1:2:row,:);
            %downsampling?
            avdata_ds = IQdatach1(:,117*2+2+1:117*2+2+16);

            clear IQdata; %Maybe delete this line

```

```

        fclose(fileID);%Maybe delete this line

        fileID_av=fopen(file_nameav);

        %Read IQ and averaged data for high PRF
        IQavdata=transpose(fread(fileID_av,...
        [34*2+17,180.0*PRF],'float')); %Read as float - 34
is the #IQ averaged - 180.0 is the record time per file in seconds
        %separating ch1 and ch2
        [row,~]=size(IQavdata);
        IQavdatach1=IQavdata(1:2:row,:);
        IQavdatach2=IQavdata(2:2:row,:);
        %downsampling to 500Hz
        avdata2_ds = IQavdata(:,34*2+1:34*2+17);%- 34 is
the #IQ averaged

        %extracting IQ for channel 1 and 2
        Iavch1=IQavdatach1(:,1:2:34*2);
        Qavch1=IQavdatach1(:,2:2:34*2);
        Iavch2=IQavdatach2(:,1:2:34*2);
        Qavch2=IQavdatach2(:,2:2:34*2);
        IQ{1,1}=complex(Iavch1,Qavch1);
        IQ{2,1}=complex(Iavch2,Qavch2);

        clear Iavch1 Qavch1 Iavch2 Qavch2 IQavdatach1
IQavdatach2;
    else %1 channel
        %Reading file
        IQdata = transpose(fread(fileID,...
        [117*2+2+32,180.0*PRF],'short'));%Read as word -
117 is the #IQ - 180.0 is the record time per file in seconds
        %downsampling?
        avdata_ds = IQdata(:,117*2+2+1:117*2+2+16);

        clear IQdata %Maybe delete this line
        fclose(fileID);%Maybe delete this line

        fileID_av=fopen(file_nameav);

        %Read IQ and averaged data for high PRF
        IQavdata=transpose(fread(fileID_av,[34*2+17,180.0*PRF],'float')); %Read as
float - 34 is the #IQ averaged - PRF*10000 is the read length

        %downsampling to 500Hz
        avdata2_ds = IQavdata(:,34*2+1:34*2+17);%- 34 is
the #IQ averaged- 180.0 is the record time per file in seconds

        %extracting IQ for channel 1
        Iav=IQavdata(:,1:2:34*2);
        Qav=IQavdata(:,2:2:34*2);
        IQ{1,1}=complex(Iav,Qav);

        clear Iav Qav IQavdata;
        fclose(fileID_av);
end

```

```

        end
        %time
        [mm,~]=size(avdata2_ds);
        Time=(0:mm-1)'*(1/500); %subtract -3 to mm when trying to
plot rawvelocity
end

```

Generación de las amplitudes de las señales de ultrasonido

Doppler - beam.m

```

function [beam, Time]=beam(Fc, Time)
%The aim of this code is to generate the amplitudes for the signal of the
%ultrasound beam.

%NOTES:
%1/PRF is used as Ts. I haven't understood why, Fc=2Mhz for Dopp Ultrasound
%Fc, Hz
%%
%Instructions for the user
uiwait(msgbox('select peaks.mat file'));
%Opening *.mat file
[filename, pathname]=uigetfile({'*.mat', 'MATLAB files(*.mat)'});

%Calculating Amplitude ~reference~ for creating the beam
[reference, ~]=amplitude(pathname, filename);
reference=0.000001; %Uncomment/comment this line if the initial reference I0
wants
%to be personalised.

timel=length(Time);

%Preallocating
beam=zeros(timel,1);

for i=1:1:timel
    beam(i,1)=reference*sin(2*pi*Fc*Time(i)); %
end

beam=complex(beam); %If complex values are not required, eliminate this line.

```

Compresión de las señales TCD – avData.m

```

function [av_data, IQdata, t] = avData(IQDATA, Av_sample, Time)
%Average IQDATA over a number ~ngiven by Av_sample~
%Returns The abs value of complex data
%    The IQDATA but averaged
%    Sampling time as a vector

```

```

%Number of rows; samples.
[row,col]=size(IQDATA);
x=1:Av_sample:row;

%Preallocating size for av_data
av_data=zeros(length(x)-1,col);
IQdata=zeros(length(x)-1,col);
t=zeros(length(x)-1,1);
for i = 1:col
    for k=1:length(x)-1
        av_data(k,i) = sqrt(real(IQDATA(x(k),i))^2 + imag(IQDATA(x(k),i))^2);
        IQdata(k,i)=IQDATA(x(k),i);
        if i==1
            t(k,i)=Time(k,1)*100;
        end
    end
end
end
end

```

Normalización a uno de las amplitudes de las señales comprimidas – norm2one.m

```

function av_datanorm=norm2one(av_data)
%Normalises av_data
if isreal(av_data)
    base=max(max(av_data));
    av_datanorm=av_data./base;
else
    avabs_data=abs(av_data);
    base=max(max(avabs_data));
    av_datanorm=avabs_data./base;
end

```

Método de aproximación de la intensidad inicial de las señales de ultrasonido TCD - amplitude.m

```

function [reference, deltavg]=amplitude(pathname, filename)
%The aim of this code is to approximate the amplitude of the reference
%signal used to plot the ultrasound echo.
%It uses the peaks calculated using 'peaksfinder.m'

%reference = Max peak of a *.mat file database
%deltavg = the average of the deviation between all sorted peaks in
%ascending order

FName=[pathname filename];
if isnumeric(pathname)
    uiwait(msgbox('Import Data Action Cancelled'));
else
    DATA=load(FName);
    peaks=DATA.peaks;
    sortpeaks=sort(peaks);%<-- Maybe move this line

```

```

maxpeak=max(peaks); %Calculating maximum peak of all data
peaks1=length(sortpeaks);
norepeaks=0;

%Avoiding repetition in values
for i=1:1:peaks1-1
    if sortpeaks(i+1)~=sortpeaks(i) && norepeaks(end)~=sortpeaks(i)
        norepeaks=[norepeaks sortpeaks(i) sortpeaks(i+1)]; %#ok<AGROW>
    end
end
norepeaks=norepeaks(2:end); %Getting rid of the 1st 0
peaks1=length(norepeaks);% New length with no repeated elements
%Preallocating
delta=zeros(1,peaks1-1);

%Calculating deviation of each sorted peak
for i=1:1:peaks1-1
    delta(i)=abs(norepeaks(i+1)-norepeaks(i));
end
deltavg=mean(delta); %Average deviation

reference = maxpeak+deltavg;
end

```

Cálculo de la amplitud máxima entre todas las señales de un archivo de los equipos Spencer o NK – peaksfinder.m

```

function peaksfinder
%The aim of this code is to generate a 'peaks.mat' file that will help to
%approximate the amplitude of a reference signal

%The interaction is with the Command Window.
% -0 for stop running
% -1 up to 9 to continue
%NOTES:
% -'peaks.mat' is created automatically, even if it has not been created
% yet.
% -a the max peak of all the signals within the file will be saved into
% 'peaks.mat'
%%
CFd=pwd;
userin=1;
while userin~= 0
userin=input('Inset 0 to stop running or other number to continue: ');
if userin == 0
    continue;
end
%Initialising
    CFd = pwd;
    avsample=100;
%-----
---
```

```

uiwait(msgbox('Select a Spencer or NK file'));
[file_name, pathname, ~] = uigetfile({'*.iq1;*.iq2;*.mat',...
' Models (*.iq1,*.iq2) or MATLAB files (*.mat)'});

if isnumeric(pathname)
    uiwait(msgbox('Import Data Action Cancelled'));
    userin=0;
else
    fullfile_name=fullfile(pathname, file_name);
    [IQdata, ~, ~, Time, systemname]=readfile(pathname, file_name);
    %Assesing whether IQdata has 1 or 2 channels
    [numchannels,~]=size(IQdata);
    %Assesing whether IQdata is Spencer or NK
    switch systemname
        case 'Spencer'
            if numchannels==1 %1 channel
                [avdata,~, Time]=avData(IQdata{1,1},avsample, Time);
                if isreal(avdata)
                    base=max(max(avdata));
                else
                    avabs_data=abs(avdata);
                    base=max(max(avabs_data));
                end
            else %2 channels
                %Not tested yet because no 2 channel Spencer file was
                %found in the R drive
                Time=Time/2;%divide time by 2
                [avdata1,~, Time1]=avData(IQdata{1,1},avsample, Time);
                [avdata2,~,~]=avData(IQdata{2,1},avsample, Time);
                if isreal(avdata1)
                    base1=max(max(avdata1));
                    base2=max(max(avdata2));
                else
                    avabs_data1=abs(avdata1);
                    avabs_data2=abs(avdata2);
                    base1=max(max(avabs_data1));
                    base2=max(max(avabs_data2));
                end
                base=[base1 base2];
            end
        case 'NK'
            if numchannels==1 %1 channel
                [avdata,~, Time]=avData(IQdata{1,1},avsample, Time);
                if isreal(avdata)
                    base=max(max(avdata));
                else
                    avabs_data=abs(avdata);
                    base=max(max(avabs_data));
                end
            else %2 channels
                Time=Time(1:end/2); %Time divided by 2

                [avdata1,~, Time1]=avData(IQdata{1,1},avsample, Time);
                [avdata2,~,~]=avData(IQdata{2,1},avsample, Time);

                if isreal(avdata1)
                    base1=max(max(avdata1));
                    base2=max(max(avdata2));
                end
            end
        end
    end
end

```

```

        else
            avabs_data1=abs(avdata1);
            avabs_data2=abs(avdata2);
            base1=max(max(avabs_data1));
            base2=max(max(avabs_data2));
        end
        base=[base1 base2];
    end
    otherwise
        uiwait(msgbox('An error has occurred'));
    end
    namefile='peaks.mat';
    %Writing base on a file
    if ~exist(namefile,'file')
        %Create the file for the first time
        newDATA.peaks=base;
        save(namefile,'-struct','newDATA');
    else
        %Open the file to write it adding new value
        newDATA=load(namefile);
        newDATA.peaks=[newDATA.peaks base];
        save(namefile,'-struct','newDATA');
    end
end
end
end

```

Apéndice C

Códigos para la reconstrucción de la sección transversal de un rayo ultrasónico obtenido experimentalmente utilizando Matlab®

Los códigos y comentarios de los códigos que se muestran a continuación fueron escritos en Matlab® R2019b durante la estancia de investigación en la Universidad de Leicester del autor de este trabajo de tesis. Estos códigos operan en conjunto para llevar a cabo la reconstrucción de la sección transversal de un rayo ultrasónico obtenido experimentalmente utilizando Matlab® a través de la GUI que se muestra en la figura 2.10 del capítulo 2. Se puede encontrar la descripción de cada uno de los siguientes códigos en el capítulo 2.

Interfaz Gráfica para reconstruir la sección transversal de un rayo ultrasónico – beampattern_GUI.m

```
function varargout = beampattern_GUI(varargin)
%-----
%*AUTHOR(S):*
%      Sergio Zaleta updated 07/02/2020
%-----
% BEAMPATTERN_GUI MATLAB code for beampattern_GUI.fig
%   BEAMPATTERN_GUI, by itself, creates a new BEAMPATTERN_GUI or raises
the existing
%   singleton*.
%
%   H = BEAMPATTERN_GUI returns the handle to a new BEAMPATTERN_GUI or the
handle to
%   the existing singleton*.
%
%   BEAMPATTERN_GUI('CALLBACK',hObject,eventData,handles,...) calls the
local
%   function named CALLBACK in BEAMPATTERN_GUI.M with the given input
arguments.
%
%   BEAMPATTERN_GUI('Property','Value',...) creates a new BEAMPATTERN_GUI
or raises the
%   existing singleton*. Starting from the left, property value pairs are
%   applied to the GUI before beampattern_GUI_OpeningFcn gets called. An
%   unrecognized property name or invalid value makes property application
```

```

%      stop. All inputs are passed to beampattern_GUI_OpeningFcn via
varargin.
%
%      *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help beampattern_GUI

% Last Modified by GUIDE v2.5 07-Feb-2020 22:18:09

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @beampattern_GUI_OpeningFcn, ...
                  'gui_OutputFcn',  @beampattern_GUI_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before beampattern_GUI is made visible.
function beampattern_GUI_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to beampattern_GUI (see VARARGIN)

% Choose default command line output for beampattern_GUI
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes beampattern_GUI wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = beampattern_GUI_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in pb_start.
function pb_start_Callback(hObject, eventdata, handles)
% hObject    handle to pb_start (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

%Displaying messages to the user
step1='Select a file containing X0 and Y0 axis positions, for a Z position,
to extract the RMS values.';
uiwait(msgbox(step1));
fullfile_meshname=main2(); %Algorithm that extracts rms values, from X and Y
positions for one Z pos, and their coordinates, mm.
mesh=load(fullfile_meshname);
coordinates=flip(mesh.coordinates);
mesh=flip(mesh.mesh);%%Allocating data as Justyna suggested
[nn, ~]=size(coordinates); %nn - x values / mm - y values
%Number of rows and collumns must be the same
%Preallocating for saving coordinates
xsteps=zeros(1,nn);
ysteps=zeros(1,nn);
for i=1:1:nn
    auxx=coordinates{i,1};
    auxy=coordinates{1,i};
    xsteps(1,i)=auxx(1);
    ysteps(1,i)=auxy(2);
end
figure();
heatmap(ysteps,xsteps,mesh,...
    'Title','Intensity Map [mv]',...
    'Colormap',parula,...
    'GridVisible','off',...
    'FontColor','black',...
    'FontName','Arial',...
    'FontSize',7,...
    'XLabel','y-position [mm]',...
    'YLabel','x-position [mm]');

% --- Executes on button press in pb_selectfiles.
function pb_selectfiles_Callback(hObject, eventdata, handles)
% hObject    handle to pb_selectfiles (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%Displaying messages to the user

step1='Select .mat file containing, X and Y values, and their coordinates';
uiwait(msgbox(step1));
[filename, pathname, ~]=uigetfile('*.mat');
fullfile_meshname=fullfile(pathname, filename);
if isnumeric(pathname)
    uiwait(msgbox('Data Import Action Cancelled'));
else

```

```

mesh=load(fullfile_meshname);
coordinates=flip(mesh.coordinates);
mesh=flip(mesh.mesh);%%Allocating data as Justyna suggested
[nn, ~]=size(coordinates);%nn - x values / mm - y values
%Number of rows and collumns must be the same
%Preallocating for saving coordinates
xsteps=zeros(1,nn);
ysteps=zeros(1,nn);
%saving coordinates
for i=1:1:nn
    auxx=coordinates{i,1};
    auxy=coordinates{1,i};
    xsteps(1,i)=auxx(1);
    ysteps(1,i)=auxy(2);
end
figure();
heatmap(ysteps,xsteps,mesh,...
    'Title','Intensity Map [mv]',...
    'Colormap',parula,...
    'GridVisible','off',...
    'FontColor','black',...
    'FontName','Arial',...
    'FontSize',7,...
    'XLabel','y-position [mm]',...
    'YLabel','x-position [mm]');
end

% --- Executes on button press in pb_exit.
function pb_exit_Callback(hObject, eventdata, handles)
% hObject    handle to pb_exit (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
clc;close;

% --- Executes on button press in pb_intensity.
function pb_intensity_Callback(hObject, eventdata, handles)
% hObject    handle to pb_intensity (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[filename, pathname, ~]=uigetfile('*.txt');

%Avoiding error due to cancellation
if isnumeric(pathname)
    uiwait(msgbox('Data Import Action Cancelled'));
else
    fullfile_name=fullfile(pathname,filename);
    [header, data]=txtread(fullfile_name);
    data=data';%Vector
    Fs=header(8)*1e3; %Sampling time, Hz
    Ts=1/Fs; %Period, sec
    samples=header(10);%Number of samples

    %Calculating sampling time
    samplingtime=0:Ts:samples*Ts;
    samplingtime=samplingtime(1:end-1);

```

```

%Ploting Amplitude vs Sampling Time
figure();
plot(samplingtime,data,...
     'Color',[0.6350 0.0780 0.1840],...
     'LineWidth',0.25);
xlabel('Time [ms]');
ylabel('Amplitude [mv]')
title(filename,...
     'FontSize',10,...
     'FontWeight','bold',...
     'Color',[0.6350 0.0780 0.1840]);
end

```

Lectura de archivos *.tdt y *.txt – filereader.m

```

function [header, data, extension]=filereader(pathname, file_name)
%This function:
%   -Reads in .tdt & .txt files
%   -Returns:
%       -'header' information. E.g. ~nNo_of_samples~n in .txt files;
%       -Free field data
%       -Extension. E.g. .txt

%NOTES:
%This function has the potential of being implemented to plot
%   -Sampling time VS Amplitude (using .txt files)

%%
if isnumeric(pathname) %Avoiding error due to cancellation
    uiwait(msgbox('Import Data Aciton Cancelled'));
else
    %Reading file
    fullfile_name=fullfile(pathname,file_name);
    %Reading extension
    extension=file_name(end-3:end);
    switch extension
        case '.txt'
            %Reading *.txt files
            [header, data]=txtread(fullfile_name);
        case '.tdt'
            %Reading *.tdt files
            [header, data]=tdtread(fullfile_name);
            data=data';
        otherwise
    end
end
end

```

Código incorporado en filereader.m para poder llevar a cabo la lectura de archivos *.tdt – tdtread.m

```

function [header,data]=tdtread(fullfile_name)

```

```

%Algorithm to open and extract relevant data from .tdt files

%%
%opening file
fid=fopen(fullfile_name);
%Preallocating for header
header=zeros(11,1);

%Saving header information
for i=1:1:11
    tline=fgetl(fid);
    %Avoiding blank lines
    if isempty(tline) || i==1
        header(i)=0; %If file is empty or first line has no numeric data
        %continue;
    else
        idx=strfind(tline, '=');
        header(i,1)=str2double(tline(idx+1:end));
    end
end
%reading 37 col

data=fscanf(fid, '%f', [37 inf]);

%closing file
fclose('all');

```

Código incorporado en filereader.m para poder llevar a cabo la lectura de archivos *.txt – txtread.m

```

function [header, data]=txtread(fullfile_name)

%Algorithm to open and extract relevant data from .txt files

%%
%opening file
fid=fopen(fullfile_name);
%Preallocating for header
header=zeros(10,1);

%Saving header information
for i=1:1:10
    %read line from file
    tline=fgetl(fid);
    %Avoiding blank lines
    if isempty(tline)
        continue;
    else
        %Reading header information
        if contains(tline, ' ')
            idx=strfind(tline, ' ');
            header(i,1)=str2double(tline(idx:end));
        elseif contains(tline, '=')
            idx=strfind(tline, '=');

```

```

        aux=str2double(tline(idx+1:end));
        if ~isempty(aux) && ~isnan(aux)
            header(i,1)=aux;
        else
            header(i,1)=0;
        end
    else
        continue;
    end
end
end

%Reading data'
data=dlmread(fullfile_name, ' ',10,0);
%closing file
fclose(fid);

```

Extracción de los valores RMS de los encabezados de los 37 archivos que contienen la información de la sección transversal del haz de ultrasonido – main2.m

```

function fullfile_meshname=main2
%For this function you need to create a folder called 'RF' within the
%folder of this project. RF will contain all data of type:
%   -"RF-Z-20-0mm-Y0-X0.txt"
%
%It will return:
%   -An array with rms values (RMS-array) of all X and Y positions of one Z
position
%   -A cell with X, Y, and Z positions that corresponds with the RMS-array

%CONSIDERATIONS:
%1)The first file imported must correspond to the X0 and Y0 positions.
%Otherwise, errors might occur

%%
foldername='RF';
if ~exist(foldername, 'file')
    uiwait(msgbox('No folder found'));
else
    %uiwait(msgbox('Select the first file of a Z-axis position'));
    [filename, pathname, ~]=uigetfile('*.txt');
    if isnumeric(pathname)
        uiwait(msgbox('Data Import Action Cancelled'));
    else
        filename_old=filename;
        %
        datatype='RF';
        %
        zaxispos=20;
        %
        ypoint=0;
        %
        xpoint=0;
        %
        zlim=47; ylim=36; xlim=36; %Potential for allowing user edit
limits

```

```

%Preallocating
mesh=zeros(xlim+1,ylim+1);
coordinates=cell(xlim+1,ylim+1);
idxx=strfind(filename,'X0');
idxy=strfind(filename,'Y0');
%idxz=strfind(filename,'Z');
msgbox('Extracting RMS values might take some time.');
```

```

    for y=0:1:ylim
        ypos=strcat('Y',num2str(y));
        if y>10
            if x==36 %#ok<NODEF> %check this line
                filename=strcat(filename(1:idxy-
1),ypos,filename(idxy+2:end));
            else
                filename=strcat(filename(1:idxy-
1),ypos,filename(idxy+3:end));
            end
        else
            filename=strcat(filename(1:idxy-
1),ypos,filename(idxy+2:end));
        end
        %filename(idxy:idxy+1)=ypos;
        for x=0:1:xlim
            xpos=strcat('X',num2str(x));
            if x>10
                if y<10
                    filename=strcat(filename(1:idxx-
1),xpos,filename(idxx+3:end));
                else
                    filename=strcat(filename(1:idxx),xpos,filename(idxx+4:end));
                end
            else
                if y<10
                    filename=strcat(filename(1:idxx-
1),xpos,filename(idxx+2:end));
                else
                    filename=strcat(filename(1:idxx),xpos,filename(idxx+3:end));
                end
            end
            [header,~,~]=filereader(pathname,filename);
            mesh(x+1,y+1)=header(7);
            coordinates(x+1,y+1)={[header(3) header(4)
header(5)]}; %mm
        end
        filename=filename_old;
    end
%Saving data into a struct
MESH.mesh=mesh;
MESH.coordinates=coordinates;

%Interacting with user to save; the x-axis and y-axis rms
%values of one Z-axis position; and the x, y, z coordinates
%(in mm)

uiwait(msgbox('Save data as a new file'));
[filemeshname, pathmesh]=uiputfile('*.mat');
```

```
%           while isnumeric(pathmesh)
%             uiwait(msgbox('Data Save Action Cancelled'));
%           end
if isnumeric(pathmesh)
    uiwait(msgbox('Data Save Action Cancelled'));
else
    fullfile_meshname=fullfile(pathmesh, filemeshname);
    save(fullfile_meshname, '-struct', 'MESH');
end
end
end
```

Bibliografía

[1] Atkinson, P., y Wells, PNT., (1977) 'Pulse-Doppler Ultrasound and Its Clinical Application' *The Yale Journal of Biology and Medicine*, 50, pp. 367-373. Disponible desde: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2595531/pdf/yjbm00139-0045.pdf> (acceso: 14 abril 2020).

[2] Baker, AC., Berg, AM., Sahin, A., Tijotta, JN., (1995) 'The nonlinear pressure field of plane rectangular apertures: experimental and theoretical results.' *J Acoust Soc Am*, 92, pp. 3510-3517.

[3] Beaver, WL., (1974) 'Sonic near fields of a pulsed Aston radiator.' *J Acoust Soc Am*, 56, pp. 1043-1048.

[4] Brandestini, M., (1978) 'Topoflow—A digital full range Doppler velocity meter' *IEEE Trans. Sonics Ultrasonics*. SU-25, pp. 287-193.

[5] Brody, WR., and Meindl, JD., (1974) 'Theoretical Analysis of the CW Doppler Ultrasonic Flowmeter' *IEEE Transactions on Biomedical Engineering, BME-21 (3)*, pp. 183-192.

[6] Cahill, MD., Baker, AC., (1997) 'Increased off-axis energy deposition due to diffraction and nonlinear propagation of ultrasound from rectangular sources' *J Acoust Soc Am*, 102, pp. 199-203.

[7] Candy, JC., and Temes GC., (eds) (1987) 'Oversampling Methods for A/D and D/A Conversion', *IEEE Transactions on Circuits and Systems*. Nueva York: IEEE Press.

[8] Chung, E., (2019) 'Exchange visit to the Nihon Kohden Ogino Memorial Laboratory in Tokyo, Japan, July 2019'. Leicester. Disponible

desde: https://le.ac.uk/~media/uol/docs/research-institutes/lpmi/iax-2019/emma-chung-brain_tv_nk_iax_visit.pdf?la=en (acceso: 7 abril 2020).

[9] Chung, E., (eds) (2018) 'Medical Physics and Clinical Engineering Scientific Support: Bringing Technology to the NHS' Leicester. Disponible desde: <https://www.leicestershospitals.nhs.uk/EasySiteWeb/GatewayLink.aspx?allId=69576> (acceso: 7 abril 2020).

[10] Chung, E., Banahan, C., Minhas, J., and Lecchini-Visintini, A. (eds) (2018) 'Doppler Techniques for Medical Applications [online]' Leicester. Disponible desde: Carpeta Privada de la Investigación. (Acceso: 31 enero 2019).

[11] Duck, FA., (1981) 'The pulsed ultrasonic field' in Moores, BM., Parker, RP., Pullan, BR. (eds.) *Physical Aspects of Medical Imaging*. Chichester: Wiley, pp. 97-108.

[12] Duck, FA., Baker, AC., Starritt, HC., (1998) *Ultrasound in Medicine* (1st ed.). London: Institute of Physics Publishing.

[13] Duerinckx, AJ., (1981) 'Modelling wavefronts from acoustic phased arrays by computer' *IEEE Trans Biomed Eng BME*, 28, pp. 221-234.

[14] Evans, DE., and McDicken, WN. (2000) 'Doppler Ultrasound: Physics, Instrumentation and Signal Processing' (2nd ed.). Wiley.

[15] Gabor, D., (1946) 'Theory of Communication' *J. IEEE*, 93 (3), pp. 429-457.

[16] Giordano, NJ., (2010) 'College Physics. Reasoning and Relationships', Julet, m. (Ed) *Sound*. California, USA: Brooks/Cole, pp. 420.

[17] Hakimi, R., Alexandrov, AV., and Garami, Z., (2020) 'Neuro-Ultrasonography', *Neurologic Clinics* 83 (1), pp. 215.

[18] Healey, AJ., Leeman, S., Weight, JP., (1997) 'Space-time imaging of transient ultrasound fields.' *Int. J. Imaging Syst. Technol.*, 8 (1), pp. 45-51.

[19] Hoeks, APG., (1982) 'On the development of multi-gate pulsed Doppler system with serial data processing' Rijksuniversiteit Limburg: Países Bajos. Disponible desde: <https://cris.maastrichtuniversity.nl/ws/portalfiles/portal/896286/guid-3449c64a-f597-46eb-b2d5-e6ffb28ae825-ASSET1.0.pdf> (acceso: 22 abril 2020).

[20] Hoeks, APG., Brands, PJ., Arts, TGJ., y Reneman, RS., (1994) 'Subsample Volume Processing of Doppler Ultrasound Signals' *Ultrasound in Med. & Biol.*, 20 (9), pp. 953-965.

[21] Hoeks, APG., Reneman, RS., and Peronneau, PA., (1981) 'A Multigate Pulsed Doppler System with Serial Data Processing' *IEEE Transactions on Sonics and Ultrasonics*, SU-29 (4), pp. 242-243.

[22] Hoeks, APG., Reneman, RS., y Peronneau (1981) 'A multi-gate pulsed Doppler System with serial data processing.' *IEEE Trans. Sonics Ultrasonics*. SU-28, pp. 242-247.

[23] Jensen, JA., Svendsen, NB., (1992) 'Calculation of pressure fields from arbitrarily shaped, apodized, and excited ultrasound transducers.' *IEEE Trans Ultrason Ferroelec Freq Contr*, 39, pp. 262-267.

[24] John-Keeton, PI., (1997) *Modern Digital Signal Processing Techniques Applied to Doppler Ultrasound*. PhD Thesis. University of Leicester.

[25] Kino, GS., (1987) 'Acoustic Waves: Devices, Imaging, and Analog Signal Processing'. Prentice-Hall, Englewood Cliffs, NJ.

[26] Kinsler, LE., Frey, AR., Coppens, AB., and Sanders, JV., (1962) *Fundamentals of Acoustics*. 2nd edn New York, etc: John Wiley & Sons.

[27] Kossoff, G., (1966) 'The Effects of Backing and Matching on the Performance of Piezoelectric Ceramic Transducers' *IEEE Transactions on Sonics and Ultrasonics*. 13 (1), pp. 20-30.

[28] Kucewicz, JC., (2004) *Tissue Pulsatility Imaging: Ultrasonic Measurement of Strain Due to Perfusion*. PhD Dissertation. University of Washington.

[29] McLeod, FC., (1967) 'A directional Doppler flowmeter' *Digest of the 7th Int. Cons. on Medical and Biological Engineering, Sweden*. pp. 123.

[30] Nihon Kohden (2019) 'Key Facts [online]' Disponible desde: <https://www.nihonkohden.com/company/facts.html> (Acceso: 7 abril 2020).

[31] Nihon Kohden (s/f) 'Our Philosophy. The intersection of medicine and engineering, [online]' Disponible desde: <https://www.nihonkohden.com/philosophy/index.html> (Acceso: 7 abril 2020).

[32] Oppenheim A. and Schafer R., (eds) (1998)'Discrete-Time Signal Processing'. Nueva Jersey: Prentice Hall.

[33] Peronneau, FA., Bournat, JP., Bugon, A., Barbet, A., and Xhaard, M., (1974) 'Theoretical and practical aspects of pulsed Doppler flowmetry: real-time applications to the measure of instantaneous velocity profiles in vitro and in vivo' *Cardiovascular applications of Ultrasound*, pp. 66-84.

[34] Reneman, RS., Van Merode, T., Hick, P., Hoeks, APG., (1986) 'Cardiovascular applications of multi-gate pulsed doppler systems' *Ultrasound in Medicine & Biology*. 12(5) pp. 357-370.

[35] Reneman, RS., van Merode, T., Hick, P., y Hoeks, APG., (1985) 'Flow velocity patterns in and distensibility of the carotid artery bulb in volunteers of varying age' *Circulation*, 71, pp. 500-509.

[36] RosarioVanTulpe, y Teflotax (modificó) (2010) 'SkullSchaedelSeitlich1.png' Dominio público, disponible desde: <https://commons.wikimedia.org/w/index.php?curid=9105493> (accesado el 31 de Marzo de 2020).

[37] Shung, KK., and Zippuro, M., (1996) 'Ultrasonic transducers and arrays' *IEEE Engineering in Medicine and Biology Magazine*. 15 (6), pp. 20-30.

[38] Sillicon Laboratories Inc, (2013) 'Improving ADC resolution by Oversampling and Averaging', pp. 1-2 y 6-8.

[39] Skidmore, G., (2010) 'Actors Sylvester Stallone and Bruce Willis on the Expendables panel at the 2010 San Diego Comic Con in San Diego, California', *Flickr: Sylvester Stallone & Bruce Willis*, 22 de Julio. Disponible desde: <https://www.flickr.com/photos/gageskidmore/4840665852/> (Accesado el 31 de marzo de 2020).

[40] Spencer Technologies (2016) 'Power M-mode Transcranial Doppler with HDO™ Digital Processing', disponible desde: <https://www.spencertechnologies.com> (accesado el 27 de marzo de 2020).

[41] Sundararajan, D., (2016) *Discrete Wavelet Transform: A signal processing approach*. John Wiley & Sons.

[42] Tan, L., and Jiang, J. (2013) 'Digital Signal Processing' (2nd ed). EEUU: Academic Press. Disponible en: <https://www.eetimes.com/multirate-dsp-part-1-upsampling-and-downsampling/> (Acceso: 10 abril 2020).

[43] Toman, K., (1984) 'Doppler and the Doppler Effect' Nueva York: Rome Air Development Center.

[44] Turner, P. (eds) (s/f) 'BainTV graphical User Interface (GUI). User Instructions' Leicester. Disponible desde: carpeta Privada de la Investigación. (Acceso: 8 abril 2020).

[45] University of Erlangen (2014) 'R&D Towards Acoustic Particle Detection' [presentación en Power Point]. Disponible en: https://ecap.nat.fau.de/wp-content/uploads/2017/03/UliKatz_pisa04.pdf (Acceso: 20 mayo 2020).

[46] Wikimedia Commons contributors (2017) 'Archivo:Optical-cavity1.png' *Wikimedia Commons, the free media repository*. Disponible desde: <https://commons.wikimedia.org/w/index.php?title=File:Optical-cavity1.png&oldid=263280539> (accesado 29 marzo 2020).

[47] Wille, SØ., (1979) Numerical models of arterial blood flow. Tesis, Instituto de Informatica, Universidad de Oslo, Noruega.