

Implementación de un sistema de adquisición para tomografía ultrasónica

Victor Gisbert Climent

Tutor: Jorge Gosálbez Castillo

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

Curso 2015-16

Valencia, 2 de septiembre de 2016

Resumen

En el presente trabajo de fin de grado se van a tratar los ensayos no destructivos (END) los cuales permiten conocer el estado interno de ciertos materiales sin alterar sus condiciones físicas, químicas o mecánicas. Existen muchos tipos de END en función del tipo de señal y técnica, pero en este trabajo nos centraremos en los END basados en ultrasonidos y más concretamente en el desarrollo de la técnica de tomografía ultrasónica. Dicha técnica requiere la toma de múltiples rayos ultrasónicos paralelos tomados desde diferentes ángulos, cada uno de estos ángulos genera una proyección que hay que combinar para reconstruir el estado interno del material. Por tanto, el objetivo primero de este trabajo será la realización de un sistema de toma masiva y ordenada de señales ultrasónicas, donde uno de los problemas será la automatización del acoplamiento físico entre el transductor y el material inspeccionado. El segundo objetivo consistirá en aplicar las técnicas de procesado tomográficas utilizando diferentes parámetros ultrasónicos (velocidad, energía, atenuaciones...). Por último, trataremos de validar esta técnica inspeccionando diferentes elementos y materiales : hormigón, vidrio y aluminio.

Resum

En el present treball de fi de grau es van a tractar els assajos no destructius (END) els quals permeten conèixer l'estat intern de certs materials sense alterar les seues condicions físiques, químiques o mecàniques. Hi ha molts tipus d'END en funció del tipus de senyal i tècnica, però en este treball ens centrarem en els END basats en ultrasons i més concretament en el desenvolupament de la tècnica de tomografia ultrasónica. Dita tècnica requerix la presa de múltiples rajos ultrasònics presos des de diferents angles, cada un d'estos angles genera una projecció que cal combinar per a reconstruir l'estat intern del material. Per tant, l'objectiu primer d'este treball serà la realització d'un sistema de presa massiva i ordenada de senyals ultrasòniques, on un dels problemes serà l'automatització de l'adaptament físic entre el transductor i el material inspeccionat. El segon objectiu consistirà a aplicar les tècniques de processat tomogràfiques utilitzant diferents paràmetres ultrasònics (velocitat, energia, atenuaciones...) . Finalment, tractarem de validar esta tècnica inspeccionant diversos elements i materials : formigó, vidre i alumini.

Abstract

In this paper, it will be treated nondestructive tests (NDT) which provide information about the internal state of certain materials without altering their physical, chemical or mechanical conditions. There are many types of NDT depending on the type of signal and technique, but in this paper we focus on NDT based on ultrasound and more specifically in the development of ultrasonic tomography technique. This technique requires measuring multiple parallel ultrasonic rays from different angles, each of these angles generates a projection which must be combined to reconstruct the internal state of the material. Therefore, the first objective of this study will be the embodiment of a system for taking massive and ordered ultrasonic signals, where one of the problems is the automation of physical coupling between the transducer and the inspected probe. The second objective will be to apply ultrasonic tomographic techniques over different processed parameters (speed, energy, fades ...). Finally, we try to validate this technique inspecting different elements and materials: concrete, glass and aluminum.

Índice

Capítulo 1. Introducción	3
1.1 Objetivos, plan de trabajo.....	3
1.2 Plan de trabajo.....	3
1.3 Ultrasonidos	4
Capítulo 2. Tomografía Ultrasónica.....	5
2.1 Principios básicos.....	5
2.1.1 Propagación de ultrasonidos en medios homogéneos	6
2.1.2 Propagación de ultrasonidos en medios inhomogéneos	7
2.2 Tomografía: proyecciones.....	13
2.2.1 Tomografía basada en Trazado de Rayos.....	13
2.2.2 Tomografía ultrasónica de Difracción (TUD).....	16
2.2.3 Tomografía ultrasónica de Reflexión (TUR)	16
2.2.4 Elección del método	17
2.2.5 Obtención de proyecciones	17
2.3 Tomografía: algoritmos de reconstrucción.....	21
2.3.1 Métodos de transformación.....	22
2.3.2 Algoritmo de Retroproyección Filtrada	22
2.3.3 Elección del algoritmo de reconstrucción	25
Capítulo 3. Medidas preliminares: tomografía por contacto.....	27
3.1 Método de trabajo y equipamiento empleado	27
3.2 Preparación de los datos.....	32
3.2.1 Generación de las posiciones del transmisor y receptor para cada proyección.....	32
3.2.2 Asociación entre posición del transmisor y receptor y medida asociada	33
3.2.3 Asociación inversa	34
3.2.4 Cálculo de parámetros	35
3.2.5 Mejora de las proyecciones	36
3.2.6 Algoritmo de reconstrucción y concepto de sinograma	36
3.3 Resultados	39
Capítulo 4. Desarrollo Experimental	41
4.1 Introducción	41
4.1.1 Técnica basada en inmersión.....	41
4.1.2 Técnica basada en transductores por contacto con aire.....	41
4.1.3 Técnica basada en transductores por contacto directo	43
4.2 Desarrollo de sistema de acoplamiento.....	45

4.3	Desarrollo de adquisición de forma sincronizada	50
4.4	Resultados preliminares con el sistema de toma de medidas masivo	58
4.5	Toma de medidas	62
4.5.1	Cambio de concepto de toma de medidas	62
4.5.2	Señal utilizada	64
4.5.3	Equipo al completo.....	64
4.5.4	Simulación previa.....	65
4.6	Resultados	69
Capítulo 5.	Conclusiones y líneas futuras	74

Capítulo 1. Introducción

La finalidad del presente trabajo de fin de grado es el diseño e implementación de un sistema para realizar una toma masiva de medidas, y así poder llegar a obtener tomografías ultrasónicas en diferentes tipos de materiales mediante ensayos no destructivos. Este sistema estará aplicado a diferentes probetas para saber cual es su estado interno.

Ya existen diferentes estudios sobre tomografías ultrasónicas, el problema es que, normalmente, el número de medidas suele ser pequeño y no podemos hacer una reconstrucción del todo fiable de la estructura interna del objeto bajo estudio.

Por tanto, se diseñará e implementará un sistema automático de toma masiva de medidas para poder recoger la suficiente información del elemento en estudio y así poder realizar una tomografía fiable.

1.1 Objetivos, plan de trabajo

Al inicio de este trabajo, se han presentado dos objetivos principales. El primero consiste en diseñar el sistema de medida automático mediante transductores ultrasónicos un encoder de posición utilizando la plataforma Arduino y Matlab. El segundo objetivo principal será implementar dicho sistema y aplicarlo a probetas de cemento, cuyo fin será poder reconocer la estructura interna del elemento que se va a estudiar.

Estos objetivos se dividen en las siguientes tareas:

-Diseñar el sistema automatizado mediante transductores ultrasónicos en contacto con la probeta, y un encoder de posición para guardar cada medida.

- Implementar dicho sistema, y adaptarlo a la probeta de hormigón.

- Realizar una herramienta software sencilla para recoger los datos de medida del encoder de posición mediante Arduino y Matlab.

-Aplicar nuestro sistema de medida a probeta de hierro u otro tipo de material.

1.2 Plan de trabajo

A grandes rasgos, la realización de este trabajo de fin de grado se ha dividido en diferentes grupos:

- Primera toma de medidas con transductores ultrasónicos en contacto con la probeta a través de gel conductor,
- Diseño de una herramienta en Matlab para hacer una reconstrucción de esas medidas,
- Implementación del sistema automatizado y herramienta software para la recogida de las medidas mediante el encoder de posición mediante Arduino y adquisición mediante HanyScope.

1.3 Ultrasonidos

Los ultrasonidos son ondas mecánicas, es decir no ionizantes, cuya frecuencia está fuera de la banda audible de las personas, ya que la banda audible del ser humano está comprendida entre 20 Hz y 20000 Hz. A partir de 20000 Hz cualquier sonido es denominado como ultrasonido o señal ultrasónica.

Como se ha citado anteriormente, los ultrasonidos son sonidos de alta frecuencia, y la acústica, ciencia del sonido, estudia el fenómeno de las vibraciones mecánicas y de su propagación en medios sólidos, líquidos o gaseosos. En el vacío no se propaga, puesto que se necesita una vibración mecánica de las moléculas de dicho medio, lo contrario a las ondas electromagnéticas.

Las técnicas de evaluación por ultrasonidos son aquellas que se basan en la transmisión y recepción de señales ultrasónicas para evaluar las propiedades físicas o el estado de un material bajo estudio. Estas señales se propagan por el material produciendo vibraciones mecánicas en sus partículas pero de forma tan pequeña que no dañan el material.

Una de las técnicas de evaluación por ultrasonidos más utilizada en la actualidad es la ecografía, la cual nos sirve para examinar al bebé en una mujer embarazada, conocer el estado de los órganos internos si están hinchados o infectados, incluso para poder examinar el cerebro de un niño pequeño. También se utiliza para guiar biopsias y diagnosticar condiciones del corazón, como puede ser el daño causado a este por un ataque.

En el siguiente apartado en el que se explica la tomografía ultrasónica, veremos más a fondo este tipo de ondas y conoceremos sus características, tanto en medios homogéneos como cuando no lo son y qué efectos produce esto en las ondas que se propagan en el objeto bajo estudio.

Capítulo 2. Tomografía Ultrasonica

La palabra tomografía se utiliza para referirse al procedimiento de obtención de medidas o proyecciones sobre un objeto, que permite, tras el adecuado proceso de reconstrucción, obtener una imagen que representa alguna de las propiedades físicas del objeto en una determinada sección plana. Para ello se “iluminará” el objeto desde diferentes direcciones de propagación con algún tipo de fuente de energía, que en este caso serán ultrasonidos.

Para poder obtener proyecciones, debemos saber cómo se propagan los ultrasonidos en cada material, ya que para cada material tendrá una serie de parámetros diferentes que habrá que tener en cuenta. Por tanto, en el siguiente apartado vamos a ver algunos principios básicos de cómo se propagan estas señales en medios homogéneos, para luego ver como en materiales inhomogéneos se producen diferentes efectos, como reflexión, refracción, difracción y dispersión. Esto nos servirá de ayuda para poder comprender diferentes métodos de tomografías ultrasonicas.

2.1 Principios básicos

Tal y como ya se ha explicado, los ultrasonidos, como ondas acústicas que son, son oscilaciones mecánicas que tienen lugar en el tiempo y en el espacio, y a diferencia de las ondas electromagnéticas, necesitan un medio material para propagarse, es decir, las partículas son desplazadas de su posición inicial dando lugar a una perturbación o vibración que se propaga por el material.

Para explicar esto, pensemos en un grupo de partículas que están equiespaciadas con distancia Δx . En el instante inicial t_0 , se perturba la primera partícula y se mueve de la posición de equilibrio pero de forma amortiguada, es decir, vuelve a la posición de equilibrio después de su variación. Debido al “choque” de partículas, el movimiento de la primera partícula se transfiere a la segunda, pero si ningún cambio en esta, simplemente un retraso Δt .

Este movimiento se traspa al resto de partículas, y por tanto se crea una perturbación que se propaga por el medio a razón de:

$$c = \frac{\Delta x}{\Delta t} \quad (1)$$

Así, sabremos la velocidad del ultrasonido en el medio.

Para saber la elongación, o variación de posición desde la posición de equilibrio podemos llegar a una función $u(x, t)$, siendo x la partícula que queremos conocer, y t el instante de tiempo en el cual queremos conocer la posición de dicha partícula. Si utilizásemos señales de tipo sinusoidal, la función queda como:

$$u(x, t) = A \sin\left(\frac{2\pi t}{T} - \frac{2\pi x}{\lambda}\right) \quad (2)$$

Donde A es la amplitud de la onda, λ es la longitud de onda, o la distancia entre dos partículas consecutivas que vibran en fase, y T el periodo de oscilación que es el tiempo en el que se completa un ciclo de movimiento entero.

Se sabe que los ultrasonidos son sonidos a partir de 20KHz, por tanto desde el generador de señales seleccionaremos el valor de la amplitud y de la frecuencia, y a partir de ahí se conocerá el valor de λ , ya que:

$$\lambda = \frac{c}{f} \quad (3)$$

Otro parámetro que no se ha comentado y es importante es el valor de la presión sonora. En los objetos a estudiar habrá puntos donde hay un mayor número de partículas y la presión sonora supera su valor normal, mientras que en zonas más dilatadas (menor número de partículas) la presión sonora es menor. De esta forma el valor de la presión sonora p está estrechamente relacionado con el valor de u de la elongación de las partículas en torno a su posición de equilibrio, mediante la expresión siguiente:

$$p = \rho c \omega u \quad (4)$$

Siendo ω la pulsación, ρ la densidad y c la velocidad en el medio. Y con esto llegamos a un parámetro importante, ya que el producto de pc es la llamada impedancia acústica característica Z , e indica la resistencia que presenta el material al paso de las ondas sonoras, es decir, esto fijará las pérdidas por reflexión y difracción. Esta impedancia nos dará información para saber cómo resolver algunos problemas.

Otro aspecto importante es saber cómo se propagan las señales ultrasónicas en medios homogéneos, para luego ver qué ocurre en medios inhomogéneos, [1].

2.1.1 Propagación de ultrasonidos en medios homogéneos

En nuestro caso vamos a ver este apartado como una forma introductoria, ya que el fin de este proyecto es trabajar con materiales que serán, normalmente, inhomogéneos. Por tanto, vamos a ver una pincelada de cómo se propagan los ultrasonidos en medios homogéneos para luego ver los parámetros o efectos que produce que haya discontinuidades en el objeto.

Para comenzar, las perturbaciones sonoras que se producen en un punto de un medio elástico se propagan a través de él con una velocidad c , que depende de la densidad ρ del medio y de su módulo de elasticidad E , de acuerdo con la ecuación:

$$c = \sqrt{\frac{E}{\rho}} \quad (5)$$

Y a partir de esto, podemos calcular la impedancia acústica Z que viene dada por el producto de la velocidad c y la densidad ρ :

$$Z = \rho c \quad (6)$$

Como resultado de la perturbación, la presión varía con respecto a un valor medio. La diferencia entre el valor instantáneo y el valor medio se denomina presión acústica, p . La intensidad I de la onda es la potencia por unidad de superficie medida en W/m^2 . Por lo tanto, podemos llegar a la conclusión de que la impedancia acústica Z , la presión acústica p y la intensidad I están relacionadas en la misma ecuación:

$$I = \frac{p^2}{Z} = \frac{p^2}{\rho c} \quad (7)$$

Al propagarse la radiación por un medio homogéneo, su intensidad sufre una atenuación exponencial de acuerdo con la ecuación:

$$I = I_0 e^{-2\alpha x} \quad (8)$$

En la que I_0 es la radiación incidente, α el coeficiente de atenuación y x es la distancia recorrida.

Esto se cumple, mientras que el ultrasonido se propague en el mismo medio, es decir, en nuestro caso (utilizando probetas) no se dará esta situación ya que el objeto bajo estudio no está formado por un solo medio, sino que tendrá diferentes heterogeneidades en medio. Si la probeta fuera totalmente homogénea por dentro enviaríamos una señal y tendríamos que recibirla igual, pero atenuada.

Veamos ahora qué sucede cuando los ultrasonidos pasan de un medio a otro y cómo reaccionan a estos cambios.

2.1.2 Propagación de ultrasonidos en medios inhomogéneos

En [1] podemos encontrar los diferentes parámetros que aparecen cuando hay inhomogeneidades (cambios de medio) en los materiales, para luego poder hacer una reconstrucción a partir de ellos.

2.1.2.1 Reflexión y refracción

En el momento en el que el ultrasonido incide de forma oblicua en una superficie plana con un ángulo de incidencia α_i con respecto a la perpendicular, se produce una onda reflejada y una onda transmitida o refractada, ya que no se transmite en la misma dirección de propagación de la onda incidente si no que tiene otra dirección. El ángulo de la onda reflejada, α_r , y el ángulo con el que se propaga la onda refractada, α_t , dependen del ángulo de incidencia y de la velocidad de los ultrasonidos en los dos medios, y pueden ser determinados de la siguiente forma:

$$\alpha_r = \alpha_i \quad (9)$$

$$\frac{\sin \alpha_i}{\sin \alpha_t} = \frac{c_1}{c_2} \quad (10)$$

Siendo c_1 y c_2 las velocidades de los ultrasonidos en los medios por los que se propaga la señal incidente y la señal refractada respectivamente.

La ecuación superior se conoce como la ley de Snell y aunque primero fue establecida por el propio Snell para óptica, resulta ser válida para todo tipo de ondas.

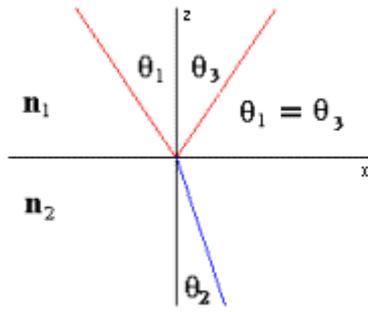


Fig. 1: Ley de Snell

Aquí se puede apreciar mejor el concepto explicado anteriormente. Pero aquí se ha relacionado mediante los índices de refracción en cada medio, n_1 y n_2 , pero es lo mismo que crear la relación con las velocidades en los medios ya que:

$$n = \frac{c}{v} \quad (11)$$

Siendo c la velocidad de la luz en el vacío y v la velocidad de la luz en dicho medio.

No obstante, existe una diferencia entre los fenómenos ópticos y las ondas acústicas, y es que en la acústica existe lo que se conoce como conversión de modos. Cuando una onda ultrasónica de tipo longitudinal incide sobre una superficie de separación entre dos medios, uno de los cuales es un sólido, se crean determinadas ocasiones ondas transversales. Como las ondas transversales se propagan con una velocidad diferente que las longitudinales, los ángulos de reflexión y refracción son diferentes, pudiéndose calcular con la ley de Snell:

$$\frac{\sin \alpha_i}{\sin \alpha_k} = \frac{c_i}{c_k} \quad (12)$$

Siendo c_i y c_k las velocidades de los ultrasonidos en los medios donde se están propagando.

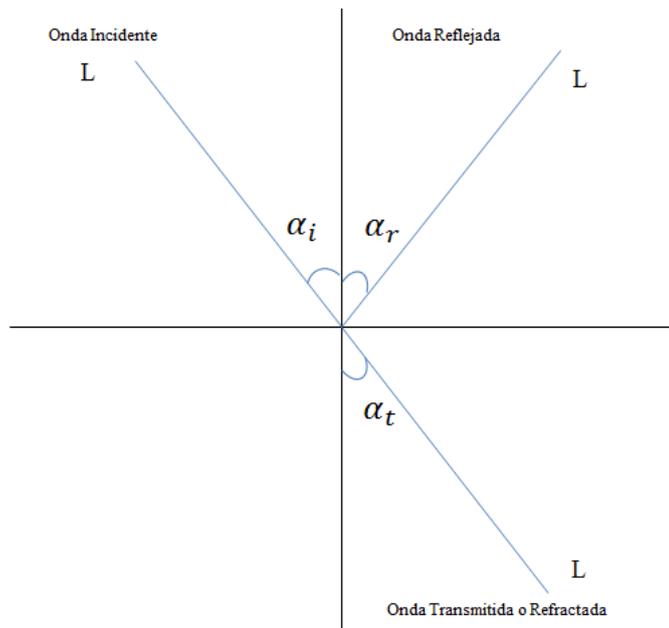


Fig.2 : Reflexión y refracción con ondas longitudinales

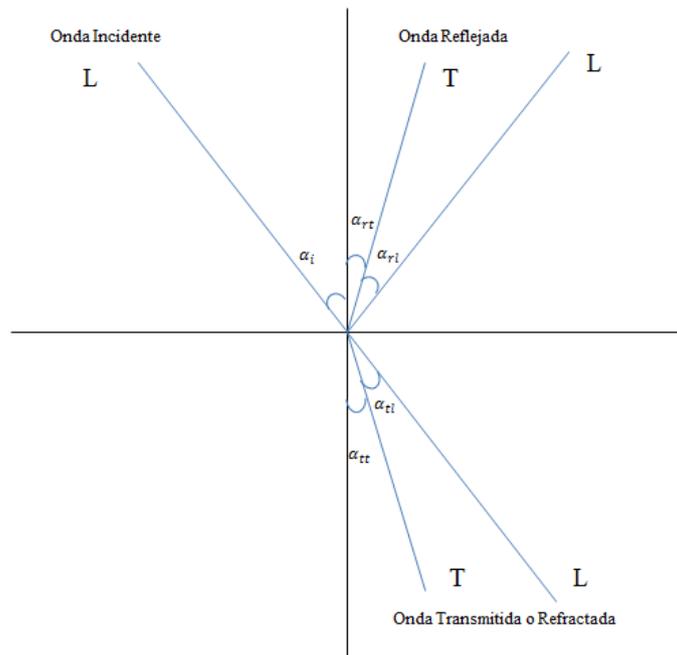


Fig. 3: Reflexión y refracción con ondas transversales.

El hecho de que las ondas cambien de medio no solo modifica la dirección de propagación de la señal, sino que varían la amplitud sonora. Esto es debido a que el coeficiente de reflexión R y el coeficiente de transmisión T establecen una relación entre la amplitud de presión sonora de la onda que incide y la amplitud de presión sonora de la onda reflejada o refractada:

$$R = \frac{P_R}{P_i} \quad (13)$$

$$T = \frac{P_t}{P_i} \quad (14)$$

Estas ecuaciones pueden variar. En el caso de que la separación entre los dos medios sea plana, estos coeficientes se determinarían para distintas situaciones:

1. En el caso en que los dos medios sean líquidos, en los que sólo se propagan ondas longitudinales, estos coeficientes son más fáciles de calcular:

$$R = \frac{Z_2 \cos \alpha_i - Z_1 \cos \alpha_t}{Z_2 \cos \alpha_i + Z_1 \cos \alpha_t} \quad (15)$$

$$T = \frac{2Z_2 \cos \alpha_i}{Z_2 \cos \alpha_i + Z_1 \cos \alpha_t} \quad (16)$$

En este caso, α_i y α_t son los ángulos de incidencia y de transmisión de las ondas longitudinales, y Z_1 y Z_2 las impedancias acústicas características de los dos medios, que vienen dadas por la siguiente ecuación:

$$Z = \frac{p}{v} = \rho c \quad (17)$$

Que es la impedancia acústica citada en el apartado 1.4.1.

2. Si uno de los dos medios es un sólido, el fenómeno de conversión de modos complica las expresiones, y han de diferenciarse en dos casos. Vamos a definir una serie de variables para luego llegar a las expresiones que nos interesan:

α : Ángulo correspondiente a la onda longitudinal en medio líquido.

α_1 : Ángulo correspondiente a la onda longitudinal en medio sólido.

α_t : Ángulo correspondiente a la onda transversal en medio sólido.

ρ_l : Densidad del medio líquido.

ρ_s : Densidad del medio sólido.

c : Velocidad de propagación de las ondas longitudinales en medio líquido.

c_l : Velocidad de propagación de las ondas longitudinales en medio sólido.

c_t : Velocidad de propagación de las ondas transversales en medio sólido.

Ahora diferenciaremos dos casos, dependiendo de si el medio 1 y el medio 2 son sólidos o líquidos:

1. Si el medio 1 es líquido y el medio 2 es sólido, para una onda longitudinal incidente en el medio líquido, se reflejará una onda longitudinal y se transmitirán una longitudinal y otra transversal con los siguientes coeficientes de transmisión y reflexión:

$$R = \frac{1}{N} \left[\left(\frac{ct}{cl} \right)^2 \sin 2\alpha_l \sin 2\alpha_t + \cos 2\alpha_t^2 - \frac{\rho_l * c}{\rho_s * cl * \frac{\cos \alpha_l}{\cos \alpha}} \right] \quad (18)$$

En la cual, N es:

$$N = \left(\frac{ct}{cl} \right)^2 \sin 2\alpha_l \sin 2\alpha_t + \cos 2\alpha_t^2 + \frac{\rho_l * c}{\rho_s * cl * \frac{\cos \alpha_l}{\cos \alpha}} \quad (19)$$

Y los coeficientes de transmisión, tanto de la onda longitudinal como de la onda transversal:

$$T_{ll} = \frac{2}{N} \cos 2\alpha_t \quad (20)$$

$$T_{lt} = - \frac{2}{N} \left(\frac{ct}{cl} \right)^2 \sin 2\alpha_l \quad (21)$$

2. Si el medio 1 es sólido y el medio 2 es líquido, para una onda longitudinal incidente en el medio sólido, se obtienen 2 ondas reflejadas (una longitudinal y una transversal) y una onda longitudinal transmitida. Dichas ondas se obtendrán a partir de sus coeficientes:

$$R_{ll} = \frac{Z_2 \cos \alpha_i - Z_1 \cos \alpha_t}{Z_2 \cos \alpha_i + Z_1 \cos \alpha_t} \quad (22)$$

$$R_{lt} = \frac{2}{N} \left(\frac{ct}{cl} \right)^2 \sin 2\alpha_l \cos 2\alpha_t \quad (23)$$

$$T_{ll} = \frac{2}{N} \frac{\rho_l * c * \cos \alpha_l \cos 2\alpha_t}{\rho_s * cl * \cos \alpha} \quad (24)$$

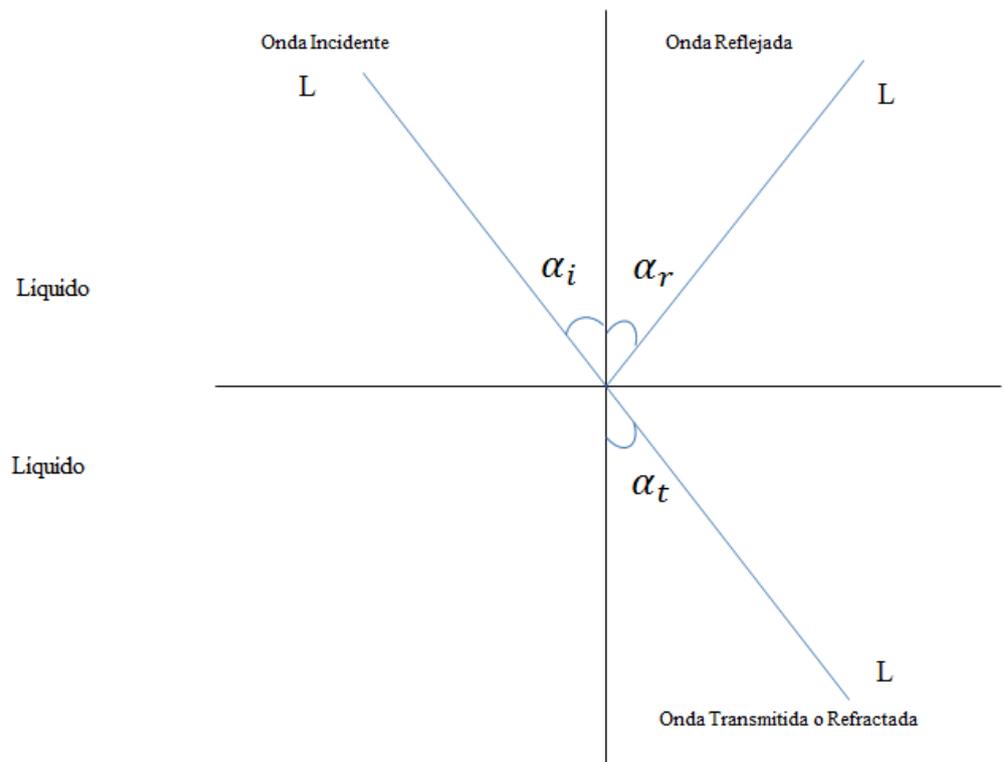


Fig.4 : Reflexión y refracción cuando los dos medios son líquidos.

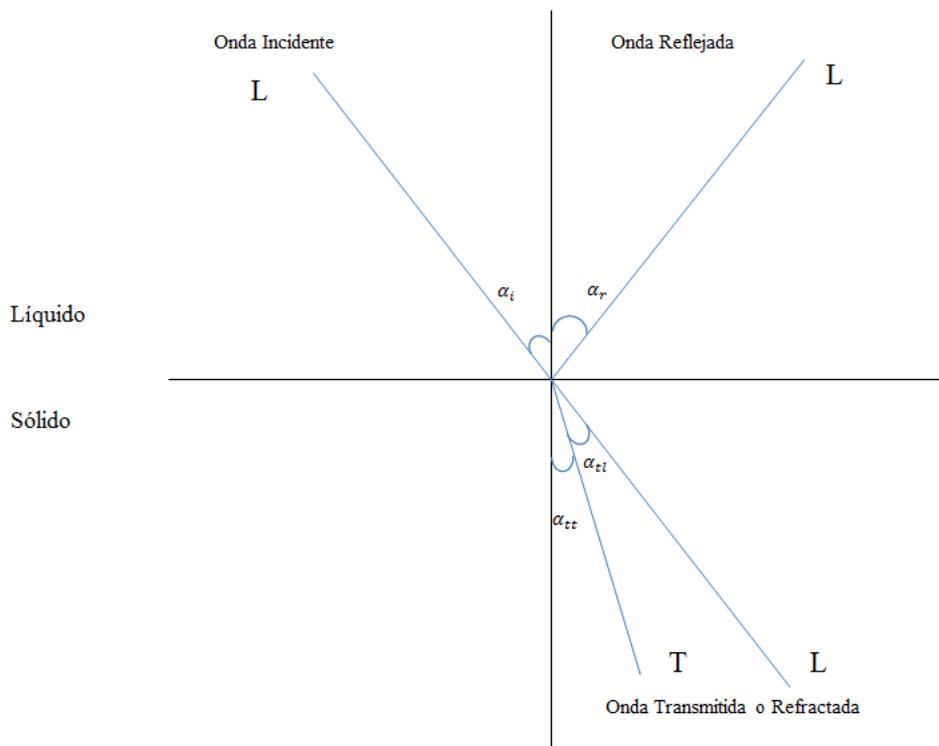


Fig. 5: Reflexión y refracción cuando el primer medio es líquido y el segundo es sólido.

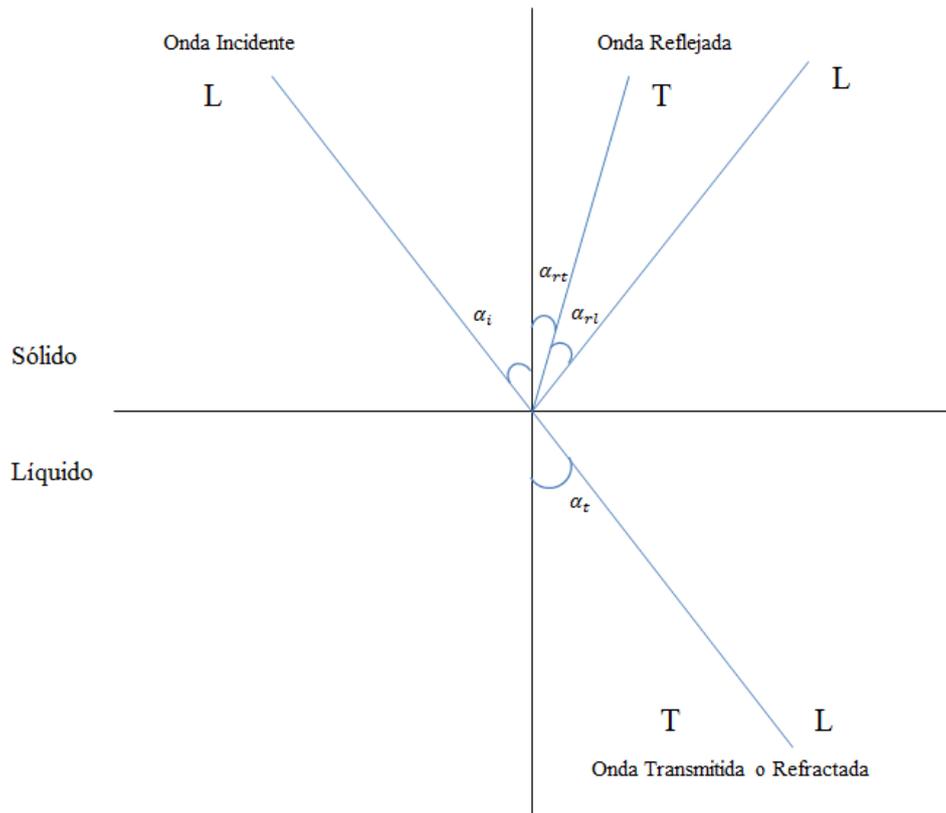


Fig. 6: Reflexión y refracción cuando el primer medio es sólido y el segundo es líquido.

2.1.2.2 Difracción y Dispersión

Cuando se produce un cambio de impedancias en el medio de propagación esto provoca que el fenómeno de reflexión sea muy grande, ya que se puede afirmar que este cambio de impedancias es como un obstáculo para la señal. Si una onda ultrasónica se encuentra en el camino un obstáculo mayor que su longitud de onda, se bloquea la onda y se produce una zona de sombra detrás del obstáculo.

Cuánto más pequeño el obstáculo comparado con la longitud de onda, la sombra será menor. Por otro lado, cuando el obstáculo tiene dimensiones comparables a la longitud de onda, se puede decir que los ultrasonidos bordean dicho obstáculo y llegan a la zona que antes era prohibida. Esto es conocido como difracción y se puede entender a partir del principio de Huygens, que establece que todo punto de un frente de ondas se puede considerar como un nuevo foco de ondas secundarias que se propagan en todas direcciones, formando la envolvente de todas las ondas secundarias el nuevo frente de onda.

Este fenómeno es conocido a veces como dispersión, y ocurre cuando existen muchos obstáculos o dispersores de tamaño reducido. Aunque se suele utilizar la palabra difracción cuando el objeto es grande, comparado con la longitud de onda, no existen diferencias reales entre ambos fenómenos, a no ser la diferencia de magnitud relativa de los efectos.

2.1.2.3 Atenuación

La reducción de amplitud sufrida por la señal ultrasónica debido a fenómenos de absorción, se denomina atenuación y puede ser debido a diferentes fenómenos físicos: se produce cuando:

1. Parte de energía se pierde en forma de calor al propagarse en el medio de transmisión, La cantidad de energía absorbida por el material depende de la naturaleza del medio y de la frecuencia de la señal y evidentemente puede ofrecer información de las propiedades de dicho medio.
2. La pérdida de energía durante el trayecto debido a la reflexión, refracción, difracción y dispersión, comentadas anteriormente, y que dependen tanto de la geometría y heterogeneidades del sistema así como de las propiedades físicas.

2.2 Tomografía: proyecciones

En primer lugar, se definirá una forma de obtener tomografías ultrasónicas de una forma general, tanto métodos para realizar las medidas, como la obtención de proyecciones y luego se pasará a ver el equipo empleado y como se han procesado esas medidas.

Para conseguir llegar a realizar una tomografía mediante un número limitado existen varios métodos para intentar cubrir toda la superficie del objeto bajo estudio. Se conocen 3 métodos para obtener tomografías:

-Tomografía Ultrasónica basada en Trazado de Rayos (TU-TR).

-Tomografía Ultrasónica de Difracción (TUD).

-Tomografía Ultrasónica por Reflexión (TUR).

Para el presente TFG, se ha utilizado el método basado en Trazado de Rayos, que no quiere decir que sea mejor que el resto, cada uno de estos realizará una reconstrucción a partir del parámetro que midan, difracción o reflexión. Se verá de forma detallada el utilizado en el TFG y de una forma más resumida los otros dos métodos.

2.2.1 Tomografía basada en Trazado de Rayos

Según [1], las ondas planas de extensión espacial infinita no existen en la naturaleza, en la realidad existen haces de sonido cuyas direcciones y secciones transversales pueden cambiar al atravesar un medio diferente al que se encontraba. Si el haz de sonido es muy direccional y la velocidad de propagación varía mucho en el medio, es conveniente pensar en rayos en vez de en ondas.

La tomografía convencional se basa en la teoría de rayos y en el concepto de propagación geométrica, ya que en 1972 Godfrey Newbold Hounsfield, comenzó a realizar tomografías utilizando rayos X para iluminar un objeto y medir qué energía llegaba al otro extremo. Estos rayos se propagan en línea recta, las proyecciones con las que reconstruiremos la imagen tomográfica, serán a partir de una serie de medidas que realizaremos. Cada una de ellas representa la integración del coeficiente de atenuación que presenta el objeto a lo largo de una línea recta, también conocida como integral de línea.

Por tanto, esta tomografía convencional se ha aplicado a otro tipo de fuentes de energía diferente, los ultrasonidos, ya que son más fáciles de generar y medir que los rayos-x y se puede hacer una reconstrucción a partir de un mapa de velocidades, aparte del mapa de atenuaciones, pero también aparecen una serie de inconvenientes, como los fenómenos de refracción y difracción en medios inhomogéneos.

Para este tipo de tomografía existen dos métodos de toma de medidas:

-Estructura basada en geometría de rayos paralelos equiespaciados, en la que obtenemos de forma consecutiva todos los rayos paralelos correspondientes a una dirección de propagación.

Para una determinada dirección de propagación habrá que mover el transmisor y el receptor para medir retardos, velocidad, amplitud o potencia correspondiente a cada rayo. Se repetirá este proceso en todas las direcciones de propagación existentes.

Con este método la obtención de proyecciones es inmediata, pues los retardos de cada rayo en una dirección constituyen la proyección en esa dirección.

- Receptor
- Transmisor

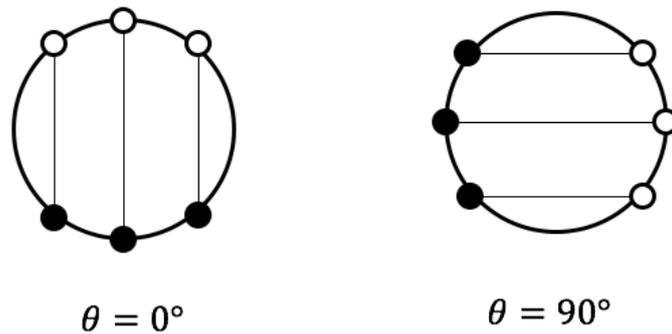


Fig. 7 : Proyecciones de 0° y 90°

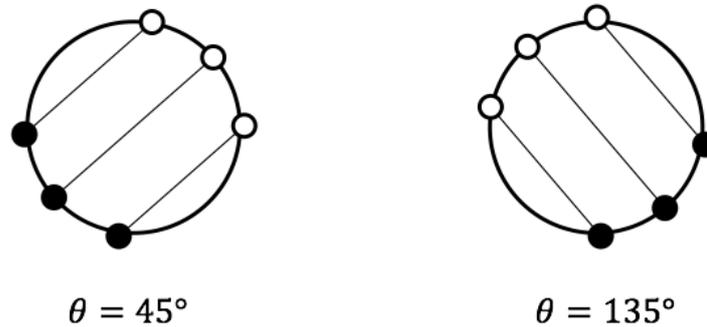


Fig. 8: Proyecciones de 45° y 135°

-Estructura basada en una geometría tipo reloj o abanico. Utilizando este método las posibles posiciones del transmisor y del receptor alrededor del objeto se encuentran equiespaciados un ángulo α . En primer lugar, se sitúa el transmisor en una de las posibles posiciones y el receptor se moverá por las otras posiciones equiespaciadas. Y se repetirá el proceso para todas las posiciones de la probeta.

- Receptor
- Transmisor

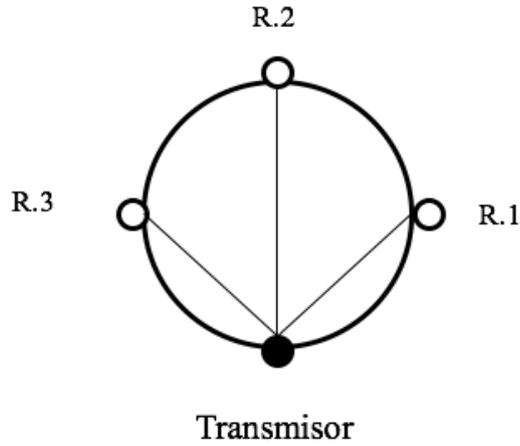


Fig.. 9 : Posiciones de transmisor y receptor en estructura de reloj

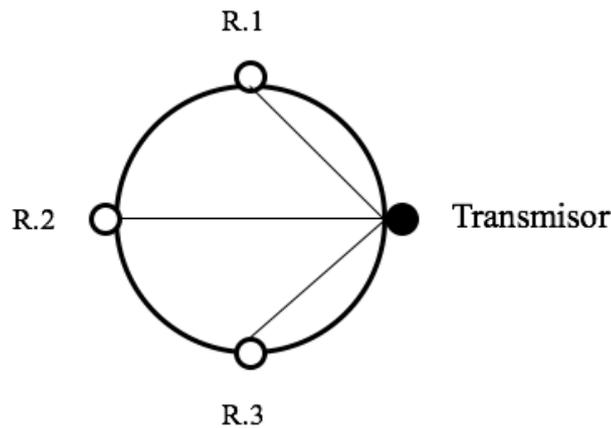


Fig. 10: Posiciones de transmisor y receptor en estructura tipo reloj.

Conociendo los dos métodos, se llegó a la conclusión de que la estructura tipo reloj es mucho más práctica, ya que no hay que mover el transmisor cada vez y esto es muy importante porque las medidas planteadas para la tomografía inicial se realizará en seco, es decir, el objeto no se somete a inmersión, por lo que se aplicará un material conductor (gel o vaselina). Por tanto, que el transmisor esté fijado en una posición facilita la tarea.

Sin embargo, esta estructura presenta un inconveniente y es que las proyecciones no se obtienen de forma directa como en la estructura de rayos paralelos, porque los rayos paralelos que describen una misma dirección no son medidos de forma consecutiva. Además, en este caso, la proyección en una determinada dirección está formada por rayos no equiespaciados. Por tanto, será necesaria una interpolación lineal para obtener rayos paralelos equiespaciados.

Otro “inconveniente” es que no podemos elegir el número de rayos, R, y de direcciones, D, ya que estos vienen dados por el ángulo de separación entre proyecciones, α :

$$D = \frac{180}{\alpha} \quad (25)$$

$$R = D - 1 = \frac{180}{\alpha} \quad (26)$$

También se puede dividir la probeta en los puntos deseados, y calcular el ángulo α .

Ahora que se conocen los dos métodos de medida, en el siguiente apartado se explicará la obtención de proyecciones, ya que utilizando un método u otro las proyecciones se obtienen mediante rayos paralelos en diferentes direcciones.

2.2.2 Tomografía ultrasónica de Difracción (TUD)

El hecho de utilizar ultrasonidos para realizar una tomografía por transmisión requiere tomar en cuenta los fenómenos de reflexión y difracción explicados anteriormente. Ahora sabemos que el método de TUD es capaz de abordar el fenómeno de refracción mediante algoritmos algebraicos combinados con técnicas de trazado de rayos o más conocido como *ray tracing*, pero no es capaz de abordar el fenómeno de difracción, en el que la onda bordea su obstáculo alejándose en este caso de lo que serían los rayos rectilíneos utilizados.

Si tuviésemos en cuenta el fenómeno de difracción, la noción del rayo carece de sentido y las proyecciones ya no pueden moldearse como integrales de línea, sino que deben ser descritas como un flujo de energía gobernado por la ecuación de onda.

Este método tiene un gran desarrollo teórico pero ha sido descrito en detalle porque escapa a los objetivos del presente trabajo..

2.2.3 Tomografía ultrasónica de Reflexión (TUR)

La tomografía ultrasónica por transmisión no es viable en aquellos casos en los que el objeto bajo estudio presenta grandes cambios de impedancia, lo que impide que la señal de ultrasonidos llegue al receptor. Como se ha visto anteriormente, un cambio de impedancias en el medio de propagación, los coeficientes de reflexión R y de transmisión T determinan qué parte de la señal se refleja y cuál se transmite. En el caso de incidencia perpendicular, los coeficientes siguen estas ecuaciones:

$$R = \frac{Z_2 - Z_1}{Z_2 + Z_1} \quad (27)$$

$$T = \frac{2Z_2}{Z_2 + Z_1} \quad (28)$$

En las expresiones anteriores, se puede observar que cuanto más grande sea la diferencia entre las impedancias de los medios 1 y 2, mayor será el coeficiente de reflexión y por tanto más grande será el eco recibido y menor la señal transmitida. Hay casos en los que la señal transmitida será 0 y la reflejada será toda la señal.

En los casos en los que la señal está bloqueada por fallos y no alcanza el receptor es imposible realizar una tomografía por transmisión, pero se pueden realizar medidas pulso-eco y obtener la localización de las faltas del objeto bajo estudio. Sin embargo, hay que tener presente las limitaciones de este tipo de tomografía, ya que por ejemplo no podemos obtener la velocidad de propagación, solo se puede caracterizar por su tamaño y localización.

Una de las técnicas más sencillas basada en medidas de reflexión consiste en iluminar el objeto desde diferentes ángulos para poder obtener información de toda el área de interés. En principio se podría pensar en un *array* de transductores dispuestos alrededor del objeto tomando imágenes en tiempo real. El problema puede ser que por limitaciones económicas no dispongamos de tantos transductores y tengamos que ir desplazando transmisor y receptor en cada medida.

Para obtener imágenes tomográficas se utilizan transductores con un haz de gran apertura para que el objeto que se va a detectar sea visible desde las distintas posiciones del transductor.

La medida realizada en cada posición mide el tiempo que ha tardado la señal en llegar al receptor, es decir, desde que se manda el pulso hasta que llega el eco.

Para reconstruir la imagen se utiliza un algoritmo basado en el trazado de un arco de circunferencia en cada una de las posiciones del transductor, a lo largo del cual se puede encontrar el objeto que ha producido el eco que se ha medido. De esta forma, la intersección de todos los arcos da forma al objeto. La imagen tomográfica resultante se obtiene quitando los píxeles de baja intensidad mediante la técnica del umbral, y resaltando los puntos brillantes de los arcos trazados. Si tenemos N medidas, la resolución será de $N \times N$ pixels, que es mucho mayor que la resolución que si se realiza tomografía por transmisión, que teniendo $R \times D$ medidas obtenemos $R \times R$ pixels de resolución.

El problema de este tipo de tomografía es que es conveniente para localizar un número de faltas limitado en una pieza, pero si hay faltas ocultas en otras, no serán reconocibles. Es cierto que las imágenes tendrán mayor resolución, pero no presentan información de la velocidad de propagación, parámetro muy importante en ultrasonidos, limitándose su caracterización a localización y tamaño.

2.2.4 Elección del método

Después de estudiar los 3 tipos que tomografía que existen, se puede decir que todos son satisfactorios dependiendo del resultado que se busque. Si lo que quiere es localizar y conocer tamaño de las faltas en un objeto, con la tomografía por reflexión (TUR) sería suficiente. Si quiere conocer mejor el objeto bajo estudio, es decir, realizar un mapa de velocidades que nos da mucha más información del objeto, recurriremos a los otros tipos de tomografía. Si el objeto no presenta altos cambios de índice de refracción, la tomografía por difracción (TUD) se puede utilizar, ya que modela la propagación de los ultrasonidos como un flujo de energía gobernado por la ecuación de onda, que tiene en cuenta los fenómenos de refracción y difracción propios de los ultrasonidos. Por otro lado, si el objeto presenta cambios bruscos de índice de refracción, y se desea caracterizar el objeto determinando las velocidades locales de propagación de los ultrasonidos, es necesaria la tomografía ultrasónica basada en el trazado de rayos (TUTR), aunque en este caso se deberían utilizar algoritmos algebraicos de reconstrucción combinados con técnicas de *ray tracing* para solucionar los problemas de la refracción. Si el tamaño de las faltas es mayor que la longitud de onda, no ocurre la difracción, por tanto se utilizarán los procesos adecuados para obtener los tiempos de propagación correctos en presencia de difracción.

Como en nuestro proyecto el material presenta grandes cambios de índice de refracción hemos utilizado el método de tomografía basada en el trazado de rayos (TUTR).

2.2.5 Obtención de proyecciones

[1] En este apartado se va a describir el proceso de obtención de proyecciones, con las que poder reconstruir una imagen tomográfica.

Por tanto, existe una $f(x, y)$ que representa el valor de alguna propiedad física distribuida en una sección plana del objeto bajo estudio, como puede ser la velocidad o la atenuación. Como se ha citado anteriormente, hay que pensar en rayos. Llamamos rayo a la línea recta que describe la trayectoria de la señal ultrasónica desde el transmisor al receptor, que viene dada por el ángulo θ que forma la perpendicular a la dirección de propagación y por la distancia t al centro.

Se llama proyección a la integral de línea que, como su nombre indica, representa la integral de $f(x, y)$ a lo largo de dicha recta AB. Por tanto, si utilizamos un sistema de coordenadas tal que en la fig(incluir), el objeto viene definido por la función objeto de $f(x, y)$ y cada una de las integrales de línea viene definido por el ángulo θ y t . Con esto, la recta AB tiene esta forma:

$$t = x \cos(\theta) + y \sin(\theta) \quad (29)$$

Y se utilizará dicha relación para definir esta integral:

$$P(t) = \int_{\text{línea}(\theta,t)} f(x,y) ds \quad (30)$$

Pero si utilizamos la función delta de dirac, la expresión anterior queda de esta forma:

$$P(t) = \iint_{-\infty}^{\infty} f(x,y) \delta(x \cos(\theta) + y \sin(\theta) - t) dx dy \quad (31)$$

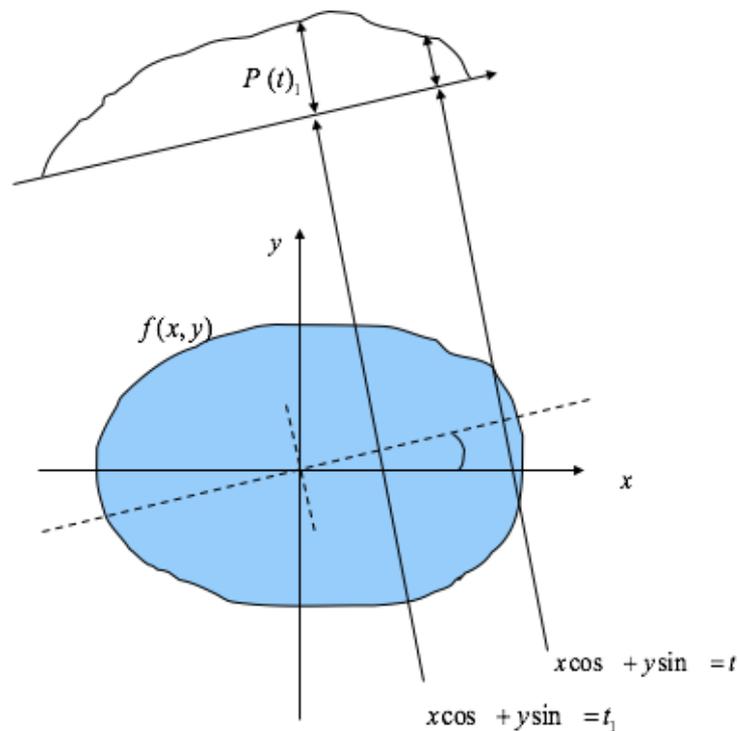


Fig. 11: Concepto de proyección [4].

Esta función $P_{\theta}(t)$ se conoce como transformada de Radon de la función $f(x,y)$ que representa la proyección de dicha función bajo una determinada dirección de propagación correspondiente al ángulo θ .

Estas proyecciones se miden experimentalmente, y pueden ser medidas de atenuación, velocidades o retardos, potencia, etc. Esto depende del objeto y material de estudio. Por ejemplo, en el primer caso, la imagen tomográfica que se construiría sería un mapa de atenuaciones, o si se midieran retardos, obtendríamos un mapa de velocidades o al revés.

En el presente TFG, se han utilizado 4 parámetros para realizar la reconstrucción de la imagen tomográfica, siendo potencia, amplitud, atenuación y velocidad. Se explicarán en el apartado 2.2.

Por tanto, para obtener la información necesaria para recuperar la imagen tomográfica del objeto, hay que trazar rayos paralelos en una determinada dirección y realizar barridos desde 0° a 180° . Para realizar esto con un coste computacional que no sea infinito deberemos escoger un

número finitos de rayos y de direcciones. Siendo R el número de rayos, que corresponde con las rectas AB, t , citadas anteriormente y siendo D el número de direcciones que coinciden con las variaciones del ángulo θ .

Al conjunto de retardos producidos, o atenuaciones producidas, por todos los rayos paralelos en la misma dirección, se le llama proyección.

Veamos ejemplos de proyecciones para diferentes θ :

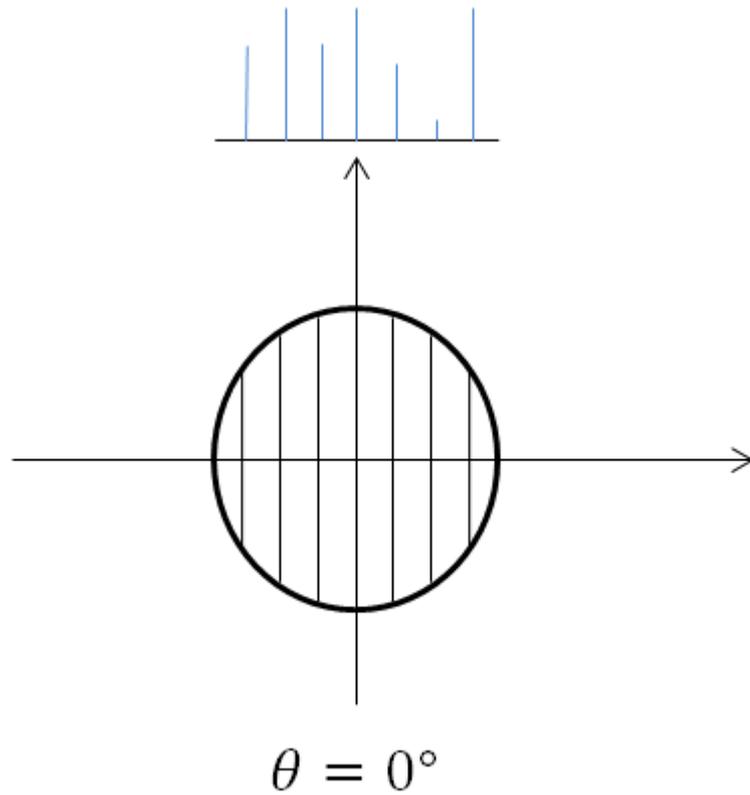


Fig. 12: Proyección para 7 rayos paralelos y $\theta = 0^\circ$

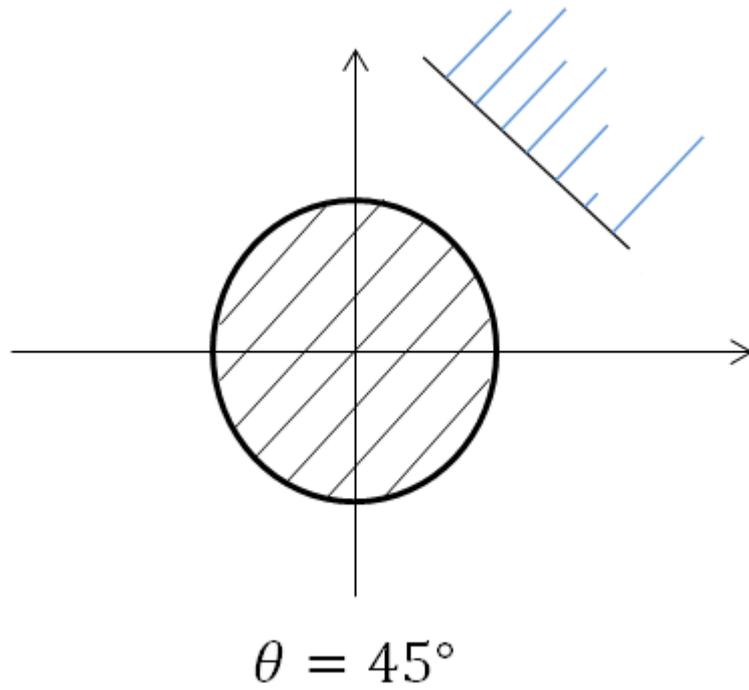


Fig. 13: Proyección 7 rayos paralelos y $\theta = 45^\circ$

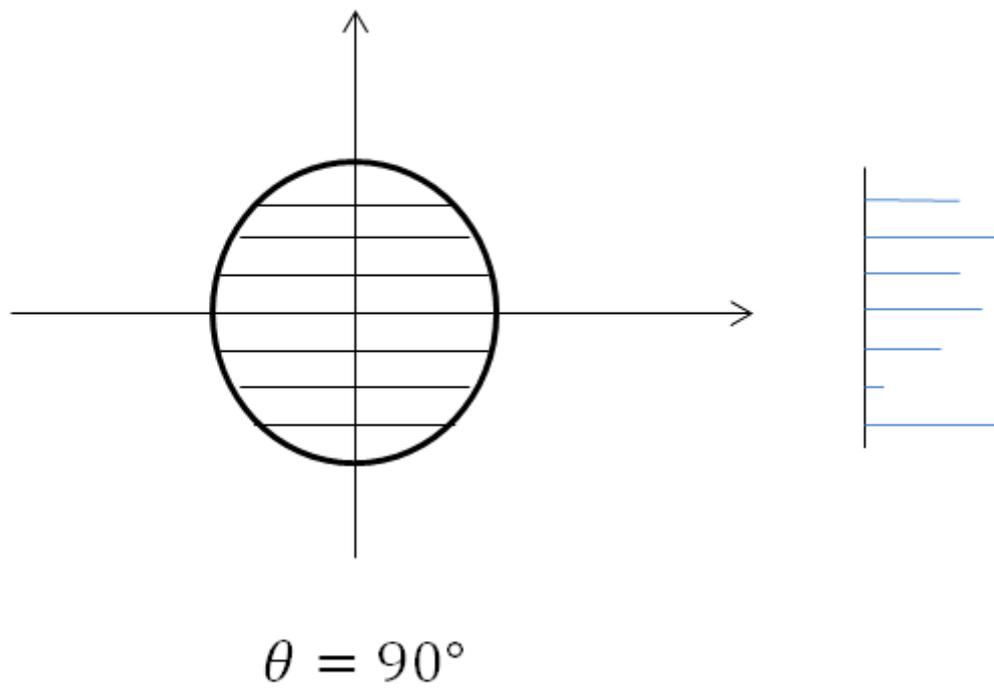


Fig. 14: Proyección 7 paralelos y $\theta = 90^\circ$

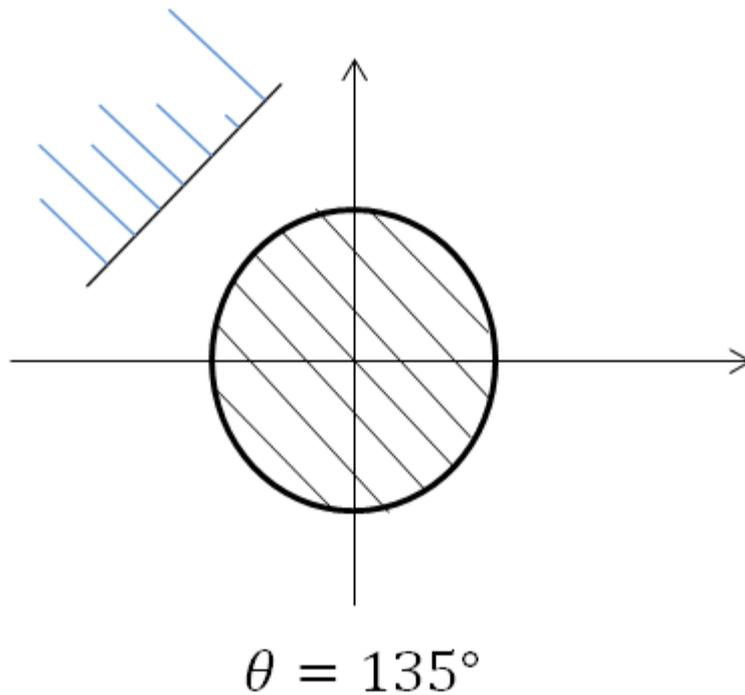


Fig. 15: Proyección 7 rayos paralelos y $\theta = 135^\circ$

2.3 Tomografía: algoritmos de reconstrucción

Según [1], para poder llegar a una imagen tomográfica, debemos aplicar algún tipo de algoritmo a los datos que tenemos, es decir, la obtención de una imagen tomográfica se realiza a partir de las proyecciones que hemos medido sobre el objeto bajo estudio. Con dichos algoritmos e introduciéndoles como entrada las proyecciones de velocidades o atenuaciones, podemos realizar la reconstrucción de la imagen.

Existen dos tipos de algoritmos de reconstrucción básicos: los que están basados en métodos de transformación y los algoritmos algebraicos. Por tanto, veamos las diferencias:

- Métodos de transformación: En este caso el enfoque parte de la formulación de un modelo matemático, en el que, tanto las proyecciones conocidas, como la función objeto se definen de forma continua. El problema se resuelve de manera analítica, adaptando después la solución a nuestros datos.
- Algoritmos algebraicos: En este caso se formula un modelo matemático con los valores discretos de las proyecciones y la función objeto. Mediante un sistema de ecuaciones y un proceso iterativo podemos encontrar la solución numérica.

Vista la diferencia de esta forma, se procederá a explicar lo que hemos utilizado en nuestro caso, los métodos de transformación.

2.3.1 Métodos de transformación

Los métodos de transformación se basan en el Teorema de Fourier Slice, el cual establece una relación entre la transformada de Fourier unidimensional de las proyecciones en una determinada dirección, con la transformada de Fourier bidimensional de la función objeto evaluada en una de sus radiales. Con esto se obtiene un único modelo de tomografía conceptualmente simple, pero se pueden obtener implementaciones de forma distinta cambiando las expresiones fundamentales que soportan esta teoría. Por tanto, existen dos algoritmos basados en el Teorema de Fourier Slice, cuya implementación es diferente, ya que en cada uno se usan diferentes aproximaciones y obteniendo, por tanto, soluciones diferentes.

El teorema de Fourier Slice establece la relación entre la transformada de Fourier de las proyecciones y las transformada de Fourier de la función objeto $f(x,y)$:

La transformada de Fourier de la proyección $P_{\theta_1}(t)$ de la imagen $f(x,y)$ tomada con la estructura de rayos paralelos bajo un ángulo θ , proporciona un corte de la transformada de Fourier bidimensional del objeto $F(u,v)$, que forma un ángulo θ con el eje u en el dominio de la frecuencia. (incluir fig hecha).

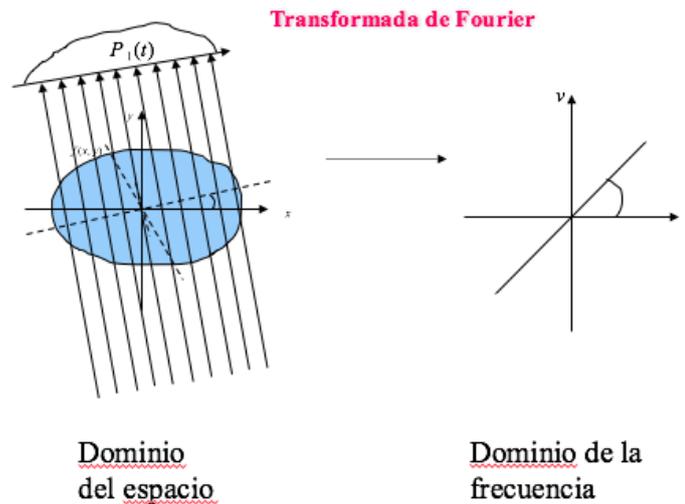


Fig. 16: En [4] encontramos el Teorema de Fourier Slice

No obstante, el teorema de Fourier Slice es el que vamos a emplear como base para explicar el Algoritmo de Retroproyección Filtrada, el cual es el que se ha utilizado en este trabajo.

2.3.2 Algoritmo de Retroproyección Filtrada

En [2] podemos ver que los métodos de retroproyección filtrada se basan en el teorema de «cortes de Fourier» o teorema de la proyección, que afirma que:

«La transformada unidimensional de Fourier de la proyección de una imagen $f(x, y)$, obtenida a partir de rayos paralelos entre sí y formando un ángulo h con el eje x , es el corte o muestreo de

la transformada bidimensional de Fourier de la imagen $F(u, v)$ a lo largo de una línea que forma un ángulo θ con el eje u .

Según este teorema, si disponemos de las proyecciones de una imagen es posible determinar cuál es esa imagen calculando una transformada bidimensional inversa de Fourier.

El resultado anterior indica que tomando P proyecciones de un objeto en los ángulos $\theta_1, \theta_2, \dots, \theta_p$ y obteniendo la transformada continua de Fourier de cada una de ellas, podemos determinar los valores de $F(u, v)$ (transformada bidimensional del objeto) en líneas que pasan por el origen formando los ángulos $\theta_1, \theta_2, \dots, \theta_p$:

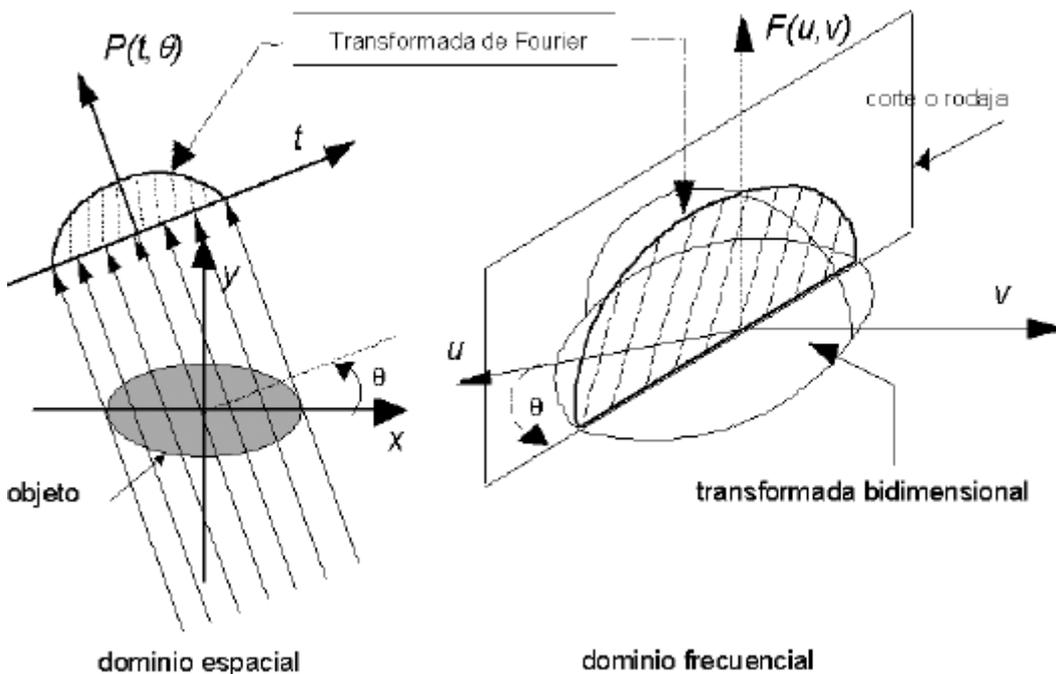


Fig.:17 El teorema de cortes de Fourier da una relación entre la transformada unidimensional de Fourier de una proyección y la transformada bidimensional de Fourier del objeto.

Aunque el teorema de cortes de Fourier sugiere un algoritmo sencillo de reconstrucción, es conveniente reescribir las ecuaciones y presentarlas como un nuevo algoritmo. Para ello, se sustituyen las coordenadas rectangulares (u, v) por las polares (ρ, θ) :

$$u = \rho \cos \theta \tag{32}$$

$$v = \rho \sin \theta \tag{33}$$

$$du dv = \rho d\rho d\theta \tag{34}$$

De forma que la transformada de la imagen se convierte en:

$$f(x, y) = \int_0^{2\pi} \int_0^{\infty} F(\rho, \theta) e^{j2\pi\rho(x \cos \theta + y \sin \theta)} \rho \, d\rho \, d\theta \quad (35)$$

Esta ecuación, considerando por separado las variaciones de θ entre $(0..180]$ y $(180..360]$ y la propiedad:

$$F(\rho, \theta + 180) = F(-\rho, \theta) \quad (36)$$

Se puede escribir como:

$$f(x, y) = \int_0^{\pi} \left[\int_{-\infty}^{+\infty} F(\rho, \theta) |\rho| e^{j2\pi\rho t} \, d\rho \right] d\theta \quad (37)$$

Donde $t = x \cos \theta + y \sin \theta$ que simplifica la expresión.

Si sustituimos la transformada bidimensional $F(\rho, \theta)$ por la transformada unidimensional de la proyección $S_{\theta}(\rho)$, obtenemos:

$$f(x, y) = \int_0^{\pi} \left[\int_{-\infty}^{+\infty} S_{\theta}(\rho) |\rho| e^{j2\pi\rho t} \, d\rho \right] d\theta \quad (38)$$

Esta ecuación se presenta como una forma sencilla de estimar la imagen $f(x, y)$. También representa una operación de filtrado (i.e. convolución), donde la respuesta en frecuencia del filtro es $H(\theta) = |\rho|$. El resultado de esta convolución se conoce como *proyección filtrada*. La suma de las diferentes proyecciones filtradas (una por cada θ) permiten estimar así la imagen $f(x, y)$.

El algoritmo de retroproyección filtrada no considera la existencia de errores. En el proceso de reconstrucción, sin embargo, aparecen errores de varios tipos, especialmente los debidos a disponer de unos datos insuficientes y aquéllos producidos por la presencia de un ruido aleatorio en las proyecciones.

En primer lugar, un muestreo por debajo del necesario (submuestreo) o no disponer de un número adecuado de proyecciones, introduce errores en la reconstrucción. Ambos existen por la discretización del algoritmo continuo de retroproyección filtrada: la imagen se reconstruye a partir de un número finito de muestras y a partir de un número finito de proyecciones. Un problema fundamental de las imágenes tomográficas en general es que los objetos, y, por tanto, sus proyecciones no están limitados en banda. En otras palabras, el ancho de banda de las proyecciones es mayor que la más alta frecuencia capaz de ser captada con una frecuencia de muestreo dada. Para reconstruir adecuadamente por retroproyección filtrada un objeto, el número de muestras por proyección debe cumplir Nyquist y adaptarse al tamaño del más pequeño objeto que se desea ver (resolución). Si ese número es K , existe un número óptimo de proyecciones que es, grosso modo, el mismo K . Una matriz de imagen adecuada es en tal caso de $K \times K$.

Por otra parte, la utilización del filtro $H(\rho) = |\rho|$ aumenta el ruido de altas frecuencias existente en las proyecciones. Es posible evitar en parte este realce del ruido eligiendo ventanas adecuadas, aunque ello implica distorsionar ligeramente la imagen.

Cabe destacar, que todo este algoritmo utilizado en la práctica es una versión estable y discreto de la transformada inversa de Radon, la cual se vio en el apartado de obtención de

proyecciones. Se utiliza este algoritmo ya que la transformada de Radon es inestable cuando hay ruido.

En las primeras tomografías, se empleaba el algoritmo de retroproyección simple, pero la reconstrucción de la imagen quedaba algo borrosa. Con el algoritmo de retroproyección filtrada la imagen final queda mucho más nítida. Veamos las mejoras de un algoritmo respecto de otro mediante dos ensayos:

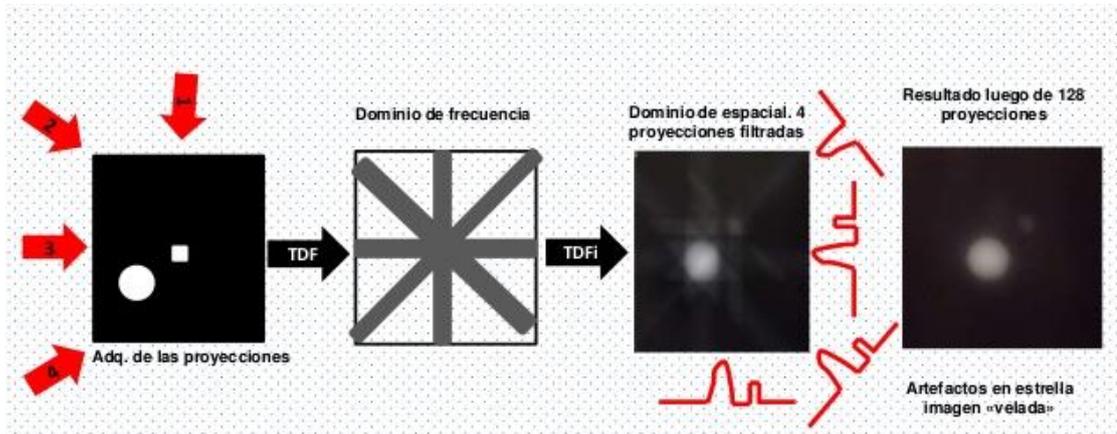


Fig. 18: Algoritmo de Retroproyección Simple.

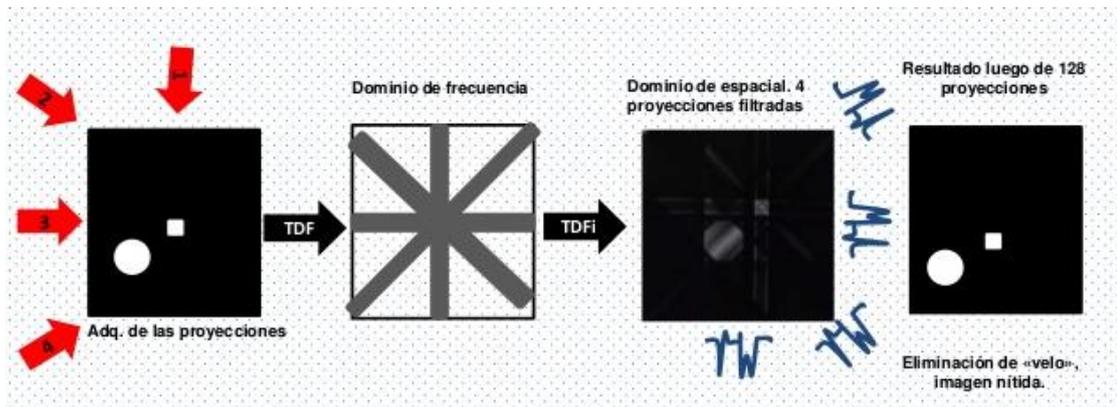


Fig. 19: Algoritmo de Retroproyección Filtrada.

2.3.3 Elección del algoritmo de reconstrucción

Finalmente y para este trabajo se ha elegido el algoritmo de Retroproyección Filtrada. Este algoritmo está implementado de forma óptima en una función interna de Matlab para representar los resultados. Dicha función es *iradon* la cual implementa la transformada inversa de Radon e implementa el algoritmo de retroproyección filtrada:

```
iradon uses the filtered backprojection algorithm to perform the inverse
Radon transform. The filter is designed directly in the frequency
domain and then multiplied by the FFT of the projections. The
projections are zero-padded to a power of 2 before filtering to prevent
spatial domain aliasing and to speed up the FFT.
```

Fig. 20: Descripción de la función iradon de Matlab.

Capítulo 3. Medidas preliminares: tomografía por contacto

Antes de comenzar con el sistema automatizado, se realizó un proceso previo para ir conociendo las tomografías ultrasónicas. En este apartado, se intentó obtener una tomografía mediante ultrasonidos sin una toma masiva de medidas lo que dificulta la calidad de la reconstrucción debido al reducido número de medidas.

En los siguientes apartados se explicará, por una parte, una forma general de obtener tomografías, con sus procedimientos de medida, obtención de proyecciones y algoritmos de reconstrucción; por otra parte, cómo se realizó en el laboratorio, los equipos utilizados, el objeto bajo estudio y la herramienta de software para el procesamiento de esas señales.

3.1 Método de trabajo y equipamiento empleado

Atendiendo a los conceptos descritos en el apartado 1.4.2, elegimos el método de trazado de rayos y en este apartado se procederá a explicar cómo se ha implementado la toma de medidas en el laboratorio.

Por tanto, para esta primera “tomografía” utilizaremos:

-Una probeta de cemento, la cual está fisurada por dentro. Incluye dos fisuras, una pequeña por una cara de la probeta, y una un poco más grande por la otra cara. Dicha probeta es circular de 0,15 m de diámetro. Se ha dividida en 24 puntos de medida lo que supone una variación entre cada proyección de 15°

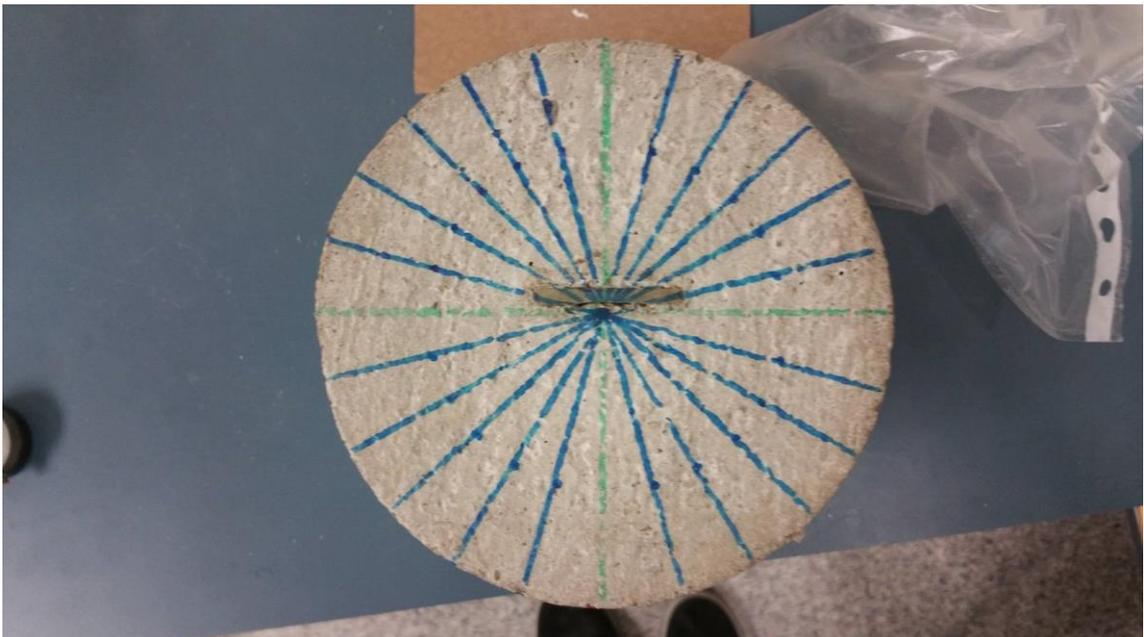


Fig. 21: Fisura pequeña de la probeta.

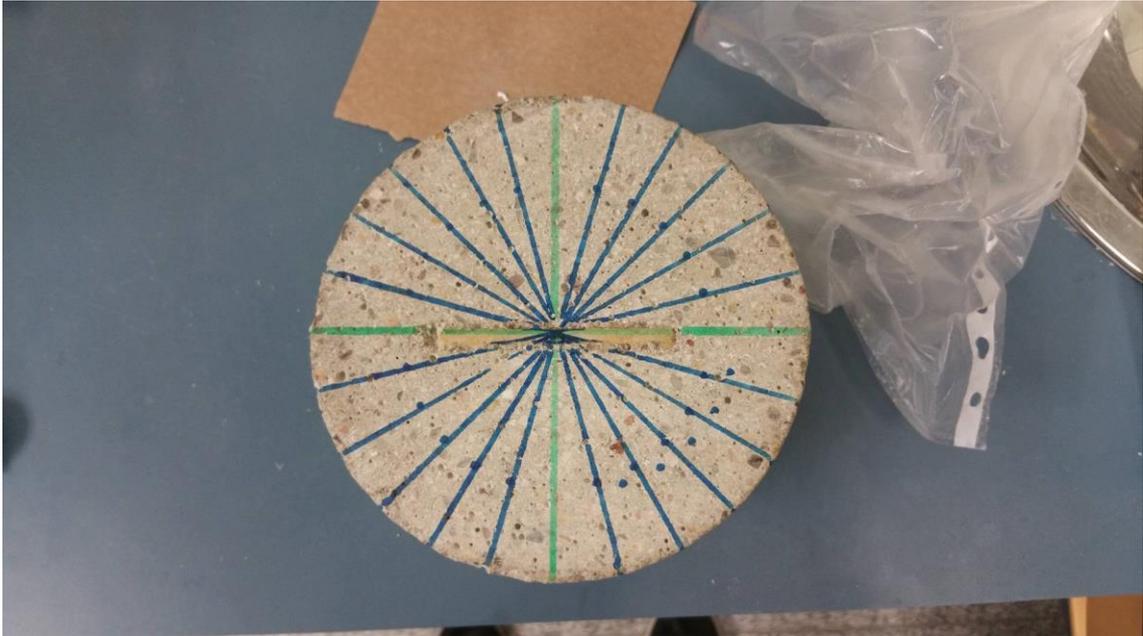


Fig. 22: Fisura grande de la probeta.

-Un generador de señales (Agilent 33120A), a través del cual, generaremos la señal que queremos transmitir. Para este experimento previo se usó una señal tipo Chirp, la cual tiene unas características especiales, ya que la señal Chirp (basada en la técnica de espectro ensanchado), realiza un barrido de frecuencias para expandir su espectro. Por tanto realizamos un barrido en frecuencia, con un margen de frecuencias que se encuentra en la banda de ultrasonidos, y así veremos que frecuencias presentan una mejor propagación.



Fig. 23: Generador de señales.

-Un osciloscopio (DPO3014) mediante el cual podremos visualizar las señales que transmitimos y las que recibimos. Aquí tendremos 3 canales, el primero será recepción, el segundo representa la señal de transmisión y en el último veremos la señal de transmisión amplificada.



Fig. 24: Osciloscopio Tektronix.

-Un amplificador en transmisión (FALCO WMA-300) que, como su propio nombre indica, se utilizará para amplificar la señal de transmisión debido a las pérdidas que se producen en el tramo transductor-probeta.

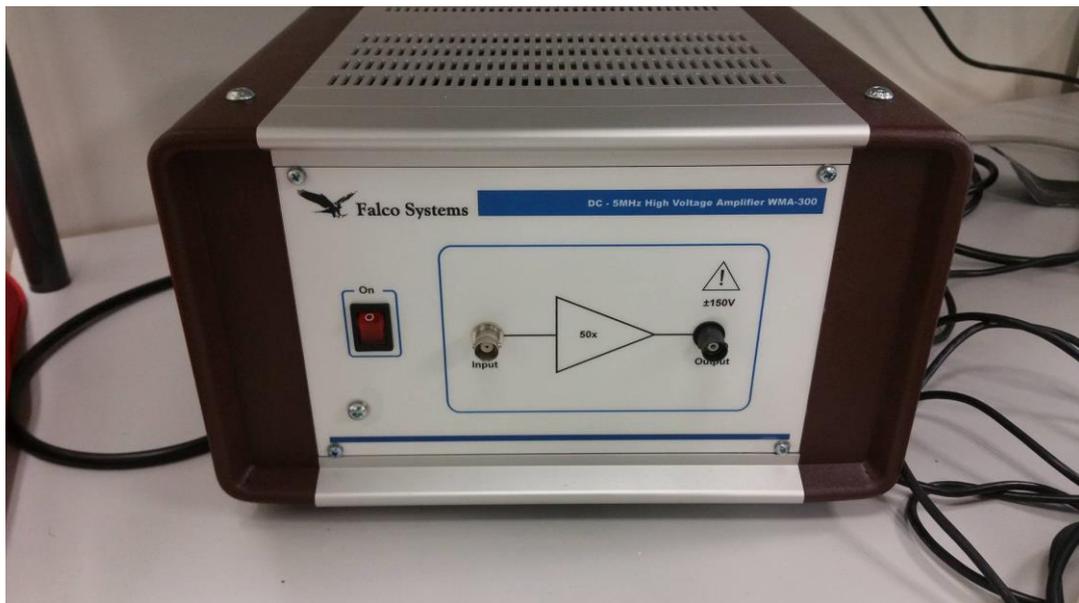


Fig. 25: Amplificador transmisión.

- Un amplificador en recepción (Panametrics) de ganancia variable de 40 dB o 60 dB, para compensar las pérdidas probeta-transductor de recepción.

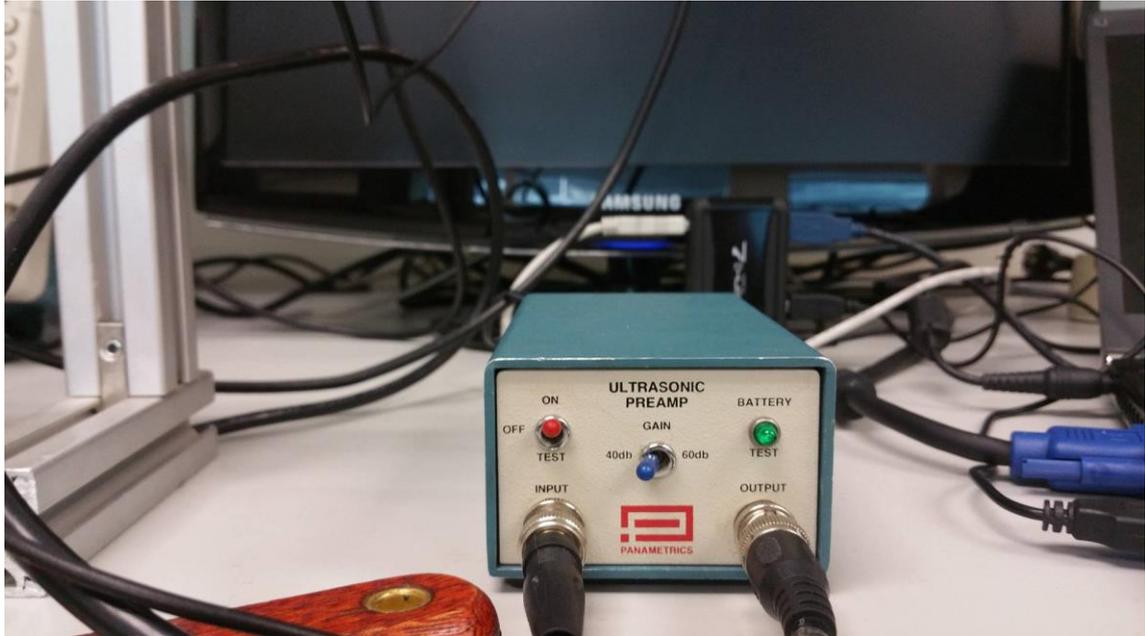


Fig. 26: Preamplificador.

- Un ordenador portátil, con el que poder guardar las medidas que hacemos. En este ordenador disponemos de un programa en Matlab, el cual nos permite mandar la señal que queremos, con sus diferentes parámetros, al generador de señales y por otro lado adquirir los resultados del osciloscopio y poder guardarlos como se citará en este mismo apartado.

Visto los equipos utilizados para realizar esta primera tomografía, procederemos a explicar cómo lo hemos realizado. Como ya se ha mencionado, para realizar nuestras tomografías hemos utilizado el método de medida basado en estructura de reloj. Por tanto, con esta estructura, debemos realizar medidas en todos los puntos de la probeta. El primer paso es cubrir los transductores con gel conductor porque si no se hiciera de este modo, la desadaptación de impedancias sería tan grande que no recibiríamos nada en el punto de recepción.

Nuestra probeta tiene 24 puntos de medida equiespaciados, de 1 a 24, entonces, para cubrir toda la superficie para cada posición del transmisor moveremos el receptor media probeta. Como visualmente esto no es tan fácil, véanse las figura 9 y 10:

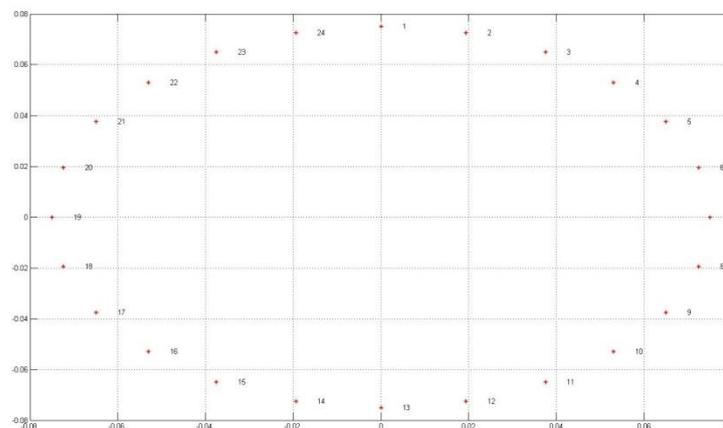


Fig. 27: Puntos de medida

Esos son los puntos de medida que tenemos. Volviendo al método de medida utilizado, nuestra primera medida se realiza de forma que, empezamos con el transmisor en el punto 1, y

el receptor en el punto 7 y tomamos una adquisición del osciloscopio, es decir, lo que se ha transmitido desde el punto 1 y que ha llegado en el punto 7. El siguiente paso es dejar el transmisor en 1 y mover receptor a 8, y así hasta 19 que sería la mitad de la probeta. Estas 13 medidas se guardan en un fichero .mat. Luego se movería el transmisor a 2 y el receptor empezaría en 8 y se realizan otras 13 medidas. Y por tanto obtenemos 24 ficheros con 13 medidas cada uno:

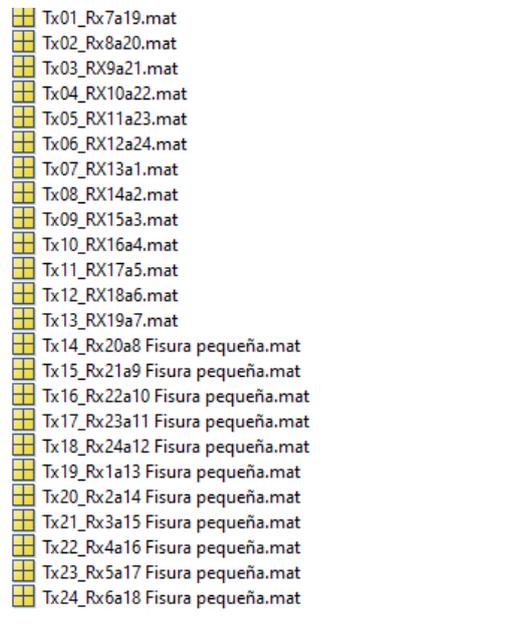


Fig. 28: Ficheros con las medidas realizadas.

Veamos un fichero por dentro para comprenderlo mejor:

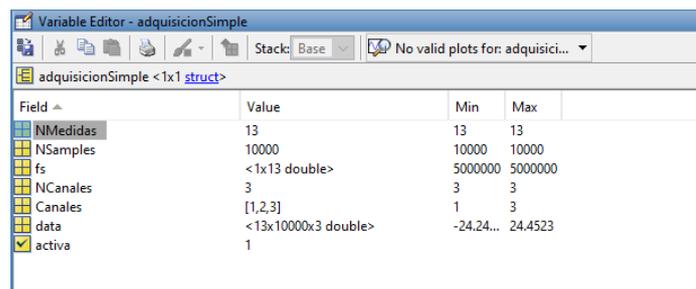


Fig. 29: Contenido de cada fichero.

Aquí tenemos toda la información que hay dentro de cada fichero. Vemos que hay 13 adquisiciones, con 10.000 muestras por medida, y con 3 canales como se citó al nombrar el osciloscopio. Pero lo importante de aquí es la matriz *data*, la cual contiene toda la información que nos interesa. Se puede observar que es una matriz tridimensional en la cual tenemos, las medidas realizadas por los puntos de medida de la probeta por los 3 canales que hemos cogido del osciloscopio.

Tenemos 24 ficheros que contienen la información que necesitamos. Para procesar toda esta información, hemos creado un fichero en Matlab el cual juntará todas las medidas para poder procesar toda la información a la vez.

3.2 Preparación de los datos

El siguiente paso, es “ordenar” toda la información que tenemos en el fichero para conformar las proyecciones a diferentes ángulos. Como bien se ha explicado antes, hemos realizado las medidas en estructura de reloj, por tanto, debemos la transformación a aplicar nos deberá generar proyecciones de rayos paralelos para posteriormente obtener los parámetros asociados de cada rayo y poder alimentar el algoritmo de reconstrucción iradon.

3.2.1 Generación de las posiciones del transmisor y receptor para cada proyección

En primer lugar y mediante un bucle en Matlab generamos las posiciones de los transmisores y receptores asociados a las proyecciones deseadas:

```
PRtx(2,:) = [ 1 2 3 4 5 6 7 ]-1;
PRrx(2,:) = [19 18 17 16 15 14 13 ]-1;
Pangle(2) = 15;

for ii=1:11
    for ii2 = 1:2
        PRtx(end+1,:) = mod(PRtx(end-1,:)+1,24);
        PRrx(end+1,:) = mod(PRrx(end-1,:)+1,24);

        Pangle = [Pangle Pangle(end)+15];
    end
end
```

Fig. 30: Generación de las proyecciones.

Cada fila de las variables PRtx y PRrx está asociada a una proyección, mientras que los valores de cada una de esas filas están asociadas a la posición del transmisor y del receptor en cada caso. Mediante el bucle, conseguimos generar todas las proyecciones necesarias para la reconstrucción. Podemos ver como se genera también el ángulo asociado a dichas proyecciones.

Podemos observar las proyecciones calculadas en la siguiente imagen:

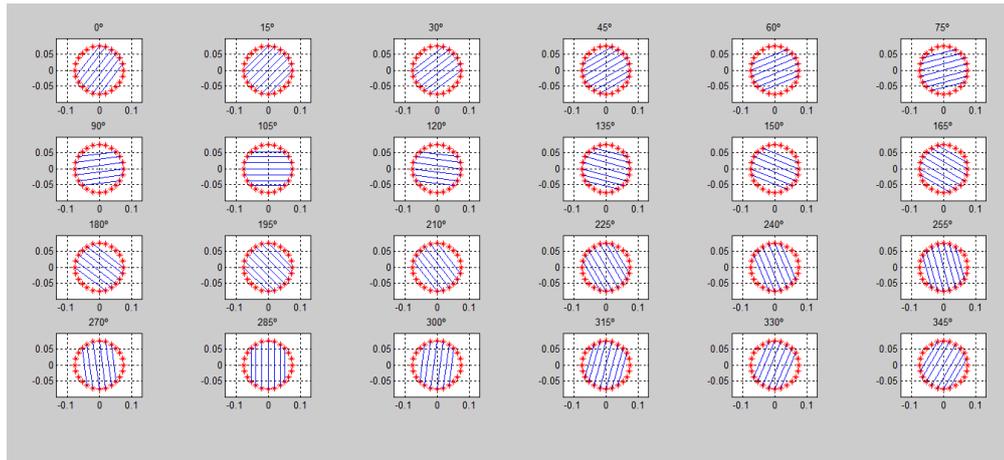


Fig. 31: Proyecciones obtenidas.

En la que la proyección 0° es la que el transductor de transmisión está en posición 1 y receptor en 7, el siguiente rayo paralelo sería transmisor en 2 y receptor en 8 y así sucesivamente. Y se va girando 15 grados ya que $360/24 \cong 15^\circ$.

3.2.2 Asociación entre posición del transmisor y receptor y medida asociada

El segundo paso necesitamos relacionar la posición del transmisor y del receptor con el número de medida asociado. Para ello se crea la matriz Matriz Medidas, en la que las filas serán la posición del transmisor y las columnas serán la posición del receptor. Creamos la matriz a partir del siguiente bucle:

```
MatrizMedidas = zeros(24,24); % Matriz de 24 txon x 24 rxon . Muchas posiciones estarán a 0

% Vamos a ir rellenando el fichero
mascara = [zeros(1,6) ones(1,13) zeros(1,5)];
indiceRegistro = [zeros(1,6) 1:13 zeros(1,5)];
MatrizMedidas(1,:) = indiceRegistro;

for iiTx=2:24
    mascara = circshift(mascara,[0 1]);
    indiceRegistro = (circshift(indiceRegistro,[0 1])+13).* mascara;

    MatrizMedidas(iiTx,:) = indiceRegistro; % Guardamos las posiciones
end
```

Fig. 32: Creación de la matriz de medidas.

Con esto obtenemos una matriz 24x24, la cual queda de esta forma:

	7	8	9	10	11	12	13	14	15	16	17	18
1	1	2	3	4	5	6	7	8	9	10	11	12
2	0	14	15	16	17	18	19	20	21	22	23	24
3	0	0	27	28	29	30	31	32	33	34	35	36
4	0	0	0	40	41	42	43	44	45	46	47	48
5	0	0	0	0	53	54	55	56	57	58	59	60
6	0	0	0	0	0	66	67	68	69	70	71	72
7	0	0	0	0	0	0	79	80	81	82	83	84
8	0	0	0	0	0	0	0	92	93	94	95	96
9	0	0	0	0	0	0	0	0	105	106	107	108
10	0	0	0	0	0	0	0	0	0	118	119	120
11	0	0	0	0	0	0	0	0	0	0	131	132
12	0	0	0	0	0	0	0	0	0	0	0	144
13	169	0	0	0	0	0	0	0	0	0	0	0
14	181	182	0	0	0	0	0	0	0	0	0	0
15	193	194	195	0	0	0	0	0	0	0	0	0
16	205	206	207	208	0	0	0	0	0	0	0	0
17	217	218	219	220	221	0	0	0	0	0	0	0
18	229	230	231	232	233	234	0	0	0	0	0	0
19	241	242	243	244	245	246	247	0	0	0	0	0
20	253	254	255	256	257	258	259	260	0	0	0	0
21	265	266	267	268	269	270	271	272	273	0	0	0
22	277	278	279	280	281	282	283	284	285	286	0	0
23	289	290	291	292	293	294	295	296	297	298	299	0
24	301	302	303	304	305	306	307	308	309	310	311	312

Fig. 33: Matriz de medidas.

En la que se puede ver, que la fila 1- columna 7, sería la medida número 1, la siguiente sería dejando el transmisor en 1 y moviendo el receptor al 8 y así.

A partir de aquí, ya podemos acceder al registro de los datos de una forma segura ya que sabemos ciertamente qué medida debemos coger atendiendo a la posición del transmisor y el receptor. Podemos ver que hay valores de la matriz que son 0, esto es debido a que hay posiciones en las que no se han tomado la medida por limitaciones físicas del montaje.

3.2.3 Asociación inversa

Cuando ya tenemos esta matriz, la convertimos en vectores. De esta forma podremos hacer una asociación inversa, es decir, a partir de un dato podemos saber en qué posiciones están el transmisor y el receptor:

```
[MatrizRx MatrizTx] = meshgrid(1:24,1:24);
VectorMedidas = MatrizMedidas(:);
VectorTx = MatrizTx(:);
VectorRx = MatrizRx(:);
indice = find(VectorMedidas==0);
VectorMedidas(indice) = []; VectorTx(indice) = []; VectorRx(indice) = [];
[VectorMedidas indice] = sort(VectorMedidas);
VectorTx = VectorTx(indice);
VectorRx = VectorRx(indice);
clear indice MatrizRx MatrizTx
```

Fig. 34: Matriz de Medidas.

Con esto ya tenemos todos los datos preparados para poder calcular los diferentes parámetros asociados a cada rayo de cada proyección que nos darán la información sobre qué está pasando dentro de la probeta.

3.2.4 Cálculo de parámetros

Como bien se ha citado anteriormente, los parámetros a medir más utilizados para tomografía ultrasónica suelen ser velocidad o atenuación. En nuestro caso se han utilizado los dos, y por otro lado, se han medido también parámetros como la potencia o amplitud.

3.2.4.1 Energía

En primer lugar, calculamos la energía de una secuencia utilizando la siguiente expresión:

$$E_x = \sum |x(n)|^2 \quad (39)$$

La cual queda implementada mediante el siguiente código donde hemos recortado la señal con el fin de coger el primer frente de onda.

```
parametros.energia(ii) = sum(data_save(ii,50:500).^2) % Potencia.
```

Hacemos el sumatorio de la señal y lo elevamos al cuadrado como dice la fórmula, pero lo normalizamos con la distancia entre el transmisor y el receptor, calculada según la siguiente fórmula que simplemente es calcular el módulo del vector que une el transmisor con el receptor.:

```
distAux = sqrt(sum((puntosMedida(VectorTx(ii),2:3) - puntosMedida(VectorRx(ii),2:3)).^2)); % Distancia recorrida por el haz
```

3.2.4.2 Atenuación

Otro parámetro utilizado, es la atenuación, la cual se calcula a partir del cociente entre la energía transmitida y la energía recibida:

$$\alpha = 10 \cdot \frac{\log_{10} \frac{E_{tx}}{E_{rx}}}{distancia} \quad (40)$$

En nuestro código queda implementado de la siguiente manera donde hemos asumido que $E_{tx} = 1$ para todas las medidas:

```
parametros.atenuacion(ii) = -log10(parametros.energia(ii))/distAux^2  
% Atenuación
```

3.2.4.3 Amplitud

Y por último, calculamos la amplitud de la señal haciendo el máximo de su valor absoluto:

```
parametros.amplitud(ii) = max(abs(data_save(ii,1500:4500))); % Amplitud
```

Estos son los 4 parámetros que vamos a utilizar para reconstruir la imagen tomográfica. Es posible que con un mapa de velocidades o atenuaciones hubiera sido suficiente, pero como estos parámetros también nos dan información útil sobre la señal, los hemos utilizado para apoyar el

ensayo.

3.2.5 Mejora de las proyecciones

Antes de realizar la reconstrucción de la imagen tomográfica, como se ha explicado en el apartado de tomografía basada en trazado de rayos, necesitamos proyecciones con el mayor número de rayos posibles. Debido al sistema de adquisición manual que estamos implementando el número de rayos por proyección es reducido. Por ello, para mejorar la visualización (que no la resolución) de la imagen reconstruida, se realiza una interpolación de cada una de las proyecciones para aumentar el número de puntos con el que hacer la reconstrucción..

Para ello, utilizamos la función *interp1* de Matlab y le aplicamos un método de interpolación lineal a cada una de las proyecciones que hemos generado para cada uno de los parámetros de potencia, amplitud, atenuación y velocidad.

```
for jj = 1:numRayos % Condicionamos el bucle en función del número de rayos
    % for jj = 1:size(PRTx,2)
    numRegistro = MatrizMedidas(PRTx(ii,jj),PRrx(ii,jj));
    if numRegistro==0
        error('Se ha cogido un par de medidas que no existen')
    else
        auxY1(jj) = parametros.velocidad(numRegistro); % Velocidad
        auxY2(jj) = log10(parametros.potencia(numRegistro)); % Potencia
        auxY3(jj) = ((parametros.atenuacion(numRegistro))); % Atenuación
        auxY4(jj) = parametros.amplitud(numRegistro); % Amplitud
    end
end

% matrizProyeccionesAuxiliar(1:numProyecciones,ii) = auxY(:);

metodoInterpolacion = 'linear'; %spline
% Interpolamos y almacenamos
matrizProyecciones(:,ii,1) = interp1(xPR,auxY1,xInterpol,metodoInterpolacion,'extrap').*mascara';
matrizProyecciones(:,ii,2) = interp1(xPR,auxY2,xInterpol,metodoInterpolacion,'extrap').*mascara';
matrizProyecciones(:,ii,3) = interp1(xPR,auxY3,xInterpol,metodoInterpolacion,'extrap').*mascara';
matrizProyecciones(:,ii,4) = interp1(xPR,auxY4,xInterpol,metodoInterpolacion,'extrap').*mascara';
% matrizProyecciones(:,ii) = interp1(xPR1,auxY,xInterpol,'spline','extrap').*mascara';
end
```

Fig. 35: Interpolación.

Con todo esto, ya tenemos los datos preparados para realizar la la tomografía aplicando un algoritmo de reconstrucción. Como se ha explicado, hay diversos métodos, pero en este caso se ha utilizado el que viene implementado en Matlab iradon, ya que se adapta a nuestras necesidades. Antes de explicar este método, veremos de manera general qué es un algoritmo de reconstrucción y el que se ha utilizado en este proyecto.

3.2.6 Algoritmo de reconstrucción y concepto de sinograma

Como se ha explicado en el apartado 2.3.3, el algoritmo de reconstrucción empleado es el de Retroproyección Filtrada. A continuación ponemos el código que hemos desarrollado:

```

figure(NFigura), NFigura = NFigura + 1; clf
for ii=1:4
    aux = matrizProyecciones(:, :, ii);
    imagenOut = iradon(aux/max(abs(aux(:))), Pangle(1:numProyecciones));
    % imagenOut = iradon(matrizProyecciones/max(matrizProyecciones(:)), Pangle);
    subplot(2,2,ii)
    % axis([0 numProyecciones 0 tamProyeccion])
    % imshow(imagenOut)
    % pcolor(imagenOut), shading interp
    pcolor(log10(abs(imagenOut))), shading interp
    axis equal
    title(strcat('Reconstrucción con ', titulos{ii}))
end

```

Fig. 36: Representación.

Pero antes de implementar el algoritmo, lo que se debe hacer es unir todas las proyecciones que tenemos a partir de los parámetros que hemos utilizado. Todas estas proyecciones juntas forman el *sinograma* que es equivalente a la transformada de Radon, veamos un ejemplo:

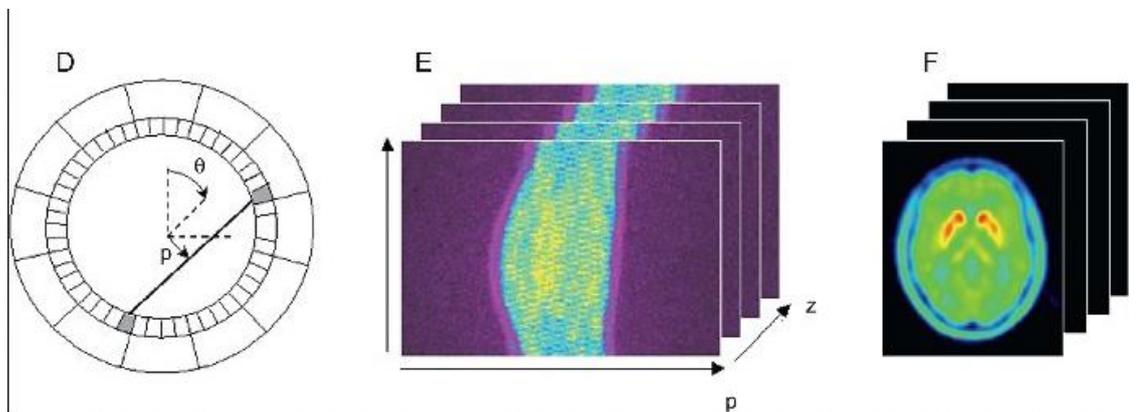


Fig. 37 : Proceso de cómo se crea el sinograma y a partir de él, la imagen tomográfica.

Podemos observar, como teniendo todas las proyecciones y juntándolas en una misma imagen, mediante un algoritmo de reconstrucción podemos llegar a formar la imagen tomográfica deseada. En nuestro caso, se han generado los sinogramas en Matlab de esta forma:

```

for ii=1:4
    subplot(2,2,ii)

    pcolor(Pangle(1:numProyecciones), 1:tamProyeccion ,matrizProyecciones(:, :, ii)), shading interp
    % colormap('bone')
    title(strcat('Sinograma de ', titulos{ii}))
    colorbar
end

```

Fig. 38 : Representación del sinograma

Y estos son los resultados obtenidos para la fisura pequeña y grande de la probeta:

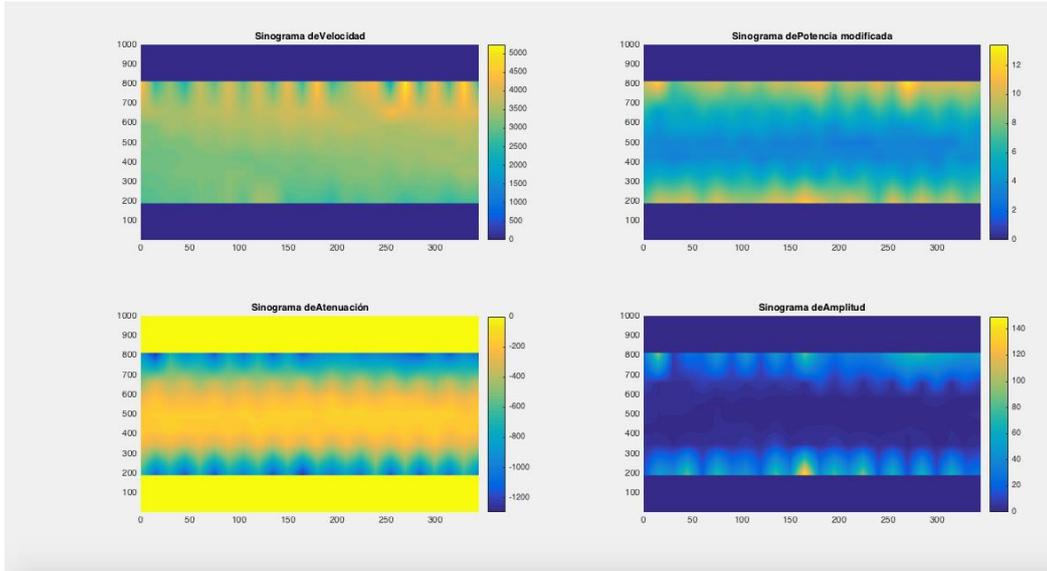
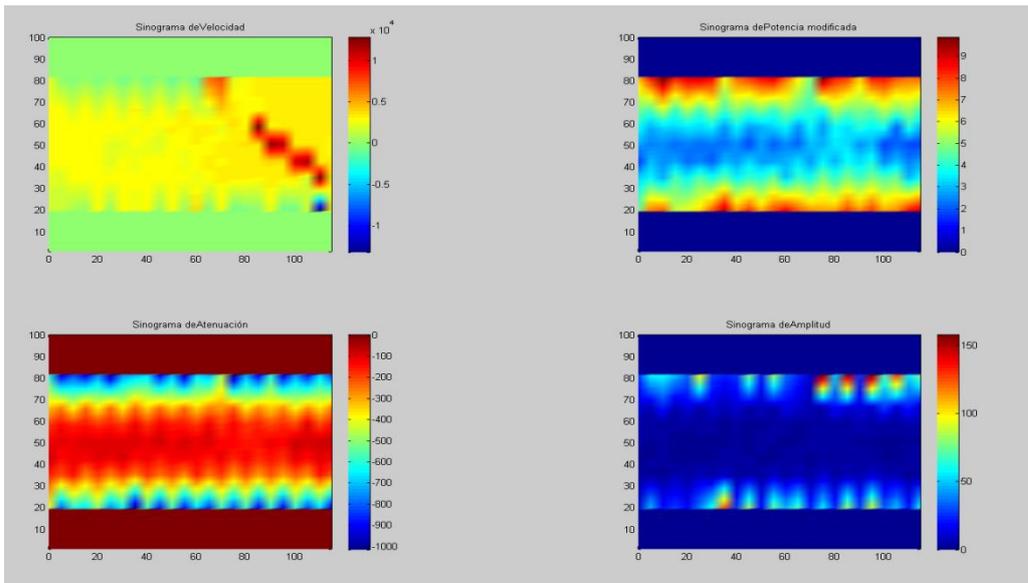


Fig. 39: Sinograma de la fisura pequeña.

Fig. 40: Sinograma de la fisura grande.

A partir del sinograma, ya podemos hacer la reconstrucción de la imagen final. En el siguiente apartado se comentarán los resultados obtenidos.



3.3 Resultados

Habiendo generado los sinogramas, ya podemos implementar el algoritmo de retroproyección filtrada. Estos son los resultados obtenidos al aplicarle la función *iradon* en Matlab:

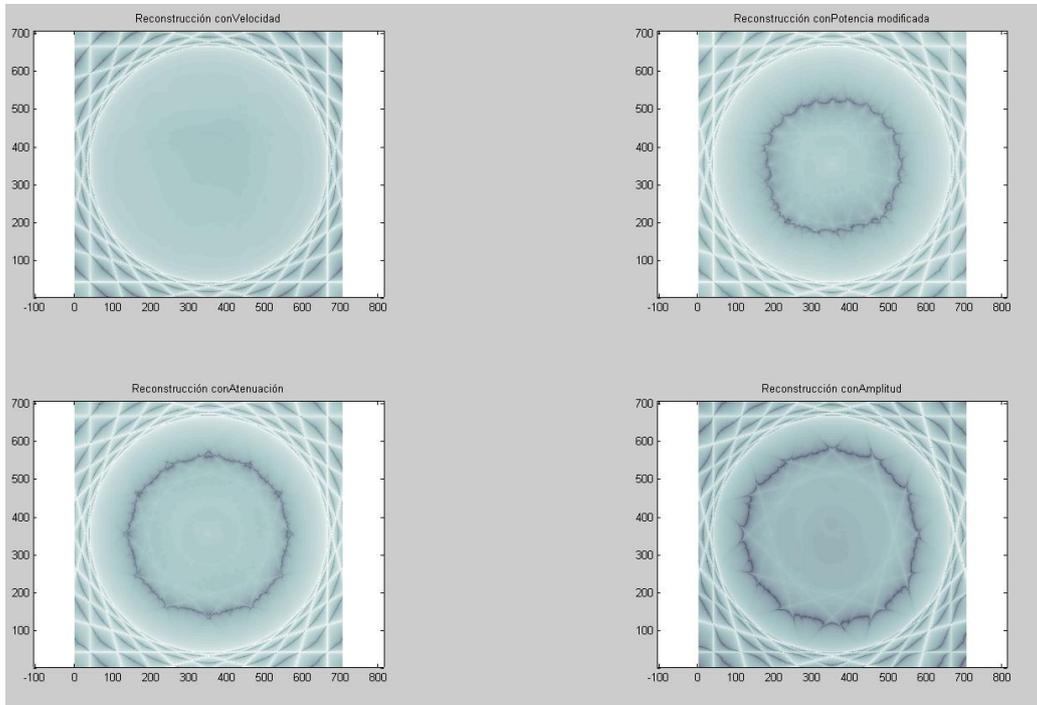


Fig. 41: Tomografía de la fisura pequeña.

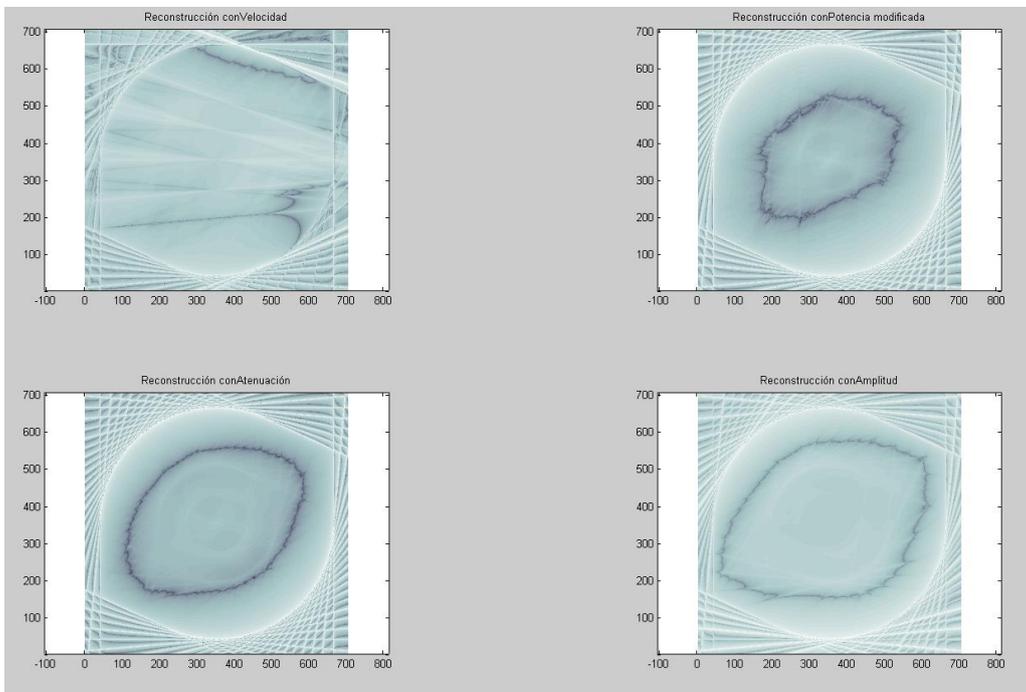


Fig. 42: Tomografía de la fisura grande.

Teniendo en cuenta la forma de la fisura de la probeta y donde está colocada, se puede decir que los resultados obtenidos no han sido los deseados. Vemos que una simple fisura

rectangular, se ve convertida como en una especie de elipsoide o “huevo” para ser un poco más exactos. Esto se debe a que no tenemos un número de medidas suficiente, es decir, tomando 13 posiciones diferentes desde cada posición del transmisor no obtenemos la suficiente información de la probeta como para poder reconstruirla de manera correcta. Y, además, esta reconstrucción está hecha con un número “redundante” de medidas, ya que en una primera toma de datos el transmisor solo se movió de la posición 1 a la 13, porque los demás rayos para las restantes posiciones del transmisor (media probeta aún) ya estaban trazados con las otras medidas.

Por tanto, como conclusión, se ha de decir que con un número limitado de medidas no es suficiente para obtener la cantidad de información necesaria para reconstruir una imagen con la resolución deseada. Para ello es necesario incrementar el número de medidas de forma más automatizada, en la que podamos recoger la información de toda la probeta desde una posición transmisor/receptor, así será posible poder utilizar esa información para la reconstrucción.

En el siguiente capítulo se verá el desarrollo experimental que se ha seguido para implementar el sistema de toma masiva de medidas.

Capítulo 4. Desarrollo Experimental

4.1 Introducción

Para este apartado, en el que había que implementar el sistema de toma masiva de medidas, se pensó en diferentes posibilidades. Un primer acercamiento para saber cómo iba a ser el sistema, fue pensar qué tipo de transductores se podrían utilizar, ya que eso condiciona todo el sistema. Hay diferentes técnicas para alcanzar este objetivo, pero como siempre, hay una que se adapta mejor a los problemas. Comentaremos las técnicas pensadas y explicaremos el porqué de cual hemos utilizado así como los inconvenientes que tienen las otras.

4.1.1 *Técnica basada en inmersión*

En primer lugar, se pensó en la técnica por inmersión, es decir, introducir la probeta en agua y utilizar transductores de inmersión los cuales utilizan el medio líquido como acoplador a la probeta. Pero no se disponía de transductores de inmersión, que los hay de este tipo, y debíamos crear un tipo de circuito hidráulico que proporcionara el agua para poder introducir la probeta en medio acuático.

Otra técnica utilizando el agua como medio conductor, es utilizando transductores por chorro de agua, en este caso no se introduce la probeta en agua si no que los transductores tienen como una carcasa que les proporciona un chorro de agua que choca con la probeta transmitiéndose de esta forma los ultrasonidos a través del chorro. Esta técnica también era algo complicada, ya que debíamos hacer un sistema para recoger el agua que utilizaban los transductores y volver a utilizarla.

Esta técnica se ajustaba bastante a nuestro ensayo por el hecho de que necesitamos un medio conductor estable para los ultrasonidos y el agua lo es. El problema es que la implementación de un sistema así lleva mucho tiempo y no era posible realizarlo para el presente TFG, así que quedó descartada.

4.1.2 *Técnica basada en transductores por contacto con aire*

En [3] que la siguiente técnica que pensamos que podía servir, es la utilización de transductores en contacto por aire, ya que no hay que utilizar agua, gel, vaselina, etc. En los últimos años se han ido desarrollando transductores, electrónica y procedimientos para la inspección ultrasónica sin contacto de diversos materiales (ultrasonidos en aire). Las mayores dificultades surgen por las enormes pérdidas de inserción (superiores a 100 dB), principalmente debidas a la desadaptación de impedancias acústicas transductor-aire-pieza. Los principales avances dieron como resultado una mayor sensibilidad, ancho de banda y resolución. Mediante distintas técnicas de optimización se han conseguido transductores de banda ancha, nuevos diseños para el enfoque de haces ultrasónicos y transductores activos en recepción, que mejoran significativamente la relación señal a ruido.

Por tanto, esta técnica evita utilizar cualquier tipo de medio conductor para los ultrasonidos lo que minimiza el coste del sistema. Pero hay algunos inconvenientes en esta técnica.

En primer lugar, el ultrasonido se atenúa en el aire proporcionalmente al cuadrado de su frecuencia, así como en la pieza inspeccionada, pero, sobre todo, existe una gran desadaptación de impedancias acústicas entre los materiales sólidos y el aire, que ocasiona las mayores pérdidas.

El aire a presión atmosférica y temperatura ambiental, tiene una impedancia acústica $Z_a=410$ Rayls, mientras que un *composite* típico (laminado de fibra de carbono o CFRP), presenta una impedancia acústica $Z_c=5.3$ MRayls, más de 104 veces superior. Una cerámica piezoeléctrica tiene una impedancia acústica típica $ZP=35$ MRayls y, el agua, $ZW=1.5$ MRayls. Como se ha citado anteriormente, los coeficientes de transmisión T y de reflexión R en la interfaz entre dos medios con impedancias acústicas Z_1 y Z_2 son:

$$T = \frac{2Z_2}{Z_1+Z_2} \quad (41)$$

$$R = \frac{Z_2-Z_1}{Z_2+Z_1} \quad (42)$$

De aquí se obtienen las pérdidas de inserción mostradas en la figura 43 para inspecciones por transmisión de CFRP con acoplamiento agua o aire, sin considerar otras pérdidas debidas a la atenuación.

Se observa que, utilizando agua como acoplante, las pérdidas de inserción se sitúan en el entorno de los 20 dB. En contraste, tales pérdidas aumentan hasta unos 160 dB cuando el medio acoplante es el aire.

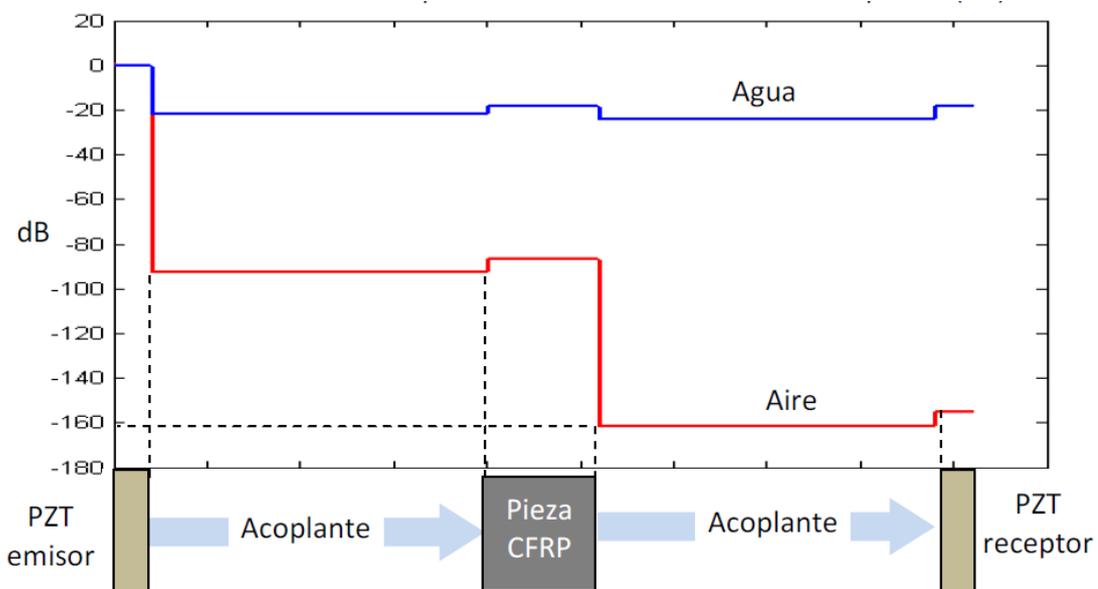


Fig. 43: Pérdidas de inserción utilizando agua y utilizando aire.

En el laboratorio, disponíamos de transductores de este tipo y los probamos para comprobar si se adaptaba a nuestras necesidades.

Un pequeño ensayo para ver si funcionaba, fue utilizar una probeta rectangular y colocar el transductor, mitad con aire y la otra mitad en la probeta. Más o menos de esta forma:

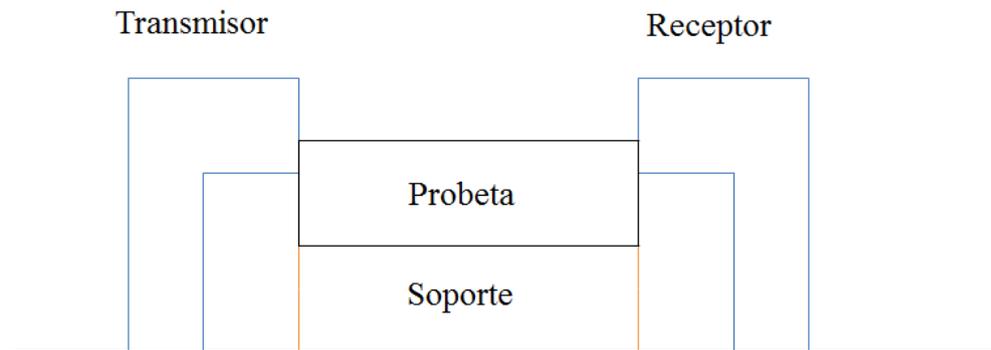


Fig. 44: Ensayo de prueba para transductores en contacto con aire.

Mediante este método debíamos transmitir la señal y ver en recepción una parte de la señal igual, es decir, sin atenuación ni retardo (la parte de la señal que va por el aire) y una parte atenuada y retrasada debida a la propagación por la probeta, tal que así:

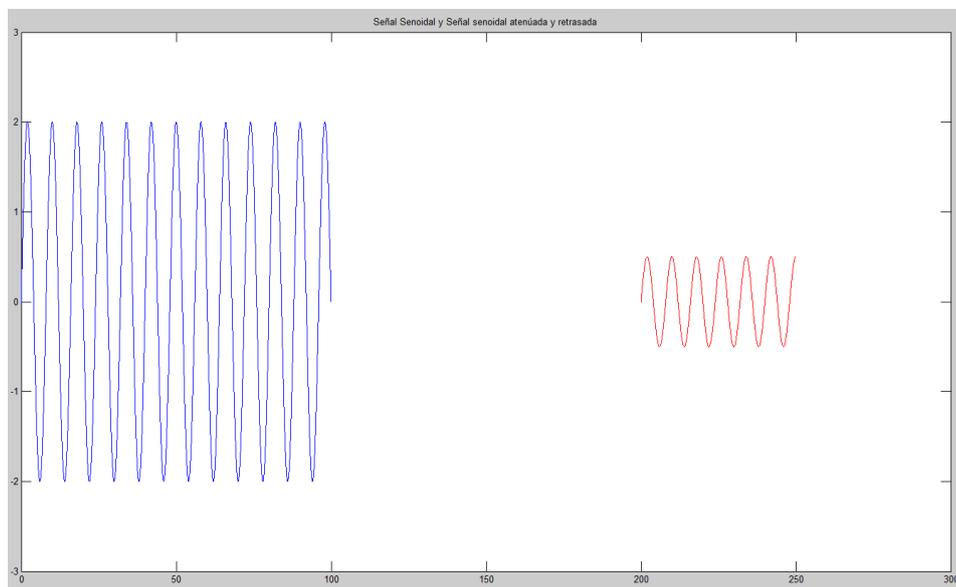


Fig. 45: Simulación de probar los transductores por aire.

Después de varias pruebas y no conseguir poder ver la parte de señal que pasa por el objeto a estudiar se decidió descartar esta técnica.

4.1.3 Técnica basada en transductores por contacto directo

Viendo que se acababan las opciones, en el laboratorio teníamos transductores que funcionan con contacto directo con el material, sin ningún tipo de medio conductor. Esto, agilizaba mucho la implementación del sistema de toma masiva de datos, por lo que se decidió probarlo. Funcionó, se probaron los transductores con la señal chirp y trabajaban bien en toda la banda de frecuencias. Por tanto, se utilizó esta técnica ya que el sistema se podía realizar de una forma más sencilla. Los transductores utilizados son los mostrados a continuación:



Fig. : Transductores en contacto directo.



Fig. : Imagen de la flecha que nos da la dirección en la que emite el emisor y recibe el receptor.

Están pegados al objeto bajo estudio y emiten en dirección de la flecha. La flecha se debe mantener apuntando en dirección hacia el centro de la probeta y el transductor va girando emitiendo y recibiendo. Se trata de un procedimiento muy útil para nuestro ensayo.

Una vez elegida la técnica a emplear para nuestro sistema, se empezó a pensar cómo iba a serlo.

Por tanto, vamos a continuar explicando cómo se ha realizado el sistema desde el diseño hasta su implementación.

4.2 Desarrollo de sistema de acoplamiento

Viendo la forma y el funcionamiento que tienen los transductores, en primer lugar se pensó en hacer un doble anillo que estuviera alrededor de la probeta y que sirviera de sujeción para los transductores, más o menos de esta forma:

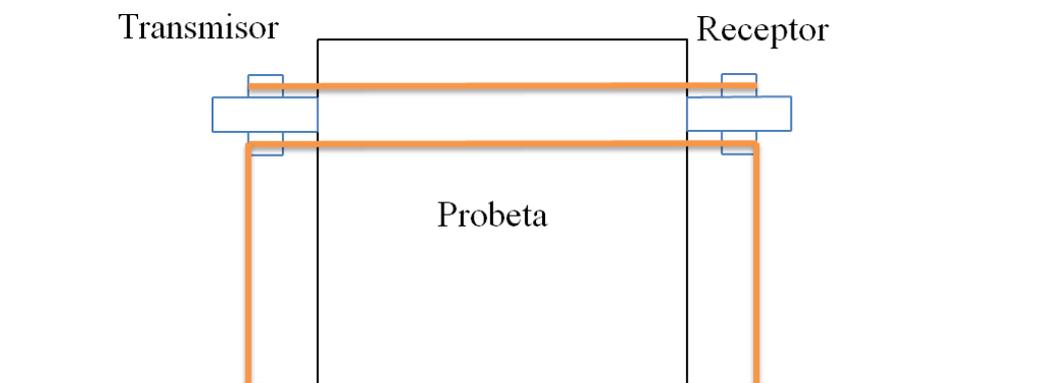


Fig. 45: Sistema con doble anillo.

A pesar de ser una opción viable, dejar los transductores en direcciones opuestas con una buena sujeción y que además pivotarán sobre la probeta complicaba su implementación. Pensamos en un prisionero a 90°, pero la idea no acababa de ser la óptima. De este modo, se planteó mover la probeta dejando los transductores fijos.

Bajo este punto de partida se diseñó unas placas con huecos del diámetro de los anclajes del transductor, que estuvieran a una distancia suficiente para rodar en la probeta junto con el hueco para la probeta. Esto se sujetaría en una estructura externa poder posicionar las placas a la altura oportuna. Para esto, primero se crearían las placas en un entorno de diseño y luego las imprimiríamos con impresora 3D con un material de densidad normal tiene y rigidez lo suficientemente grande como para sujetar bien los transductores.

Para el diseño de las placas se utilizó Solid Works, es un software CAD (diseño asistido por computadora) para modelado mecánico en 3D, y no es tan complicado como otros y viendo algún tutorial se pudo realizar el diseño, ya que no era muy complicado. Este fue el resultado:

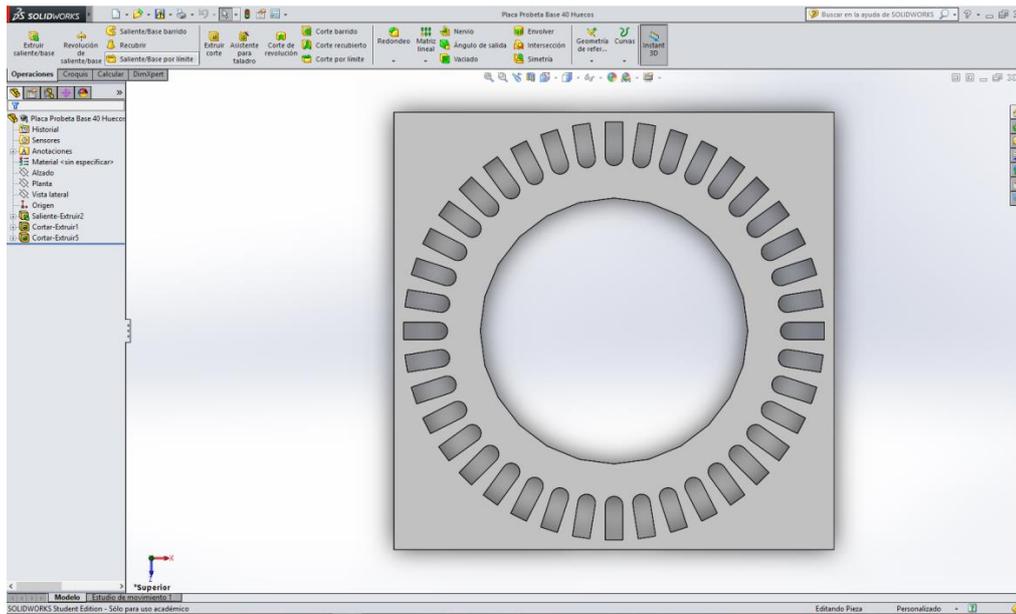


Fig. 46: Placa Solid Works.

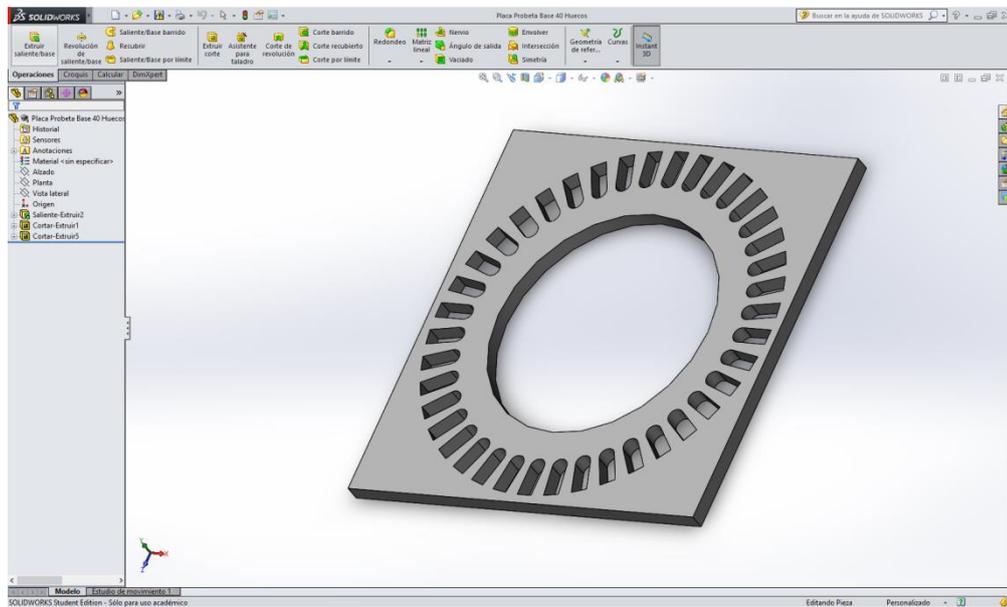


Fig. 47: Pieza Solid Works (para que se vea el relleno).

Y las medidas reales, para que nos encajara todo, son estas:

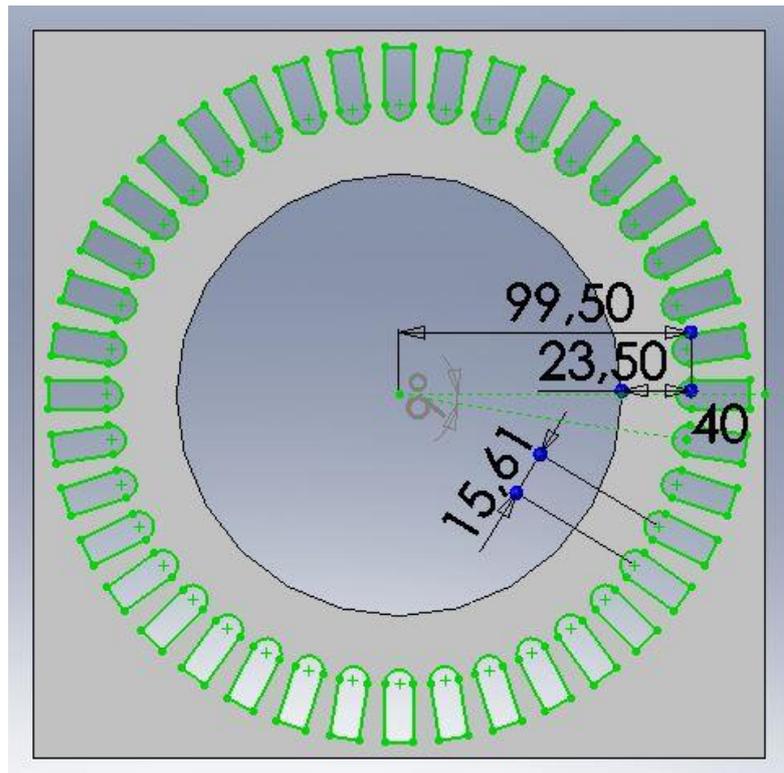


Fig.48: Medidas reales de la placa.

Las medidas en la imagen están en milímetros. Simplemente se realizó un cuadrado de 25x25 cm y en el centro un círculo de radio 15.2 cm, ya que la probeta es de 15 cm pero no es cilíndrica del todo, en algunos lugares tiene imperfecciones, sobre todo en las juntas que sobresalen un poco y en ciertos puntos de la probeta tenemos 15.1 o 15.2 cm por eso se le dio un poco más de margen. Luego, los 2.35 cm desde donde acaba la probeta hasta que empiezan las posiciones de los transductores, son la mitad del diámetro de lo que mide el transductor que son 4.7 cm. La idea era que el transductor estuviera en la parte circular de su espacio para posicionarlo, y le dejamos un margen por detrás en el caso de que se encuentre con alguna imperfección. La distancia de 9.950 cm es la suma de la mitad de la probeta (7.2 cm) más la distancia citada antes. Y la distancia restante, era la que más nos preocupaba por el tema de la robustez de la placa. En un primer momento, se diseñó esta misma placa para 30 posiciones, dejando una distancia entre los centros de las medias circunferencias de las posiciones de 2.08 cm, lo que nos pareció un poco grande, es decir, que podríamos forzarla un poco más.. En las pruebas preliminares contábamos con 24 en las que se enviaba y recibía un rayo. Ahora, serían 30 posiciones obteniendo la información de toda la probeta en una misma tirada. 30 posiciones nos parecían muchas en el momento que se comenzó con el diseño, pero como vimos que se podía acortar la distancia, se hicieron pruebas para añadir más posiciones hasta llegar a 40, con una distancia de 1.58 cm entre cada posición.

Se le dio un grosor de 1 cm, lo cual era suficiente,

Los resultados se muestran a continuación:

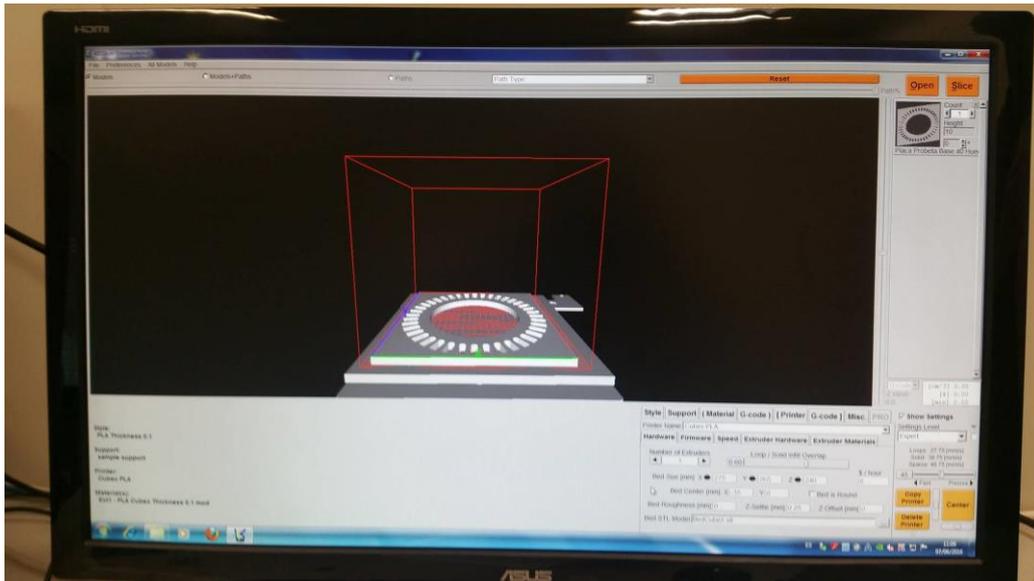


Fig. 49: Software de la impresora 3D, en el que podemos ver cómo quedaría la placa al final de la impresión.

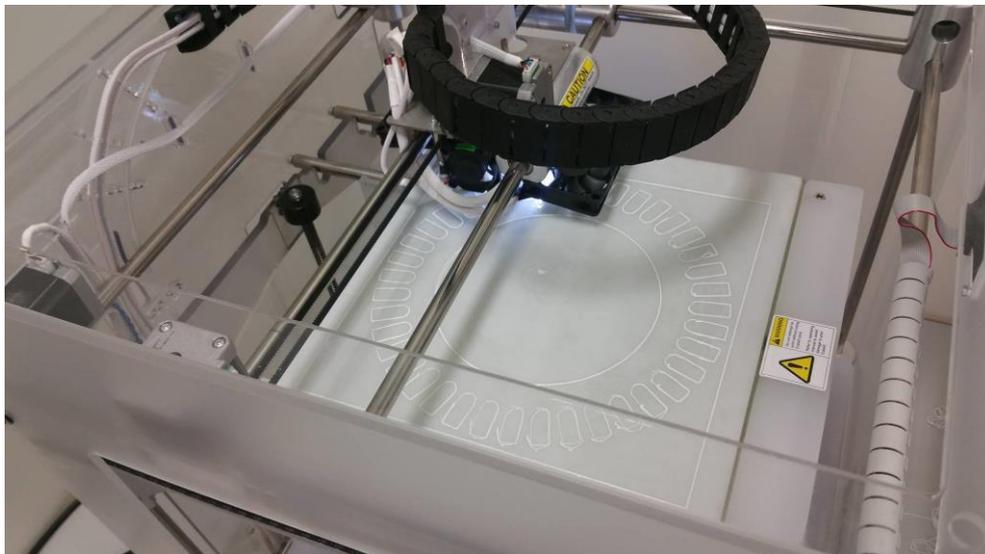


Fig.50 : Proceso de impresión. Primero dibuja los huecos.

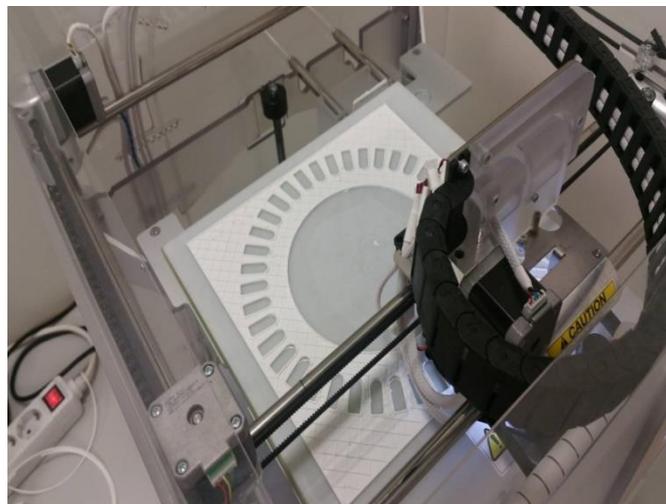


Fig. 51: Proceso de impresión. Luego rellena lo restante.



Fig.52: La placa recién impresa. Proceso de despegarla de la base de impresión.

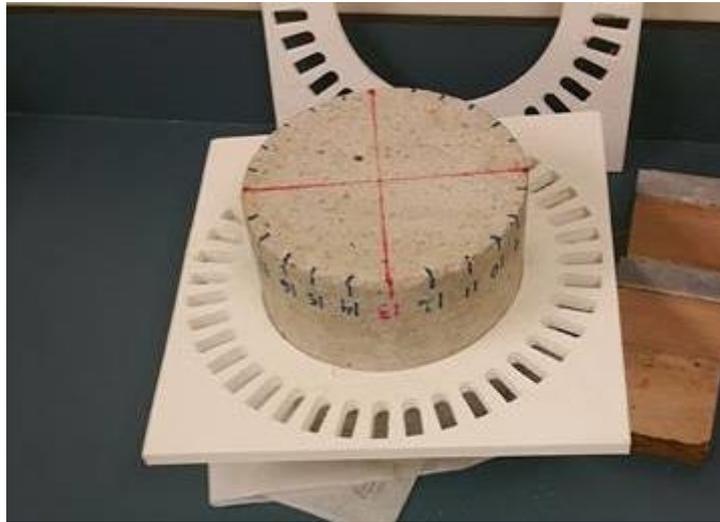


Fig 53: La placa final y la probeta.

Este proceso fue largo, ya que la impresora tardó en imprimir 7 horas cada placa a lo que hubo que sumar dos o tres intentos de impresión. Finalmente el resultado es bueno, ya que es robusta y de un material que, en nuestro caso, nos ha sido muy útil ya que es manejable y nos ha permitido adaptar ligeramente las medidas con un pequeño lijado.

Teniendo ya las placas impresas, se creó una estructura para ellas para poder posicionarlas en lugar apropiado y ser capaces de recoger la información que queremos de la probeta. Lo más sencillo era hacer un cubo, mediante perfilería de aluminio y con unos ángulos de 90° que nos permitan variar la altura.

El cubo debía ser de las dimensiones de la placa, por lo menos en longitud y ancho para que encajara, es decir, 25x25 cm. Y el alto también lo hicimos de 25 cm ya que la probeta mide unos 17 cm aproximadamente..

A continuación se muestran las imágenes del proceso:

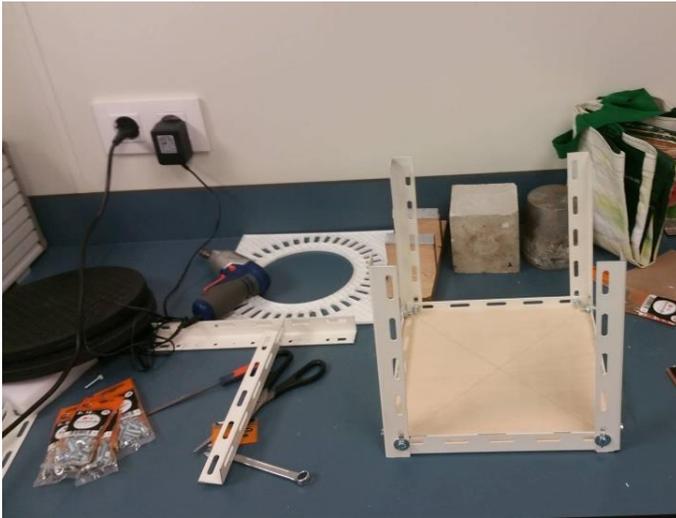


Fig.54 : Proceso de montaje de la estructura.

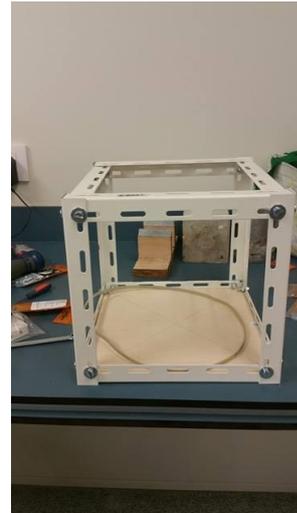


Fig.55 : Estructura final.

4.3 Desarrollo de adquisición de forma sincronizada

Una vez el sistema estaba montado, el siguiente paso era montar el sistema de medida y adquisición. Como se ha citado anteriormente en la explicación del montaje del sistema, cada posición del transductor le daremos una vuelta entera a la probeta.

Por tanto, teníamos que crear un sistema que fuera adquiriendo mientras la probeta fuera girando, y de alguna forma saber que ha dado la vuelta entera y parar. Para ello utilizamos un encoder de posición, a través del cual podremos definir el límite de la medida. Y por otra parte, necesitamos un osciloscopio para poder recoger la señal que mandamos y la que obtenemos en recepción.

En primer lugar, se verá la parte del encoder de posición y más tarde se hablará de la parte del osciloscopio, ya que no se ha utilizado el mismo que en el capítulo de Medidas Preliminares.

Por tanto, podremos acceder a la información del encoder mediante un Arduino conectado a él. Y mediante el sistema de poleas que hemos implementado, la probeta girará y en consecuencia girará el encoder, en el cual se ajustarán el número de vueltas que corresponden a una vuelta entera de la probeta.

El encoder utilizado es este:

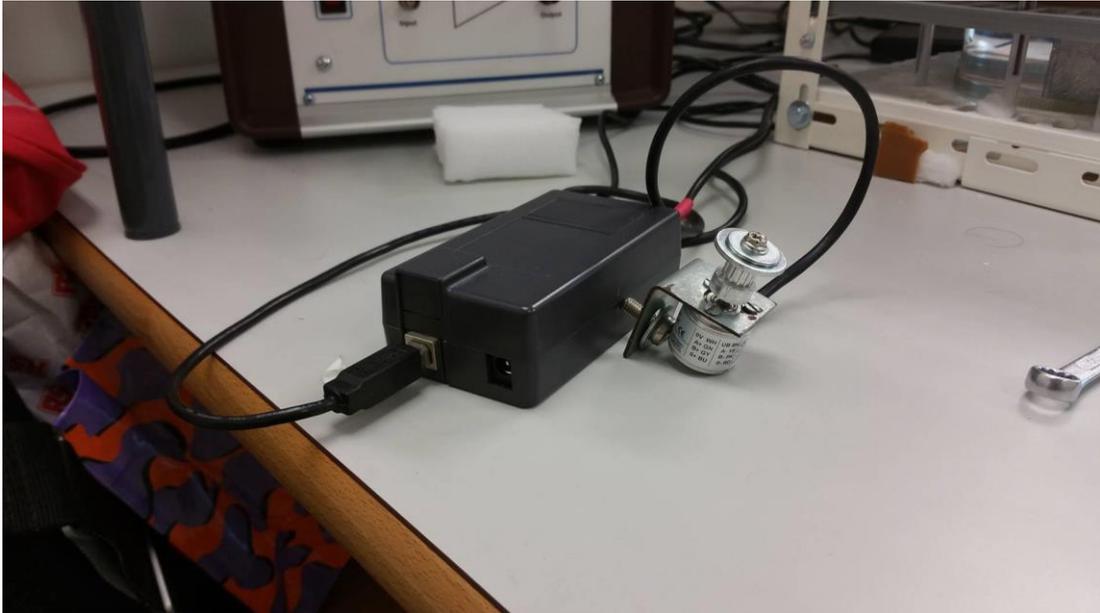


Fig.56 : Encoder

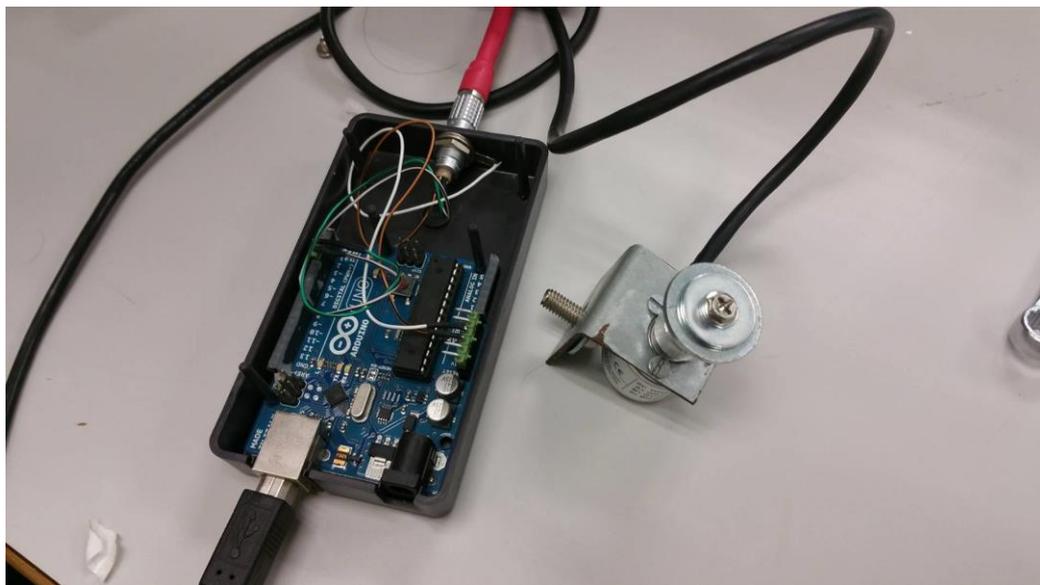


Fig.57 : Encoder conectado a arduino.

El cual se ha introducido en el sistema de esta forma:

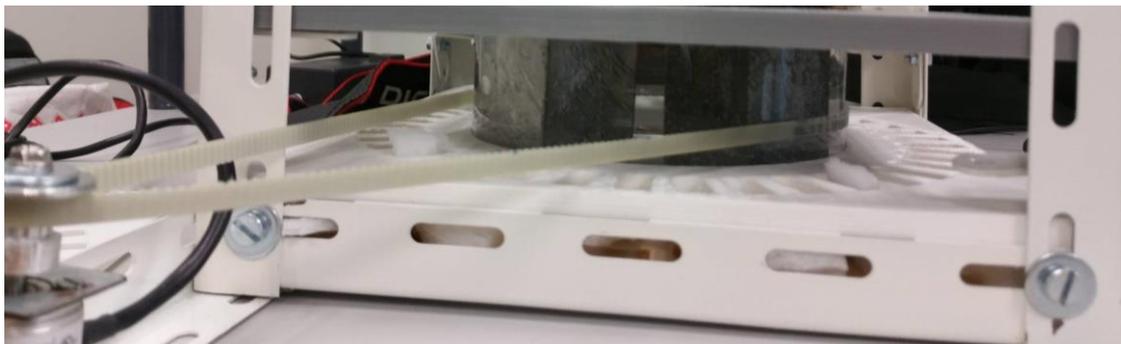


Fig.58 : Enganche del encoder a la probeta.

De este modo, dándole vueltas a la probeta iría girando el encoder y realizando medidas cuando nosotros lo programemos.

Para la adquisición de la información del encoder se ha generado una pequeña interfaz en Matlab. En la que tenemos diversas opciones:

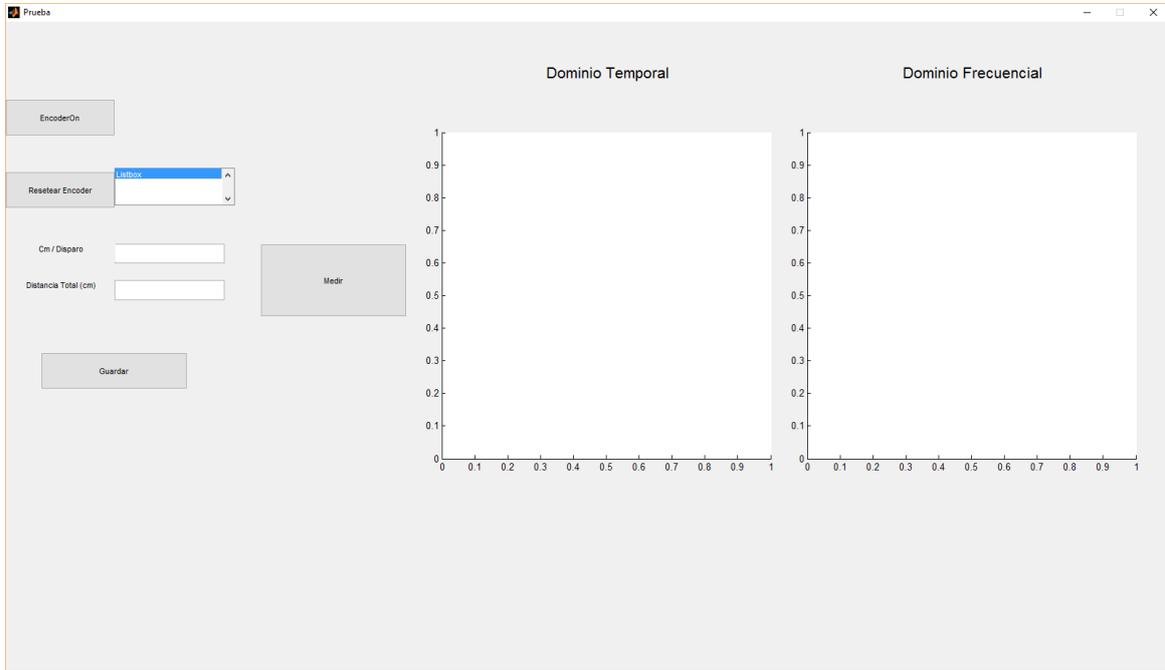


Fig.59: Interfaz para adquirir desde el encoder y osciloscopio.

Lo primero que se debe hacer al conectar el encoder y abrir la interfaz, es abrir la conexión con él, que lo hacemos desde el puerto serie:

```
% --- Executes on button press in EncoderOn.
function EncoderOn_Callback(hObject, eventdata, handles)
% hObject    handle to EncoderOn (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
    global ard
    disp('Hola')

    delete (instrfind({'Port'}, {'COM3'}))
    ard = arduino ('COM3');

    encoderAttach(ard,0,2,3);
    encoderStatus(ard,0)
    encoderReset(ard,0);
    disp ('Encoder is RESET to 0')

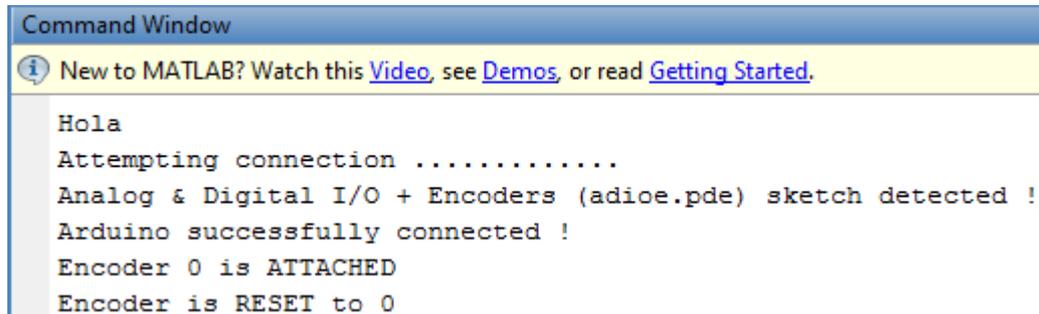
    i=1;

    pinMode(ard,8,'output');
    pinMode(ard,9,'output');

    Inicializar_variables();
end
```

Fig.60 : Inicialización de la conexión de Arduino

Este paso se podría haber realizado mediante el programa de Arduino, pero instalando unas librerías en Matlab es mucho más cómodo hacerlo en el mismo programa. Cuando conectamos el encoder y creamos la conexión y todo va bien debe salir este mensaje:



```
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
Hola
Attempting connection .....
Analog & Digital I/O + Encoders (adioe.pde) sketch detected !
Arduino successfully connected !
Encoder 0 is ATTACHED
Encoder is RESET to 0
```

Fig. 61: Conexión con Arduino

A partir de aquí ya podemos trabajar con él. Veamos que opciones nos permite esta interfaz:

- Resetear encoder: Por si ha habido algún problema o simplemente para asegurarse que todo en el Arduino está a 0, al presionar este botón, en el cuadro de texto saldrá el siguiente mensaje:



Fig.62 : Reset Encoder.

- Darle los Cm/Disparo y la distancia total en Cm, con lo cual establecemos el número de medidas que queremos. Por ejemplo si ponemos 1 Cm/Disparo y 10 Cm en total, se harán 10 medidas.
- Representación en el dominio temporal y frecuencial de la señal adquirida del osciloscopio.
- Medir: este botón accede tanto al encoder como al osciloscopio, ya que calcula la distancia para cada medida y adquiere del componente creado en Matlab para el osciloscopio. Aquí se puede ver dentro de la función para este botón, en este caso está hecho para que cada 200 vueltas haga una medida:

tiempos, o disparo, o configurar cualquier cosa que si podíamos hacer en los osciloscopios tradicionales.

Para ello es necesario crear una interfaz para poder visualizar que estamos transmitiendo, recibiendo, etc. Para esto se ha utilizado Matlab e igual que antes, mediante la herramienta GUI, se ha implementado una interfaz para el dispositivo:

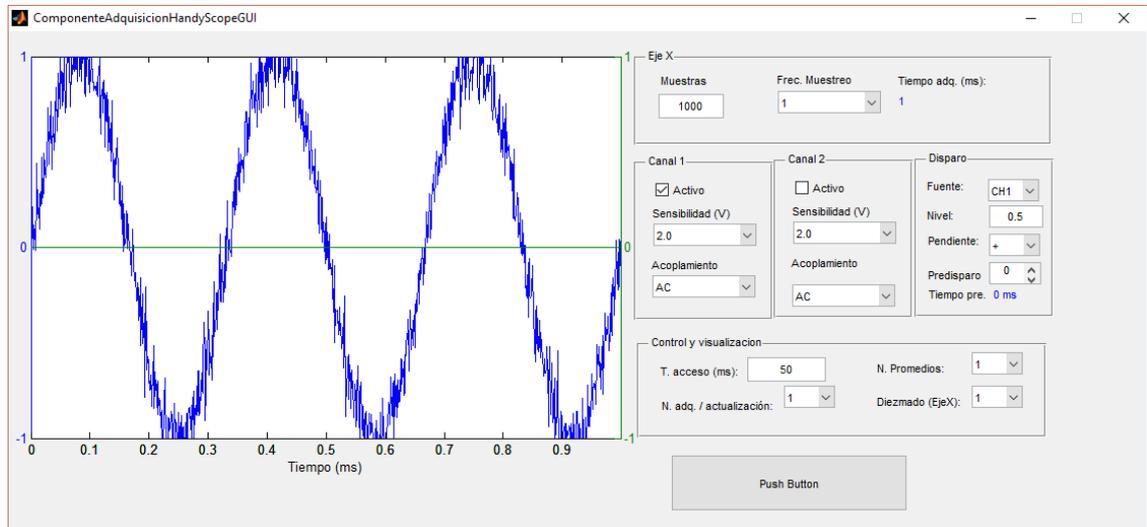


Fig. 65: Interfaz para HandyScope.

Como se puede ver, tenemos los dos canales que tenemos en el osciloscopio, un menú para configurar el disparo, un menú para modificar el número de muestras y frecuencia de muestreo y por último, un menú para variar el tiempo de acceso, n° de promedios y demás que no serán útiles para el sistema. También cabe destacar, que si encuentra un dispositivo conectado trabaja con ese dispositivo y si no, trabaja con datos simulados.

Este programa estaba ya desarrollado por el grupo de investigación, ya que este osciloscopio ha sido utilizado para multitud de ensayos y hubo la necesidad hacerlo.

Este código se ha introducido en el del encoder, pudiendo acceder desde el código del encoder a este. Antes se ha dicho que se explicaría la función *getDataFromInstrumentEncoder* que es la función que nos permite recoger los datos que genera el osciloscopio.

Para esto, creamos un objeto global en el GUI del encoder:

```

function Inicializar_variables()
% Preparación de los datos para adquisición periódica
% Todo esto lo podríamos poner para que estos valores fueran establecidos
% desde la GUI
global useSimulatedData
global timerParams
global distParams
global timerObj
global osciloscopio
global acquiringData
global data_save
global fsAux
global xSonido
global x_save
global fs
osciloscopio = ComponenteAdquisicionHandyScopeGUI;
acquiringData = false;
useSimulatedData = 0;
timerObj = []; %%Creamos el objeto para el timer
data_save = [];
x_save = [];
fs = 0;

```

Fig.66 : Declaración objeto global.

A esta función se le llama cuando establecemos conexión con el encoder, en la cual se inicializan todas las variables globales del código y a parte, se llama al componente del HandScope. En esa llamada, tendremos una estructura llamada osciloscopio que dentro tiene estas variables:

```

osciloscopio =
ComponenteAdquisicionHandyScope handle
Properties:
  NAdquisiciones: 168
  constantes: []
  sDLL: 'HS3'
  triggerTimeOut: 0.5000
  timeIni: 666557367558
  vectorDescarga: [1x2 struct]
  activo: 0
  semaforoLibre: 1
  NCanales: []
  NPuntos: 1000
  xData: [1x1000 double]
  yData: [1x1000 double]
  fs: 1000000
  yDataBuffer: [1x1000x2 double]
  xUnits: 'Seconds'
  yUnits: 'Volts'
  configuracion: [1x1 struct]
  nombre: 'Datos simulados'
  NCanalesMax: 2
  value: []
Methods, Events, Superclasses

```

Fig.67 : Estructura con las variables del componente de HandyScope.

Ahora que sabemos cuáles son las variables que hay en la estructura del osciloscopio, y cuales nos interesan para adquirirlas.

En la función *getDataFromInstrumentEncoder* tenemos:

```

if distParams.num == 0
    % timerParams.data = zeros(timerParams.numTotal,length(yData)); % Creamos la
    distParams.data = NaN*ones(distParams.numTotal,length(osciloscopio.yData));
    distParams.fs = NaN*ones(1,distParams.numTotal);
    distParams.fs(1) = fs;
end

% Incrementamos el contador de adquisiciones
distParams.num = distParams.num + 1;

% Almacenamos los datos
distParams.data(distParams.num,:) = osciloscopio.yData; % Almacenamos los datos

% Verificamos que la frecuencia de muestreo no sufre ningún cambio
distParams.fs(distParams.num) = osciloscopio.fs;

```

Fig. 68: Adquisición de datos del osciloscopio.

```

axes(handles.hAxesRaw);
% plot(xData,yData);
plot(osciloscopio.xData,osciloscopio.yData);
grid;
xlabel(osciloscopio.xUnits,'FontSize',14); ylabel(osciloscopio.yUnits,'FontSize',14);

%
% axes(handles.hAxesMath,'FontSize',14);
axes(handles.hAxesMath);
[freq,fftdata] = powerSpectrum(osciloscopio.xData, osciloscopio.yData);
plot(freq,fftdata);
grid;
xlabel('Frequency (Hz)','FontSize',14); ylabel('Amplitude','FontSize',14);

```

Fig.69 : Adquisición de los datos del osciloscopio para representarlos.

A continuación se presentan algunas medidas comprobando el funcionamiento, en este caso son datos simulados:

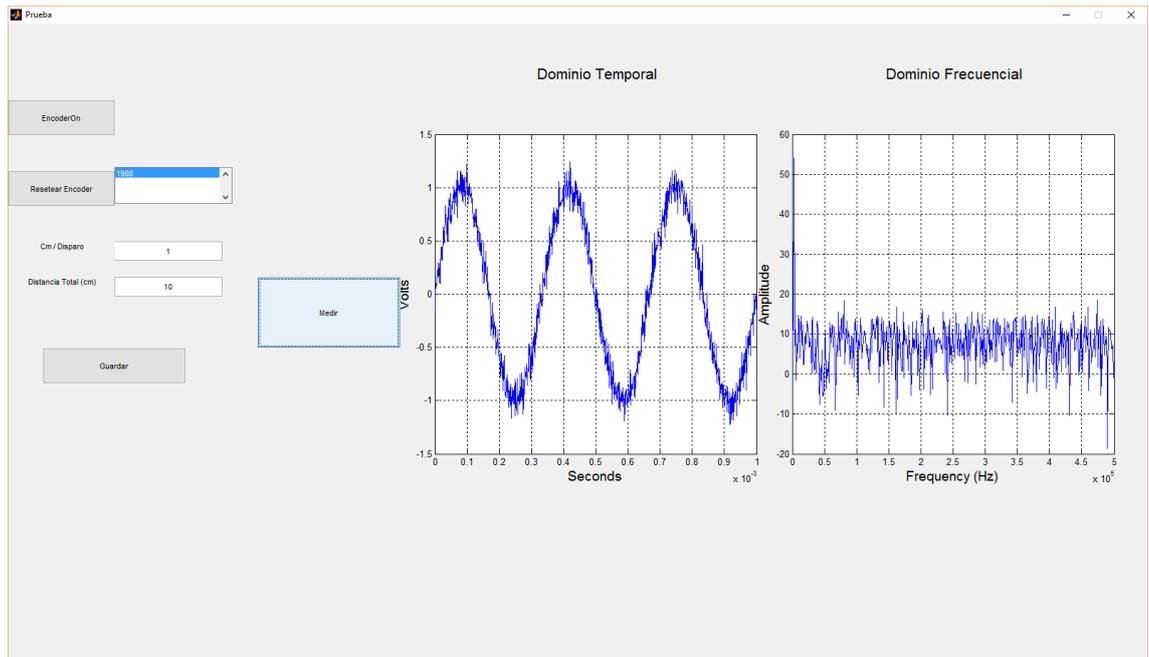


Fig.70 : Medida encoder.

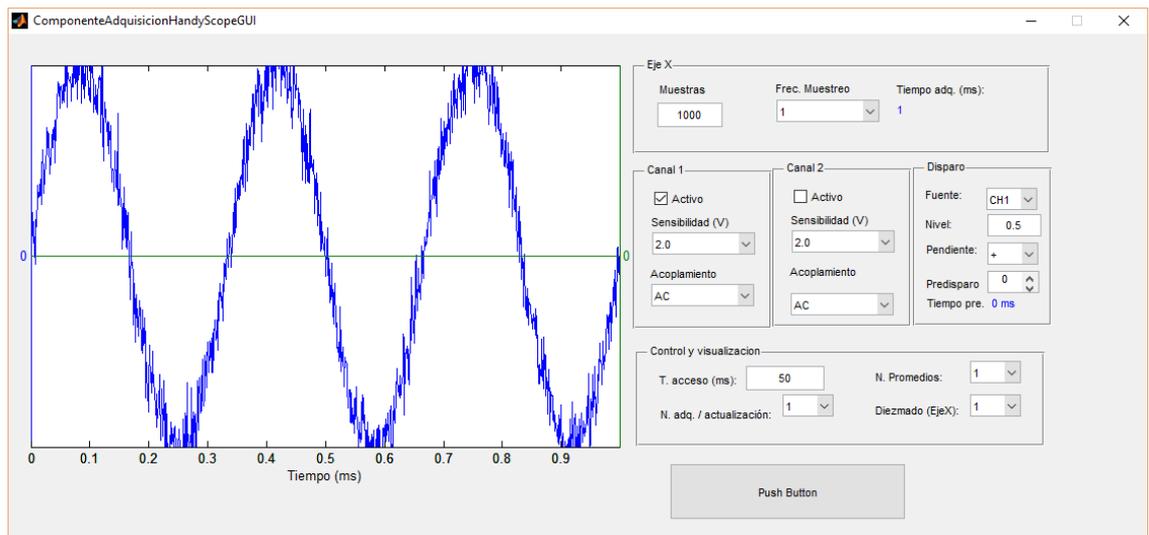


Fig.71 : Medida con osciloscopio.

4.4 Resultados preliminares con el sistema de toma de medidas masivo

Cuando el sistema estaba acabado, al colocarlo todo y hacer pruebas, los transductores no quedaban completamente pegados a la probeta y no podían atravesar las probetas de cemento. Al no tener la suficiente presión la señal se atenúa en exceso y sumando además que la probeta estaba seco lo que aumentaba aún más la atenuación y obteniendo una relación señal a ruido muy baja para obtener parámetros confiables.

Como alternativa, se utilizó una probeta de cristal llena de agua, tipo un vaso grande, a la que le añadimos dos probetas ,de diferentes materiales para poder detectar algún cambio de medio:



Fig.70: Probeta utilizada.

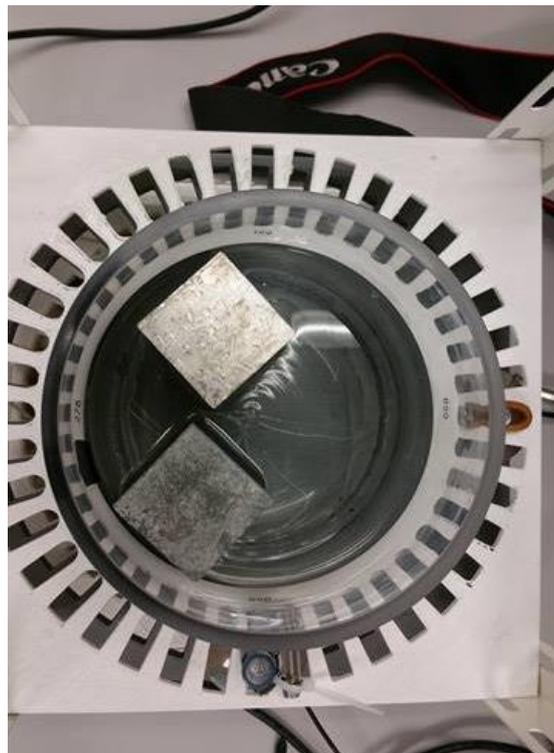


Fig.71 :Probetas introducidas para crear un cambio de medio.

El agua es un buen conductor de los ultrasonidos por lo que facilita transmisión. A parte se cubrió la probeta con una goma que sirve de adaptador para los transductores, los cuales giran mucho mejor:

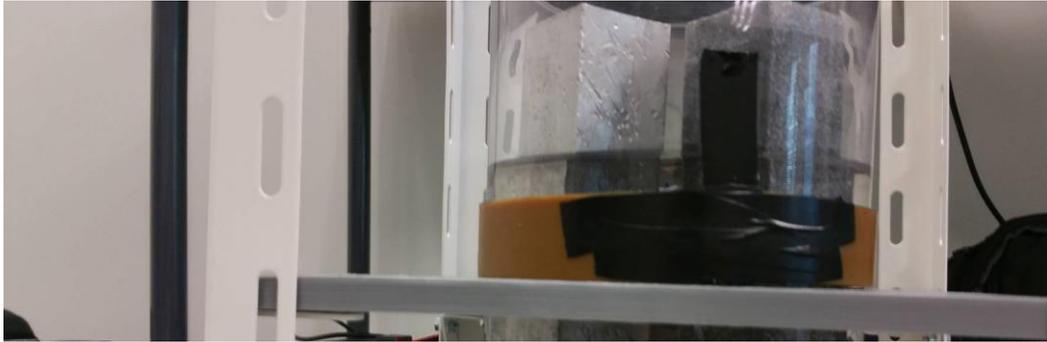


Fig.72 : Goma ayudante para la conducción.

Como se ha citado anteriormente, en el apartado 3.1.3 los transductores deben estar dirigidos siempre al centro de la probeta, lo que hizo que tuviéramos que fijar mediante una abrazadera y un tope con la misma goma utilizada como adaptador para fijar los transductores correctamente:



Fig.73 : Abrazadera utilizada para la sujeción del transductor.

Como ultima idea para el sistema, se puso un tope que estableciera el comienzo y el final del giro de la probeta. Como la base del sistema es de madera de contrachapado, centramos la probeta y le pusimos un tornillo:

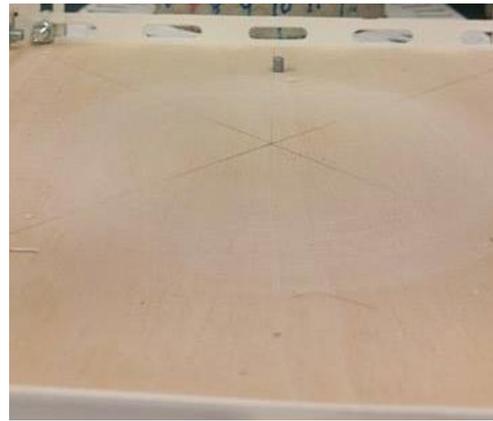
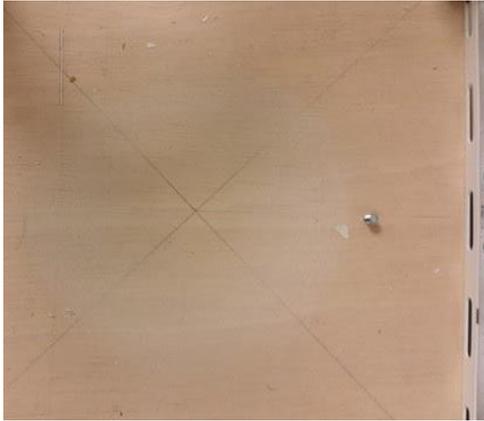


Fig.74 a) y Fig.74 b) : Tornillo tope.

Y para esto, se diseñó una pequeña base en SolidWorks con un tope y se pegó a la probeta con termosellado:



Fig.75 : Base con tope para la probeta.

Habiendo realizado este tope, se numeró la placa donde se sujetan los transductores, de modo que donde está el tornillo fuera 0° y permitiera saber el comienzo para realizar cada medida:



Fig.76 : Posiciones en grados de la placa

4.5 Toma de medidas

4.5.1 Cambio de concepto de toma de medidas

En este caso, la toma de medidas se realiza de una forma distinta a la descrita en el Capítulo 3 ya que se ha cambiado la forma de tomar las medidas. Ahora, los transductores están fijos y movemos la probeta, por tanto el concepto cambia respecto a lo descrito en dicho capítulo.

En la primera toma de medidas utilizamos la toma de medidas basada en una estructura de reloj, en la cual el transmisor se movía por todas las posiciones posibles, y el receptor, para cada posición del transmisor, recorría todas las posiciones desde 90° a 270° si el transmisor estuviera en 0° . Ahora todos esos movimientos del transmisor y receptor se hacen realiza cuando la probeta gira, es decir, que teniendo el transmisor y el receptor en una determinada posición girando la probeta calculamos todos los rayos que pasan por ella con la misma separación angular.

En nuestro sistema ahora tenemos estas posiciones:

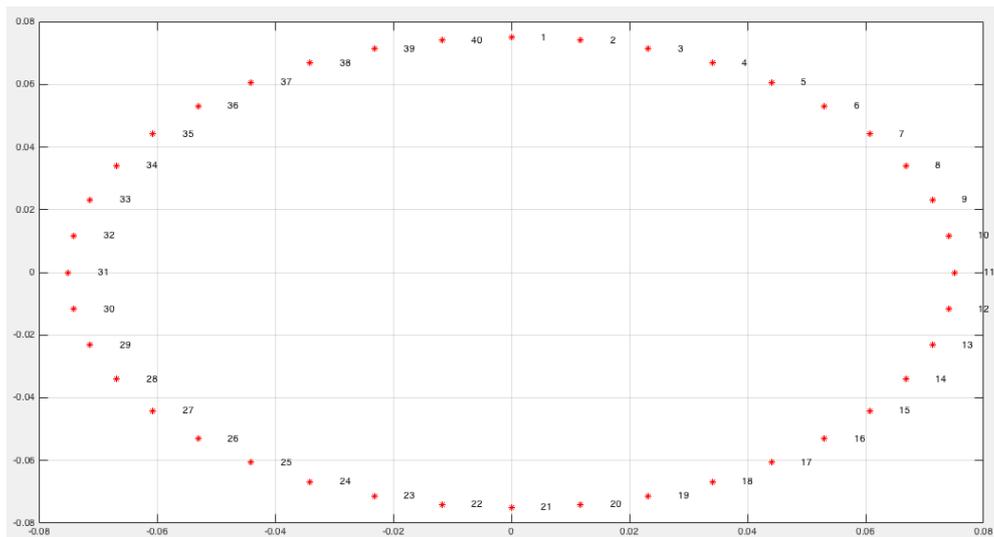


Fig.77 : Nuevos puntos de medida.

Para la toma de medidas, pondremos el transmisor en la posición 1 o 0° , y moveremos el receptor desde la posición 31 o 90° hasta la posición 11 o 270° , que a priori es lo mismo que hicimos en la primera tanda de medidas, pero teniendo 21 posiciones que recorrer mientras que antes teníamos 13.

Pero, hay que tener en cuenta que ahora para cada posición transmisor-receptor no cogemos un rayo sino que cogemos información de toda la probeta, es decir, antes cuando teníamos los 24 puntos en la probeta para la misma separación angular entre transmisor y receptor nos movíamos 15 grados por lo que cogíamos pocos puntos.

Ahora, se ha configurado el encoder de posición para que realice una medida cada 2° , por lo que para cada posición realizamos 180 medidas como se puede ver en uno de los archivos que corresponden a una medida:

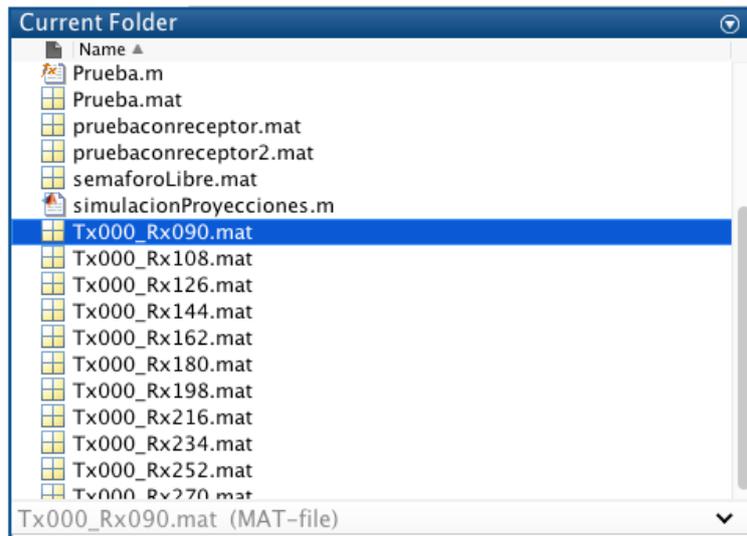


Fig.78 : Ficheros generados en la primera toma de medidas. Están numeradas por su posición en grados.

Y cada uno de estos ficheros contiene las estructuras *distParams* y *timerParams*, generadas en la interfaz del encoder, pero la que de verdad nos importa es *distParams* :

```
% Parámetros espaciales
distParams.distTotal      = 0; % Distancia total en cm
distParams.resDist       = 0; % Resolución en distancia
distParams.vectorDist    = 0; % Vector de distancia
distParams.data          = []; % La matriz de datos, se creará en la primera llamada
distParams.fs            = [];
distParams.num           = 0; % Numero de medidas tomada
distParams.numTotal      = 0; % Número total de medidas
distParams.tol           = 10; % Tolerancia en tanto por cien. Implica que soportamos un error de +/- tol*resDist/100
distParams.vectorPosEncoder = []; % Posiciones del encoder que hay que leer
distParams.vectorPosEncoderLeida = []; % Posiciones del encoder realmente leida
distParams.vectorPosMedidas = []; % Controla las posiciones que hemos medido
distParams.PosReceptor   = []; %Nos dice la posición del receptor en grados
```

Fig.79 : Estructura distParams.

Por tanto en el archivo guardado tenemos:

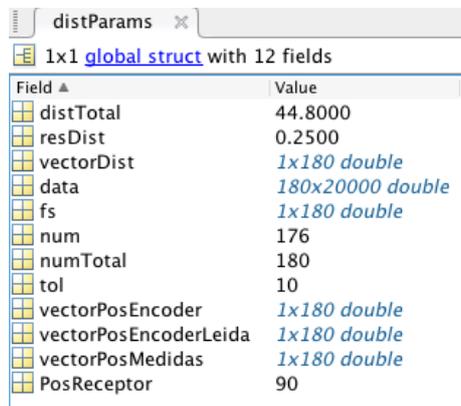


Fig.80 : Contenido de distParams

Como se puede ver en *data*, tenemos una matriz de 180×20000, que son las 180 medidas que ha realizado por 20000 que son el número de puntos adquiridos. Se puede observar que también guardamos la posición del receptor y lo que es muy útil el *vectorPosEncoderLeida* el cual guarda la posición justa en la que ha medido, a partir de esto podremos saber a que grados corresponde cada rayo.

Para poder obtener o reconstruir la imagen tomográfica es necesario procesar toda la información a la vez, para ello se creó un programa similar al que se realizó en el apartado de medidas preliminares para juntar toda la información necesaria de cada fichero en un fichero único en el que podamos acceder a todas las medidas:

```

for ii=1:length(ficheros)

    ficheros(ii).name;
    load(ficheros(ii).name);

    datos = [];
    rotacion = [];
    Receptor = [];

    datos = distParams.data;
    datos_fin = [datos_fin ;datos];

    rotacion = distParams.vectorPosEncoderLeida;
    posicionEncoder_final = [posicionEncoder_final; rotacion];

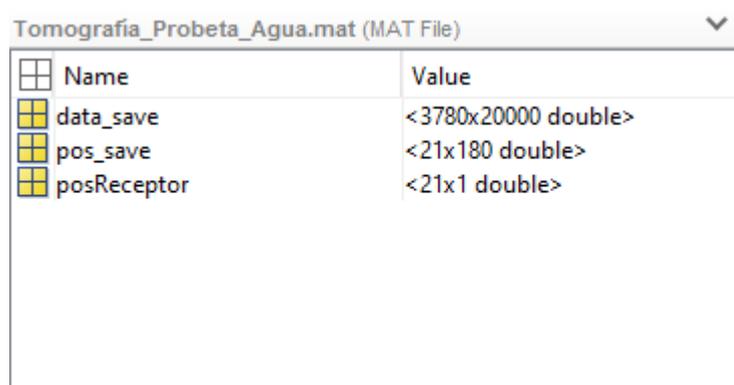
    Receptor = distParams.PosReceptor;
    Receptor_fin = [Receptor_fin; Receptor];

end

```

Fig.81 : Programa para juntar los ficheros.

Como se puede ver, de cada fichero obtenemos los datos que corresponden a la señal, el vector de posición leída con el que podremos obtener la rotación correspondiente a cada rayo y la posición del receptor. Al ejecutar este código se nos genera un archivo con la siguiente estructura:



Name	Value
data_save	<3780x20000 double>
pos_save	<21x180 double>
posReceptor	<21x1 double>

Fig.82 : Fichero final con el que se realizará la tomografía.

El resultado era de esperar, ya que en *data_save* tenemos una matriz de 3780x20000 que corresponden a las 180 medidas que se realizan para cada posición por las 21 posiciones que tenemos. En la matriz *pos_save* tenemos la posición que se ha leído para cada medida de las 180 para todas las posiciones de medida.

4.5.2 Señal utilizada

La señal utilizada es la misma que en la primera toma, lo que ahora se ha utilizado el generador de señal del osciloscopio HandyScope, el cual tiene su propio generador de señal y desde un componente creado en Matlab podemos mandarle la señal que queramos. La señal tipo Chirp la generamos desde una función propia Matlab y la cargamos al componente del HandyScope.

La señal Chirp, como se ha comentado en el apartado 2.1.2, es una señal que incrementa o decrementa la frecuencia con el paso del tiempo. En algunos casos, esta señal recibe el nombre de señal de barrido. Se utiliza para sonar y radar normalmente, pero también en técnica de espectro ensanchado, muy utilizada en comunicaciones.

4.5.3 Equipo al completo

Aquí se puede observar cómo quedaría todo el equipo para la realización de la toma de medidas:

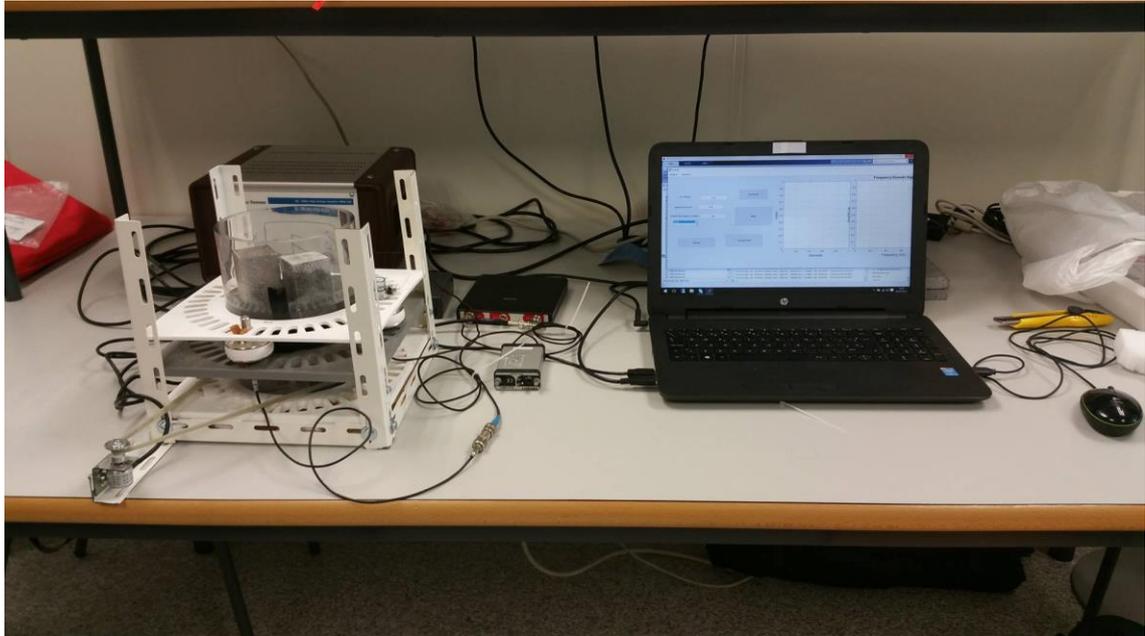


Fig. 83: Todo el equipo utilizado.

En este caso, se ha utilizado prácticamente el mismo equipo que en el apartado de medidas preliminares, cambiando solo el osciloscopio, que hemos utilizado el HandyScope como se ha citado en el apartado 3.3, y el amplificador de recepción, lo cual no cambia nada, simplemente que amplifica un poco menos.

4.5.4 Simulación previa

Al principio de este capítulo se ha explicado que en este caso, al girar la probeta, cambia un poco el concepto para la toma de medidas. Para ver que información íbamos a poder adquirir en este caso se realizó una pequeña simulación para ver cuántas medidas se obtendrían y cuántas proyecciones generaríamos.

Para ello generamos las posiciones del transmisor y del receptor tal y como mediríamos nosotros, le aplicamos la rotación que tendría para cada posición y las vamos almacenando en un vector:

```

% Simulación proyecciones
anguloTx = 0; % Posición transmisor
anguloRx = 90:9:270; % Posición receptor

rotacion = 0:2:359; % Rotación transductor . Equivale a rotar la probeta

radio = 0.152/2; %radio de la probeta 15 cm en nuestro caso

vectorAnguloTx = [];
vectorAnguloRx = [];

for iiTx = 1:length(anguloTx) % Cogemos el angulo del Tx

    anguloTxSelec = anguloTx(iiTx);

    for iiRx = 1:length(anguloRx) % Cogemos el angulo del Rx

        anguloRxSelec = anguloRx(iiRx);

        % Generamos las posiciones de los transductores
        vectorAnguloTx = [vectorAnguloTx anguloTxSelec+rotacion];
        vectorAnguloRx = [vectorAnguloRx anguloRxSelec+rotacion];

    end

end

```

Fig. 84: Simulación de las proyecciones. Generación de las posiciones de transmisor y receptor

Y obtenemos esto como resultado:

Name	Value	Min	Max
anguloRx	<1x21 double>	90	270
anguloRxSelec	270	270	270
anguloTx	0	0	0
anguloTxSelec	0	0	0
iiRx	21	21	21
iiTx	1	1	1
radio	0.0760	0.0760	0.0760
rotacion	<1x180 double>	0	358
vectorAnguloRx	<1x3780 double>	90	628
vectorAnguloTx	<1x3780 double>	0	358

Fig.85 : Resultado de la primera parte de simulación.

Obtenemos el resultado esperado, ya que tenemos 3780 ángulos por los que pasarán el transmisor y el receptor, es decir, las 3780 medidas comentadas anteriormente.

Con todos los ángulos generados, convertimos las coordenadas a cartesianas para obtener las posiciones en el eje x y en el eje y para las posiciones de transmisor y receptor y así poder simular todos los rayos:

```

%Posibles posiciones de TX y RX: pasamos de polares a cartesianas
[posTxX posTxY ] = pol2cart(vectorAnguloTx*2*pi/360,radio);
[posRxX posRxY ] = pol2cart(vectorAnguloRx*2*pi/360,radio);

% Calculamos el vector de orientación
vectorOrientacionX = posRxX-posTxX;
vectorOrientacionY = posRxY-posTxY;

figure(1), clf
% aux = plot([posTxX(1) posRxX(1)],[posTxY(1) posRxY(1)])
% axis([-0.2 0.2 -0.2 0.2])
axis('equal')

% title(num2str(length(posTxX)));

for ii=1:length(posTxX)

    figure(1)
    %Representamos los rayos
    plot([posTxX(ii) posRxX(ii)],[posTxY(ii) posRxY(ii)])
    hold on
    %Para que nos quedemos siempre en el mismo rango de visualización
    axis([-0.2 0.2 -0.2 0.2])

    title([num2str(ii) ' de ' num2str(length(posTxX)) ' rayos']);
    drawnow

end

```

Fig.86: Cálculo de las coordenadas x e y para todas las posiciones de transmisor y receptor.

A partir de las coordenadas x e y de las posiciones podemos calcular el vector de orientación, con el que podremos obtener el ángulo. Con esta parte de código dibujamos todos los rayos posibles:

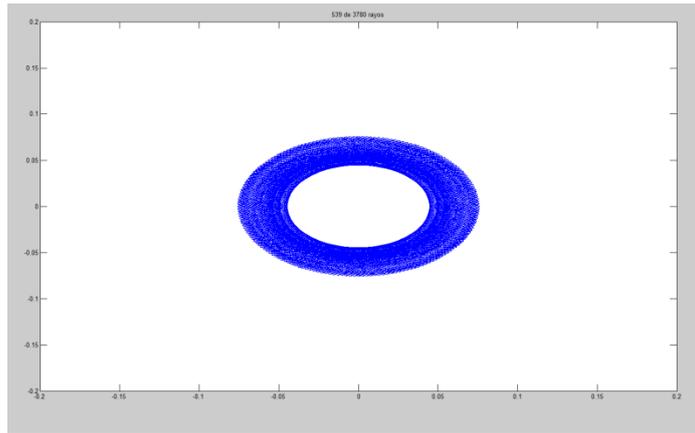


Fig.87 : Simulación de los rayos.

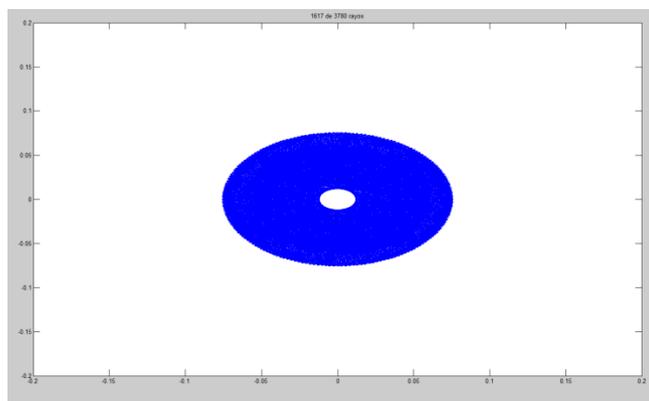


Fig.88 : Simulación de los rayos.

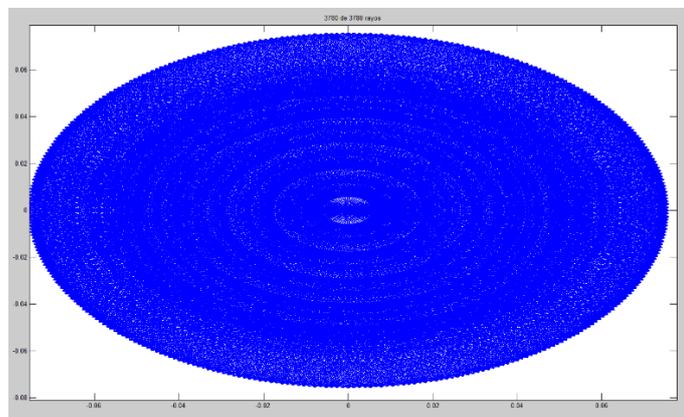


Fig.89 : Final de simulación.

Como se puede ver, cuando tenemos los 3780 rayos o medidas, quedan huecos libres, es decir, que con todas estas medidas no estaríamos abarcando toda la probeta. Si ahora pensamos en las medidas preliminares realizadas nos damos cuenta de que trazando 312 rayos en una probeta perdemos una cantidad de información muy grande, la cual es necesaria para poder reconstruir una imagen tomográfica fiable del objeto bajo estudio.

Más tarde calculamos el ángulo para cada vector de orientación y realizamos el histograma para todos los ángulos y la rotación, para así saber cuántos rayos obtendríamos por cada ángulo de rotación:

```

figure(2), clf
umbral = 1e-15;
vectorOrientacionX = vectorOrientacionX.*(abs(vectorOrientacionX)>umbral);
vectorOrientacionY = vectorOrientacionY.*(abs(vectorOrientacionY)>umbral);

angulo = angle(vectorOrientacionX+j*vectorOrientacionY)*180/pi;
%angle devuelve los ángulos de los complejos entre [-pi,pi]. Para pasar el
%rango de [0,360°] tenemos que sumarle 360 en el caso en que el ángulo sea
%negativo.
for i=1:length(angulo)
    if angulo(i)<0
        angulo(i)=angulo(i)+360;
    end
end

hist(angulo,rotacion)

```

Fig.90 : Cálculo de los ángulos y dibujar su histograma.

Y obtenemos:

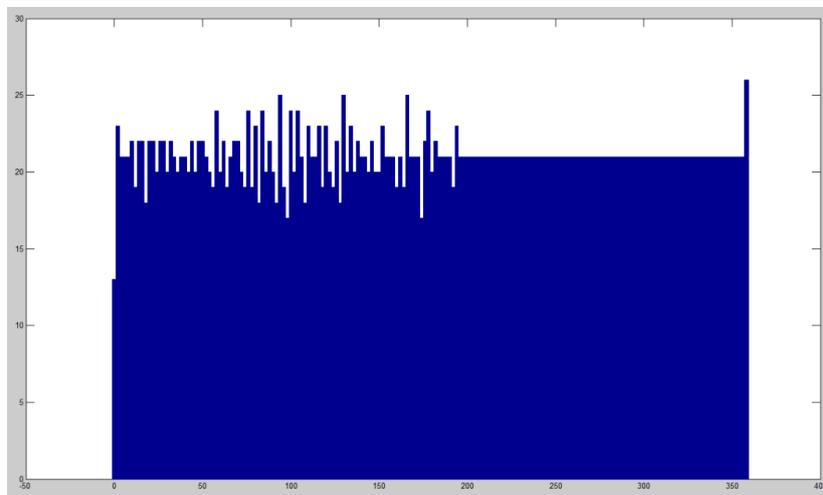


Fig.91 : Histograma.

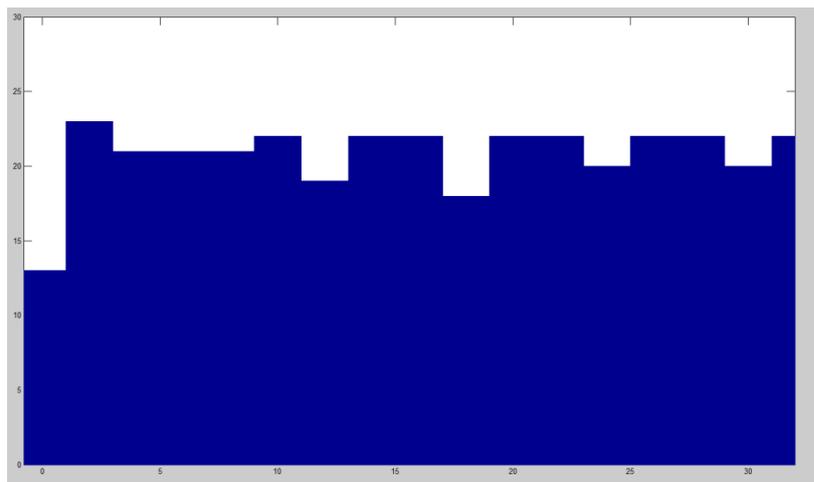


Fig.92 : Histograma ampliado.

Aquí se ve que para el ángulo 5° obtendríamos 21 rayos, para 10° 22 rayos, para 15° 22 también, etc.

Más tarde, hemos realizado un apartado para saber cuántos rayos tiene cada proyección, buscando a partir de un índice que compara el ángulo buscado con todos los ángulos con una tolerancia:

```
%% Ejecutamos representación

anguloBuscado =0;

indice = find(abs(angulo-anguloBuscado)<=1);
figure(3),clf

for ii=indice, plot([posTxX(ii) posRxX(ii)],[posTxY(ii) posRxY(ii)]), hold on, end,
axis([-20 20 -20 20]), axis('equal')

title(['Proyección: ' num2str(anguloBuscado) ' °'])
```

Fig.96 : Representación de todos los rayos pertenecientes a la proyección del ángulo buscado.

Por ejemplo para 0°:

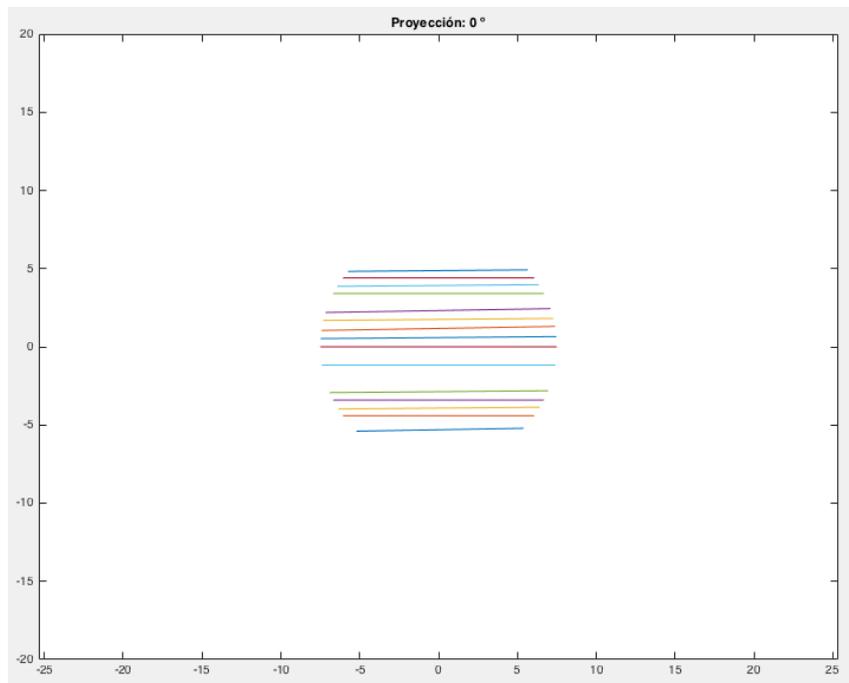


Fig. 97: Proyección a 0°.

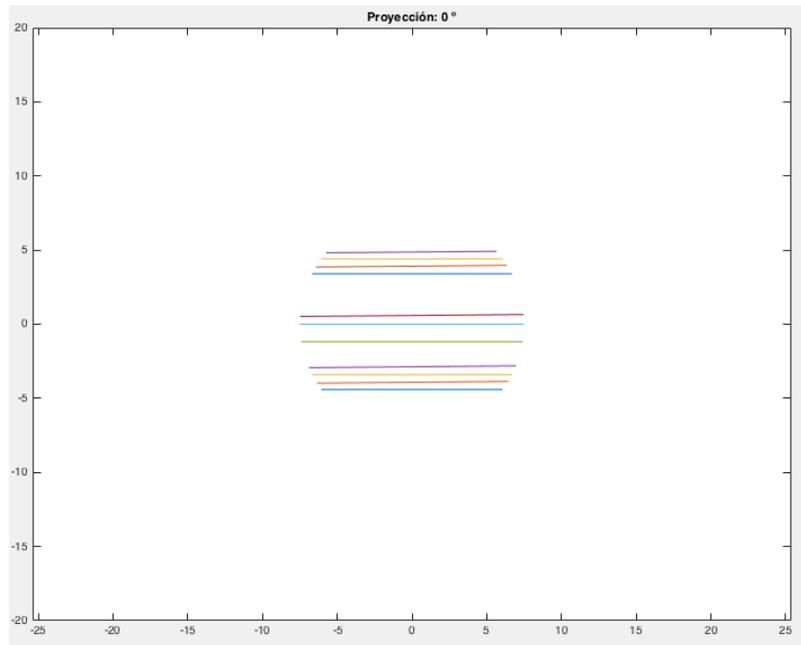


Fig.98 : Proyección a 0° con una tolerancia de 0.5°.

Vemos como si bajamos la tolerancia se pierden rayos, lo cual es normal, para hacer la reconstrucción, hemos dejado una tolerancia de 1 grado, ya que es mejor que coja algún rayo de otra proyección a que pierda rayos de la propia.

Por tanto, el siguiente paso es calcular los parámetros a partir de las señales que tenemos para realizar la reconstrucción. En este caso, se han utilizado los mismos parámetros que se utilizaron para realizar la tomografía en el apartado de medidas preliminares.

Estos son los parámetros utilizados:

```
%% Paso de los datos a iradon

angulo_ordenado=unique(angulo);
aux=data_save;
amplitud=max(abs(aux)');
potencia = sum((aux').^2);
atenuacion = -log10(potencia)./distanciaTxRx.^2;
[auxV, auxT ] = US_speed(5 , 1500 , 0.1 , 50e6 , data_save);
velocidades = distanciaTxRx*1e-2./auxT;% en m/s
```

Fig.99 : Parámetros utilizados para la reconstrucción.

Igual que en el ejemplo que hemos realizado, vamos buscando en cada uno de estos parámetros y seleccionamos los rayos pertenecientes a cada ángulo:

```
for i=1:length(anguloProyecciones)

    indice=find(abs(angulo-anguloProyecciones(i))<=0.5);

    %Amplitudes de la proyección seleccionada.
    amplitudSeleccionada = amplitud(indice);
    %Potencias de la proyección seleccionada.
    potenciaSeleccionada = potencia(indice);
    %Velocidades de la proyección seleccionada.
    velocidadesSeleccionada = velocidades(indice);
    %Atenuaciones de la proyección seleccionada.
    atenuacionSeleccionada = atenuacion(indice);
```

Fig. 100: Rayos seleccionados para cada proyección.

E interpolamos y almacenamos para representar:

```
% Interpolamos
amplitudInterpolada = interp1(xPR, amplitudSeleccionada, xPRinterpolada);
potenciaInterpolada = interp1(xPR, potenciaSeleccionada, xPRinterpolada);
velocidadInterpolada = interp1(xPR, velocidadesSeleccionada, xPRinterpolada);
atenuacionInterpolada = interp1(xPR, atenuacionSeleccionada, xPRinterpolada);
% Guardamos los parámetros
MatrizAmplitud(i,:) = amplitudInterpolada;
MatrizPotencia(i,:) = potenciaInterpolada;
MatrizVelocidad(i,:) = velocidadInterpolada;
MatrizAtenuacion(i,:) = atenuacionInterpolada;
```

Fig.101 : Interpolación y almacenamiento de los datos.

Por tanto, solo nos queda representar. Antes de representar vimos que al hacer la interpolación, había muchos valores que se convertían en NaN y en la representación salía así:

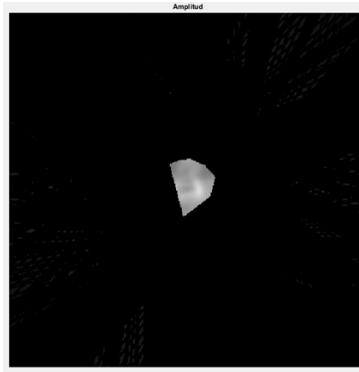


Fig.102 : Tomografía representando los valores NaN.

En la que se puede observar que no se aprecia nada, quitando dichos valores, los resultados son los siguientes:

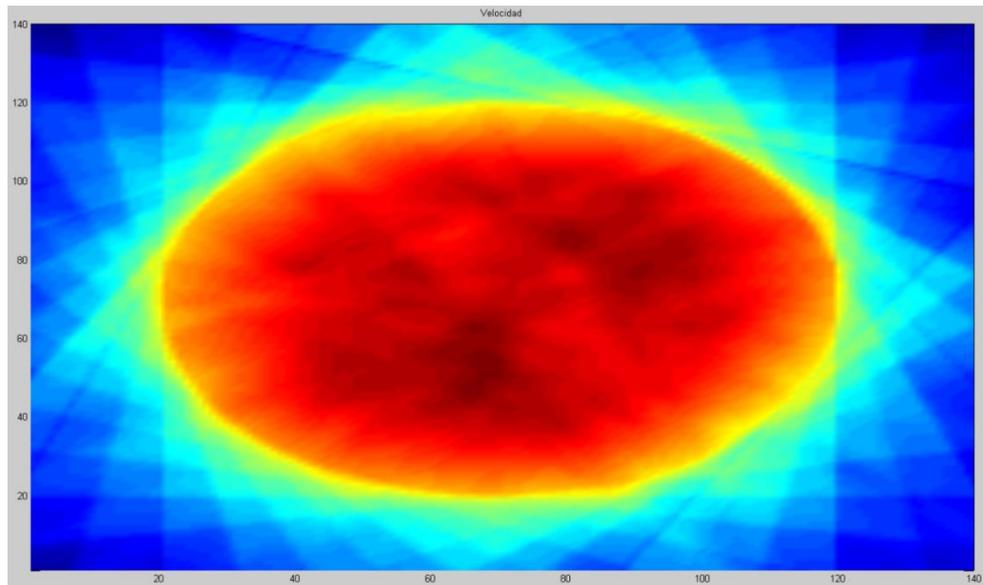


Fig. 103: Tomografía final.

El parámetro utilizado para el ensayo final ha sido la velocidad, ya que los otros parámetros por razón de transductores, que estén bien fijados o que estén sometidos a la fuerza necesaria para que la señal sea buena, no nos parecían tan fiables ya que la amplitud y la potencia variaban de una forma significativa. Hemos utilizado la velocidad, porque la velocidad de una

señal ultrasónica en cada material no varía. A parte, esta compensada por la distancia que hay entre la posición del transmisor y del receptor.

Después de haber obtenido la imagen, podemos asegurar que el borde de cristal de la probeta ha podido interferir mucho en la señal, ya que tiene un grosor de aproximadamente 3 mm conformando una lente que genera efectos de reflexión, difracción y atenuación.

De todos modos, la imagen obtenida se aprecian las dos probetas introducidas en la inspección, la de aluminio y la de cemento:

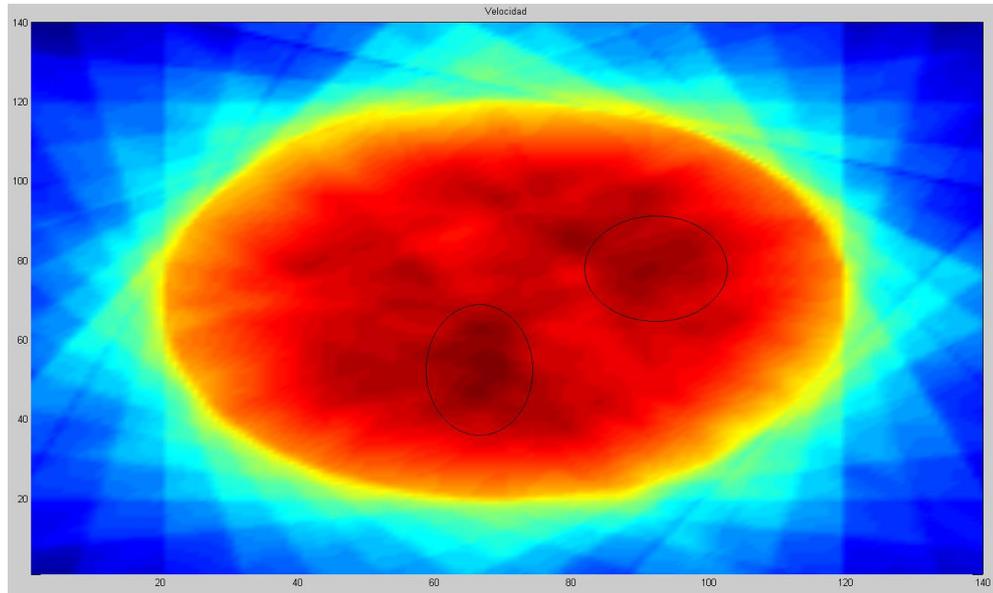


Fig. 104: Tomografía.

Las dos zonas marcadas con una velocidad mayor, se pueden asociar a las probetas introducidas. Dichas zonas no salen cuadradas tal y como se esperaba, pero se puede observar que hay un cambio de velocidades en dichas zonas que sería compatible con los materiales introducidos.

Capítulo 5. Conclusiones y líneas futuras

En este trabajo se han trabajado y alcanzado los siguientes apartados:

- Se ha estudiado los fenómenos que se producen en la propagación ultrasónicas así como los conceptos fundamentales de tomografía ultrasónica: trazado de rayos, algoritmos de retroproyección, toma de medidas...
- Se ha implementado un sistema hardware para la toma masiva de medidas y su almacenamiento para su posterior procesamiento. Se ha implementado un sistema de posicionamiento mediante encoder circular y Arduino, todo ello conectado mediante Matlab al osciloscopio digital Handyscope.
- Dentro de la parte de software se ha desarrollado diferentes programas donde cabe destacar:
 - Software de adquisición, encargado de comunicarse con Arduino y con el equipo de adquisición para tomar las medidas para las diferentes posiciones espaciales
 - Software de preparación de datos, este software ha sido el encargado de juntar los datos y montar las proyecciones para la reconstrucción tomográfica
 - Software de extracción de parámetros ultrasónicos y reconstrucción tomográfica. Tal y como se ha explicado en el Capítulo 2 y 3, por cada medida, había que obtener un parámetro ultrasónico, que tuviera en cuenta la propagación por el material así como la distancia recorrida y después realizar la reconstrucción interna del material.
 - Software de simulación, también se ha realizado un software de simulación para valorar el número de proyecciones y el área inspeccionada
- En cuanto a las medidas realizadas, se hizo una primera puesta a punto de los sistemas con una tomografía realizada de forma manual, dicha tomografía tenía unas XX proyecciones con 5 puntos útiles por proyección. Se comprobó que la tomografía obtenida carecía de la calidad necesaria. Después de implementar el sistema de toma masiva, se paso a XX proyecciones con XX puntos útiles.

Los resultados finales fueron mejores con el sistema de toma masiva que con el sistema manual, pero no todo lo satisfactorio que se esperaban. Esto se presupone que es debido a varios fenómenos:

- Alta atenuación de los materiales, sobre todo en el caso del hormigón que deja señales con una mala relación señal a ruido

- Múltiples materiales intermedios que generan efectos de difracción y reflexión que desenfoca la imagen: sobre todo en la probeta de cristal con aluminio y mortero
- La propagación de los ultrasonidos no es en línea recta como ocurre con los Rayos X debido a la propia naturaleza de la onda.
- Existe un compromiso entre longitud de onda (resolución) y capacidad de penetración.

Este proyecto, deja muchas líneas de trabajo donde destacamos los siguientes aspectos:

- Tomografía por inmersión, lo que mejoraría la automatización del acoplamiento entre el sensor y la probeta
- Algoritmos de reconstrucción basados en redes neuronales, con el fin de compensar la difracción de los rayos X
- Desarrollo de algoritmos de reconstrucción con un conocimiento a priori de la heterogeneidad a buscar.

Bibliografía

- [1] Candelas, P. (2002). *Contribución al ensayo no destructivo de materiales mediante tomografía ultrasónica*. Valencia. Universidad Politécnica.
- [2] Juzga, A. y Rodriguez, B. *Método de retroproyección filtrada*. Consultado el , en <http://matematicas.uis.edu.co> .
- [3] Moreno, J. M. , Topete, J. , Fritsch, C. , González-Bueno, R. y Giacchetta, R. (2015) *Técnicas de inspección por ultrasonido en aire*. 13º Congreso nacional de ensayos no destructivos. Sevilla.
- [4] Naragon, R. (2013-2014). *Matemáticas de los sistemas de imágenes biomédicas(II)*. <http://slideplayer.es/slide/3480183/> .
- [5] Alvarez, A. , Arámbula, E. y Caro, S.(2008). Tomografía computarizada con rayos-x y sistema imágenes de agregados (AIMS) y para el estudio de mezclas asfálticas y agregados. *Revista de ingeniería e investigación*, 28 (2).
- [6] Natterer, F. y Wübbeling, F.(1995). A propagation-backpropagation method for ultrasound tomography. *Inverse problems*, 11, 1225-1232.
- [7] Daigle, M. , Fratta, D. y Wang, L.B. (2005). Ultrasonic and X-ray Tomographic Imaging of Highly Contrasting Inclusions in Concrete Specimens. *Geofrontier 2005 Conference*.Austin, Tejas.