

IMÁGENES VOLUMÉTRICAS POR ULTRASONIDOS A TRAVÉS DE INTERFACES MEDIANTE EL MÉTODO DEL ARRAY VIRTUAL

Cosarinsky, Guillermo^{1,2,3} (g.cosarinsky@csic.es); F. Cruza, Jorge¹; Muñoz, Mario¹; Camacho, Jorge¹

¹ Grupo de Sistemas y Tecnologías Ultrasónicas (GSTU), Instituto de Tecnologías Físicas y de la Información Leonardo Torres Quevedo (ITEFI). Consejo Superior de Investigaciones Científicas (CSIC), c/ Serrano 144, 28006, Madrid, España

² Departamento de Ensayos No Destructivos, Comisión Nacional de Energía Atómica, Av. Gral. Paz 1499, B1650 Villa Maipú, Provincia de Buenos Aires, Argentina

³ Departamento de Electrónica, Escuela Politécnica, Universidad de Alcalá de Henares, Ctra. Madrid-Barcelona, Km. 33,600, 28805 Madrid, España

ABSTRACT

Ultrasound images are widely used both in medicine and Non Destructive Testing (NDT). Image formation requires calculating times of flight (TOF) of waves through the inspected material. Biological tissues inspected in medicine have an almost constant propagation speed (that of water), thus only distance computations are needed to calculate TOF.

Nevertheless, in NDT it is commonplace to use a coupling medium whose propagation speed is very different to that of the inspected component. Refraction of ultrasound at the interface between the two mediums complicates calculating TOF. In this case an iterative computationally intensive method is needed, usually based in Fermat's principle.

In this work we propose a method to optimize TOF calculations for volumetric imaging (3D) with matrix arrays in the case of two propagation mediums. The method is based on the definition of a virtual array in the second propagation medium such that TOF from its elements approximate precisely the exact ones, but operating without refraction. The mathematical development of the method and its experimental validation with reference and real components is presented.

RESUMEN

Las imágenes por ultrasonidos se utilizan extensamente tanto en medicina como en Ensayos No Destructivos (END).

La conformación de imágenes requiere de cálculos de los tiempos de vuelo de las ondas a través del material inspeccionado. En medicina se inspeccionan tejidos biológicos cuya velocidad de propagación es aproximadamente constante (la del agua), por lo cual el cálculo de tiempos de vuelo implica solamente el cálculo de distancias en línea recta.

En cambio, en END es frecuente el uso de un medio de acoplamiento con una velocidad de propagación muy diferente a la del material ensayado. La refracción de los ultrasonidos al atravesar la interfaz entre los dos medios complica considerablemente el cálculo de los tiempos. En este caso es necesario un cálculo iterativo y computacionalmente costoso, basado generalmente en el principio de Fermat.

En este trabajo se propone una optimización del cálculo de los tiempos de vuelo en presencia de dos medios para arrays matriciales e imágenes volumétricas (3D). Se basa en definir un array virtual en el segundo medio, cuyos tiempos de vuelo sean equivalentes a los reales, pero operando sin refracción. Se presenta el desarrollo matemático del array virtual y su validación mediante experimentos con probetas de calibración y piezas reales.

Palabras Clave: imágenes por ultrasonidos, arrays, Total Focusing Method, GPU, tiempo de vuelo, imagen 3D, Ensayos No Destructivos

1. INTRODUCCIÓN

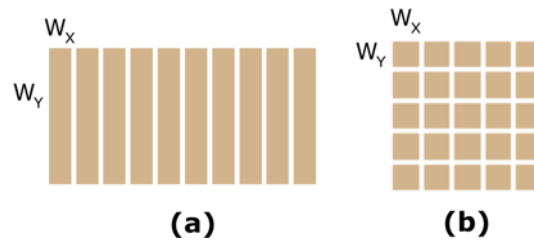


Figura 1 – Elementos de un array de transductores. a) array lineal, b) array matricial

El uso de ultrasonidos para Ensayos No Destructivos ha avanzado notablemente mediante el uso de transductores de tipo array [1]. Estos transductores están formados por un conjunto de pequeños elementos que sirven tanto para transmitir como para recibir el ultrasonido [2]. La elección del instante en que excita de cada elemento permite la generación de distintos tipos de campos ultrasónicos: haces colimados, ondas planas y ondas omnidireccionales son las formas más utilizadas. A su vez, las señales recibidas por los elementos se pueden combinar mediante distintos algoritmos para la obtención de imágenes, siendo el método de Retardo y Suma el más utilizado, conocido comúnmente por su nombre en inglés: Delay And Sum (DAS). El proceso de combinación de las señales se conoce como conformación de haces (beamforming) [3].

En NDT el tipo de array utilizado más frecuentemente es el lineal, compuesto por una hilera de elementos rectangulares de dimensiones (W_x , W_y), tales que $W_x \ll W_y$ (figura 1). Esto permite generar un haz colimado en la dirección Y (dirección pasiva o de elevación). En cambio, en dirección X (activa o azimutal), cada elemento tiene un ancho similar o menor que la longitud de onda, y la emisión es aproximadamente omnidireccional, de modo que la activación de todos los elementos con un conjunto de retardos apropiado (ley focal), permite deflectar y/o enfocar el haz en un amplio rango angular dentro del plano XY, permitiendo obtener imágenes planas (2D).

Existen también los arrays matriciales [4], formados por una rejilla de elementos cuadrados o rectangulares pero de lados similares ($W_x \sim W_y$). Este tipo de arrays permite deflectar el haz en todas las direcciones espaciales, superando la limitación de las imágenes bidimensionales de los arrays lineales. Sin embargo, este tipo de arrays no se utilizan usualmente en la industria. Una de las razones, es que, para obtener buena resolución lateral en ambas direcciones, es necesaria una apertura grande, y por lo tanto un número elevado de elementos.

Al poder deflectar el haz ultrasónico en todas las direcciones espaciales, los arrays matriciales permiten obtener imágenes volumétricas. Esto añade una dificultad a la hora de calcular leyes focales cuando hay un medio de acoplamiento entre el array y el componente inspeccionado, dado que para calcular los tiempos de vuelo desde los elementos del array a los puntos focales en el interior del componente, es necesario tener en cuenta la refracción en la interfaz entre los dos medios de propagación. Esta dificultad se presenta también en el caso de arrays lineales e imágenes 2D, pero con un grado de libertad menos.

Para el cálculo del tiempo de vuelo entre dos puntos, se debe hallar el punto de entrada del rayo en la interfaz que satisfaga el principio de Fermat [5]. Esto requiere en general de una búsqueda iterativa, dado que para una interfaz de forma arbitraria, salvo en algún caso específico [6], no hay una solución mediante una fórmula cerrada. Si se desea generar una imagen compuesta por L líneas y K focos por línea utilizando un array de N elementos, se deben calcular $N \times L \times K$ tiempos de vuelo (del orden de 10^6 búsquedas iterativas para 128 elementos). Esto es asequible para una única imagen estática, pero no si se desea obtener imagen en tiempo real, a unas 30 imágenes por segundo.

Para simplificar el cálculo de tiempos de vuelo en presencia de interfaces se ha desarrollado el método del array virtual [5]. Este método permite reducir el problema al caso de un solo medio de propagación, mediante la definición de un array virtual en el segundo medio, cuyos tiempos de vuelo sean equivalentes a los reales, pero operando sin refracción. En [5], el método fue

desarrollado para arrays lineales e imagen 2D. Los arrays lineales presentan limitaciones para la inspección de componentes de formas complejas, por lo cual el uso de arrays matriciales en NDT puede habilitar nuevas soluciones a problemas de inspección difíciles. Con este objetivo en mente, en este trabajo se generaliza el array virtual para el caso 3D con arrays matriciales e imágenes volumétricas, y se muestran ejemplos con probetas de calibración y piezas reales.

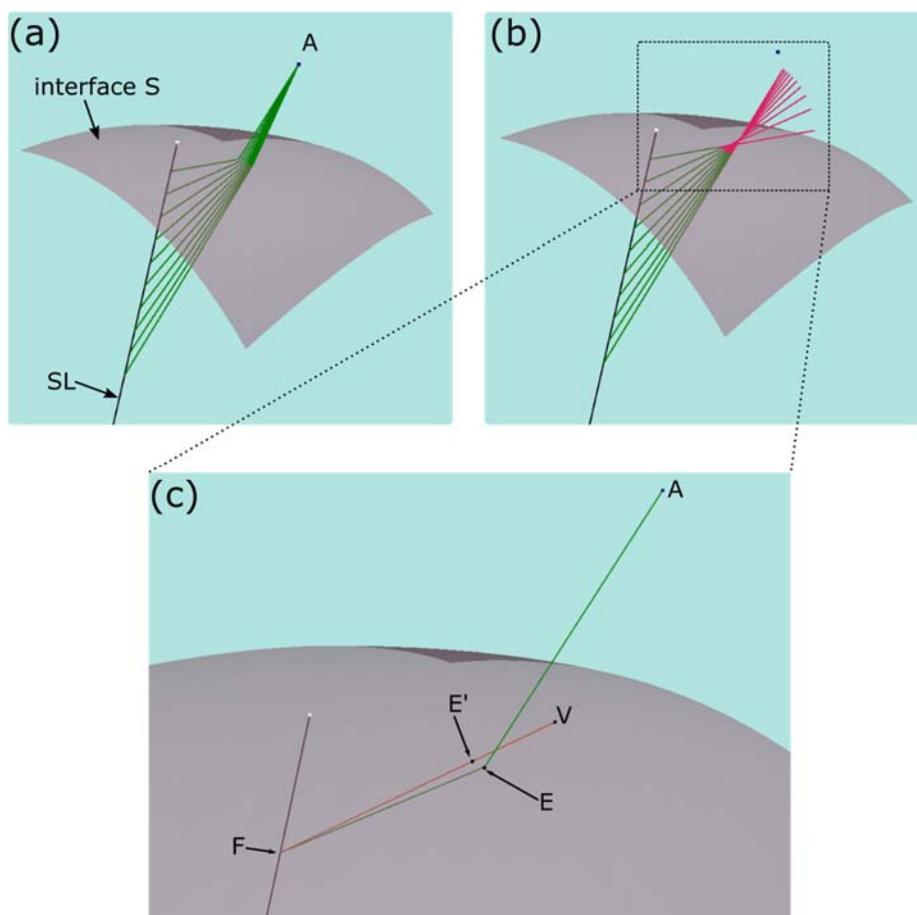


Figura 2 – Esquema de rayos emitidos desde una fuente A que se refractan en la interfaz entre dos medios de propagación con velocidades diferentes. a) Varios rayos desde A hacia puntos sobre una línea SL en el segundo medio. b) Rayos refractados prolongados hacia el primer medio. c) Detalle de un rayo y el correspondiente rayo virtual desde el punto V

2. FORMULACIÓN MATEMÁTICA DEL ARRAY VIRTUAL EN 3D

En la figura 2.a se esquematiza la propagación de rayos de ultrasonido desde una fuente A en el medio 1, con velocidad de propagación c_1 , hacia puntos focales sobre una línea SL de la imagen que se desea conformar. Los puntos focales se hallan en el interior de la pieza, que es el medio 2 con velocidad de propagación c_2 . En la figura 2.b se observan los rayos refractados en la interfaz S , prolongados hacia el medio 1. Estos rayos no se intersectan en un punto como sucede en la óptica paraxial [7], pero existe una pequeña zona que todos atraviesan, en la cual se puede pensar en ubicar un punto V que actúe como fuente virtual, de modo de poder calcular el tiempo de vuelo desde A hasta un foco F sobre SL , como si el ultrasonido se propagase desde V hasta F únicamente a través del medio 2 y por tanto, sin refracción. El tiempo de vuelo exacto se calcula mediante el principio de Fermat:

$$t(A, F) = \min \left\{ \frac{d(A, E)}{c_1} + \frac{d(E, F)}{c_2} \right\} \quad (1)$$

donde E es el punto de entrada del rayo en la superficie S del componente.

La aproximación que se propone es buscar un punto V y una constante t_K tal que:

$$t(A, F) = \frac{d(V, F)}{c_2} + t_K \quad (2)$$

Para ello, reformulamos (2) del siguiente modo:

$$\begin{aligned} t_K &= t(A, F) - \frac{d(V, F)}{c_2} = \frac{d(A, E)}{c_1} + \frac{d(E, F)}{c_2} - \frac{d(V, F)}{c_2} \\ &= \frac{d(A, E)}{c_1} + \frac{d(E, F)}{c_2} - \left(\frac{d(V, E')}{c_2} + \frac{d(E', F)}{c_2} \right) \end{aligned} \quad (3)$$

donde E' es la intersección de la línea \overline{VF} , que podríamos llamar "rayo virtual" (figura 2.c). El t_K exacto depende del foco F . El objetivo es aproximarlos por una constante para toda la línea SL , independiente de la posición de F . Para ello vamos a suponer que podemos realizar la aproximación $E \cong E'$, mediante la cual obtenemos:

$$t_K = \frac{d(A, E)}{c_1} + \frac{d(E, F)}{c_2} - \left(\frac{d(V, E)}{c_2} + \frac{d(E, F)}{c_2} \right) = \frac{d(A, E)}{c_1} - \frac{d(V, E)}{c_2} \quad (4)$$

Se puede realizar una simplificación adicional de la ecuación (4) utilizando el principio de Abbe [7] de la óptica geométrica. Según este principio, si el radio de curvatura de la superficie es grande en comparación con las distancias involucradas, resulta la siguiente relación:

$$d(A, E) = \frac{c_1}{c_2} d(V, E) \quad (5)$$

Utilizando (5) en la ecuación (4) se obtiene:

$$t_K = \frac{d(A, E)}{c_1} \left(1 - \left(\frac{c_1}{c_2} \right)^2 \right) \quad (6)$$

El siguiente paso es encontrar el punto V . Para ello se eligen dos focos F_1 y F_2 sobre la línea SL , y se calcula para ellos los puntos de entrada E_1 y E_2 resolviendo (1) mediante un método iterativo. Para esto es necesario conocer una descripción de la interfaz S como una función $z = f(x, y)$ o como una representación discretizada por puntos. Una vez conocidos los tiempos de vuelo exactos t_1 y t_2 desde A hasta F_1 y F_2 , se impone la condición de que esos tiempos calculados mediante la fuente virtual sean iguales a los exactos para esos dos puntos:

$$\begin{cases} t_1 - t_K = \frac{d(V, F_1)}{c_2} \\ t_2 - t_K = \frac{d(V, F_2)}{c_2} \end{cases} \quad (7)$$

El valor de t_K según (6) depende del punto de entrada E , y por lo tanto de F . Para que sea una constante en toda la línea SL , utilizamos el valor que corresponde a E_1 ó a E_2 indistintamente. De este modo, podemos interpretar las ecuaciones (7) como dos esferas de radios $R_1 = c_2(t_1 - t_K)$ y $R_2 = c_2(t_2 - t_K)$. Estas esferas tienen sus centros sobre SL , y su intersección es una circunferencia OV de radio h_V cuyo centro se encuentra en SL en la posición r_V medida con respecto al punto de entrada de E_{SL} de SL (figura 3). Cualquier punto sobre OV puede ser utilizado como fuente virtual V . Los parámetros (r_V, h_V) se calculan mediante las siguientes fórmulas:

$$\gamma = \frac{R_1^2 - R_2^2 + (r_2 - r_1)^2}{2(r_2 - r_1)} \quad (8)$$

$$h_V = \sqrt{R_1^2 - \gamma^2} \quad (9)$$

$$r_V = r_1 + \gamma \quad (10)$$

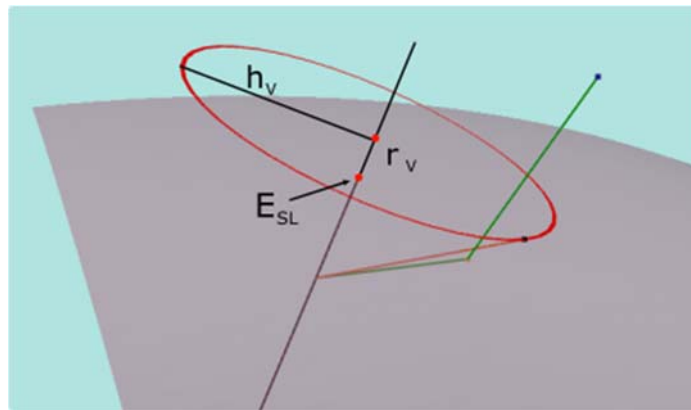


Figura 3 – Círculo OV centrado en línea SL . Cualquier punto en OV sirve como fuente virtual para calcular el tiempo de vuelo hacia focos en SL

Para obtener el array virtual, se aplica el procedimiento a cada elemento A del array. Esto debe hacerse para cada línea de la imagen, de modo que para cada una de ellas se deben realizar sólo dos búsquedas iterativas (ecuación (7)), en lugar de una por cada foco. Se consigue entonces pasar de $N \times L \times K$ a $N \times L \times 2$ búsquedas iterativas, normalmente, entre 2 y 3 órdenes de magnitud menos operaciones.

3. VALIDACIÓN EXPERIMENTAL

Se realizaron ensayos en inmersión, utilizando un array de 11×11 elementos, 3 MHz, con un pitch de 1mm. El equipo de ultrasonido utilizado fue el SITAU de DASEL. El tipo de adquisición fue Full Matrix Capture (FMC) [8]. Se ensayaron dos componentes:

- Un bloque de aluminio de 30 mm de espesor, con 8 agujeros de 2mm de diámetro y distintas profundidades (figura 4.a).
- Un tubo de latón, de 32 mm de radio y 9 mm de pared, con defectos internos que simulan corrosión (figura 4.b).

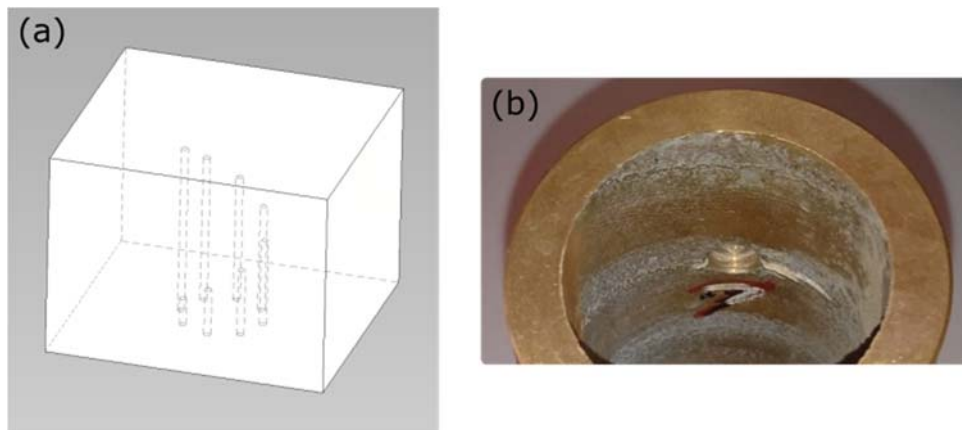


Figura 4 – Componentes utilizadas para los ensayos. a) Bloque de aluminio con agujeros de diferentes profundidades, b) Tubo de latón con defectos internos que simulan corrosión

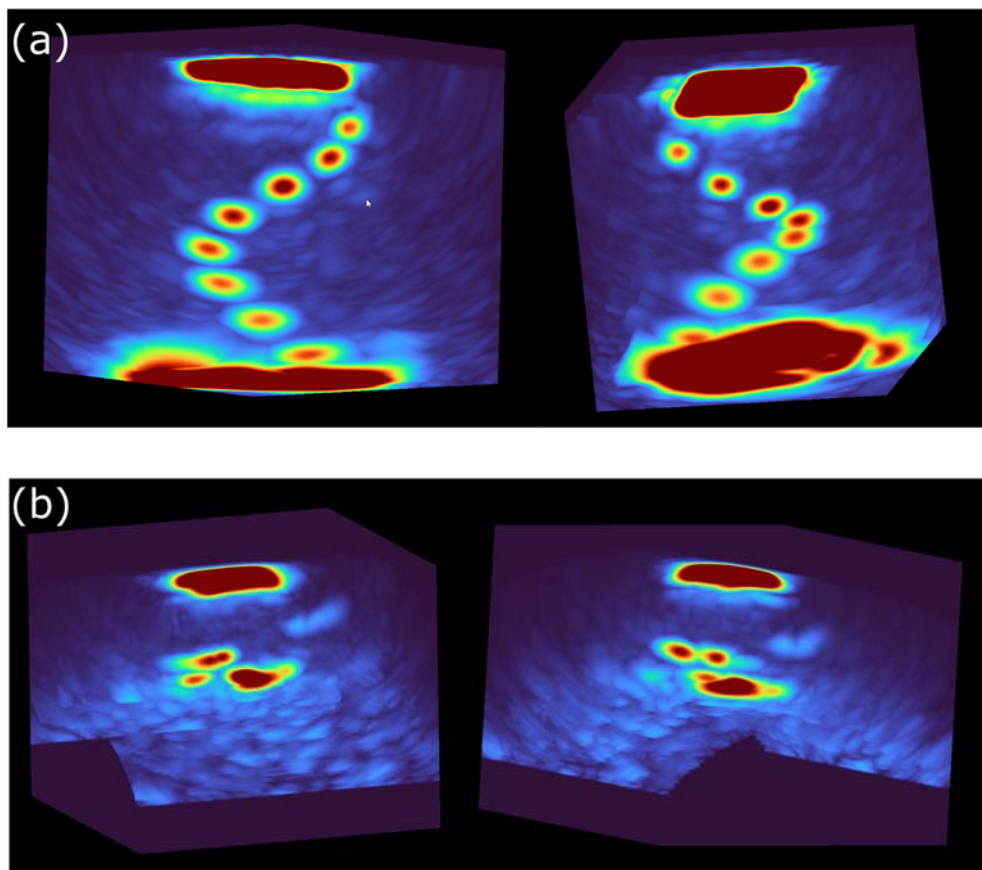


Figura 5 – 2 vistas 3D de las imágenes volumétrica TFM calculada con el array virtual. a) Pieza A, b) Pieza B

Para cada componente se conformó una imagen en una región de Interés (ROI) con el método Total Focusing Method (TFM) [8]. La ROI es un volumen en forma de prisma rectangular, discretizado en $N_x N_y N_z$ píxeles (o voxeles al tratarse de un volumen). Cada línea de píxeles a lo largo del eje Z es una de las líneas SL para las cuales se calcula un array virtual, de modo que la imagen contiene $L = N_x N_y$ líneas.

Las imágenes TFM calculadas con el método del array virtual se muestran en 2 vistas 3D en la figura 5.a para la pieza A, y en la figura 5.b para la pieza B. El tamaño de las imágenes es 60x60x140. La implementación del cálculo de TFM exacto (búsqueda iterativa para todos los focos) y del TFM con array virtual fue implementada en GPU sobre el módulo Pyopencl de Python [9]. El tiempo de ejecución del TFM exacto fue 5 s, mientras que para el TFM con array virtual fue de 35 ms. Es decir, el método del array virtual permitió un cálculo 142 veces más rápido de la imagen volumétrica TFM. EL hardware utilizado para ejecutar el código fue un ordenador con procesador AMD Ryzen 9 3900X 12-Core 3.79 GHz y una GPU NVIDIA GeForce RTX 3080.

En la figura 5.a se observa como los 8 agujeros de la pieza A se encuentran bien enfocados, lo cual indica que los tiempos de vuelo están bien calculados a todas las profundidades de la imagen. En efecto, la imagen no muestra diferencias apreciables con la imagen TFM exacta. El valor medio de la diferencia entre la imagen exacta y la calculada con el array virtual es de 0.2% del valor máximo de la imagen exacta.

La figura 5.b muestra como el método permite visualizar claramente las indicaciones que simulan corrosión en el interior de la pieza B.

4. CONCLUSIONES

Se ha presentado un método de aproximación para el cálculo de tiempos de vuelo para el caso de dos medios de propagación con velocidades diferentes y arrays matriciales. Este método, basado en la definición de fuentes virtuales que permiten reducir el problema al caso de un único medio, permite optimizar el cálculo de imágenes volumétricas con arrays matriciales. El método es especialmente útil en NDT, dado que es usual la inspección de componentes a través de un medio acoplante con velocidad de propagación muy distinta a la del material del componente.

Se ha mostrado como la aplicación del método del array virtual para el cálculo de imágenes TFM volumétricas ha permitido, en el ejemplo presentado, una reducción de 142 veces en el tiempo de procesamiento, en comparación a la implementación del cálculo exacto de tiempo de vuelo para todos los focos de la imagen. Esta optimización abre las puertas hacia la posibilidad de inspecciones 3D con arrays matriciales en tiempo real, para velocidades de barrido moderadas.

AGRADECIMIENTOS

Este trabajo fue realizado en el contexto del Proyecto RTI2018-099118-A-I00 de MCIU/AEI/FEDER, UE, y el contrato predoctoral PRE2019-088602 del Ministerio de Ciencia e Innovación de España.

REFERENCIAS

- [1] B. W. Drinkwater and P. D. Wilcox, "Ultrasonic arrays for non-destructive evaluation: A review," *NDT E Int.*, vol. 39, no. 7, pp. 525–541, 2006
- [2] J. Krautkramer, *Ultrasonic Testing of Materials*, Springer-Verlag (Alemania), 4ta edición, 1990
- [3] T. L. Szabo, *Diagnostic Ultrasound Imaging: Inside Out*, Elsevier Science & Technology (USA), 2013.
- [4] L. W. Schmerr, "Fundamentals of ultrasonic phased arrays," *Solid Mechanics and its Applications*, vol. 215, Springer, 2015.
- [5] J. Camacho, J. F. Cruza, J. Brizuela, and C. Fritsch, "Automatic dynamic depth focusing for NDT," *IEEE Trans. Ultrason. Ferroelectr. Freq. Control*, vol. 61, no. 4, pp. 673–684, 2014
- [6] M. Weston, P. Mudge, C. Davis, and A. Peyton, "Time efficient auto-focussing

- algorithms for ultrasonic inspection of dual-layered media using Full Matrix Capture,” *NDT E Int.*, vol. 47, pp. 43–50, 2012
- [7] M. Born *et al.*, *Principles of Optics*. Cambridge University Press (Gran Bretaña), 1999.
- [8] C. Holmes, B. Drinkwater, and P. Wilcox, “The post-processing of ultrasonic array data using the total focusing method,” *Insight Non-Destructive Test. Cond. Monit.*, vol. 46, no. 11, pp. 677–680, 2004
- [9] A. Klöckner, N. Pinto, Y. Lee, B. Catanzaro, P. Ivanov, and A. Fasih, “PyCUDA and PyOpenCL: A scripting-based approach to GPU run-time code generation,” *Parallel Comput.*, vol. 38, no. 3, pp. 157–174, 2012