

XII JORNADAS DE REDES DE INVESTIGACIÓN EN DOCENCIA UNIVERSITARIA

El reconocimiento docente: innovar e investigar con criterios de calidad

ISBN: 978-84-697-0709-8



Diseño: Gabinete de Imagen y Comunicación Gráfica de la Universidad de Alicante

XII JORNADES DE XARXES D'INVESTIGACIÓ EN DOCÈNCIA UNIVERSITÀRIA

El reconeixement docent: innovar i investigar amb criteris de qualitat

Coordinadores

María Teresa Tortosa Ybáñez

José Daniel Álvarez Teruel

Neus Pellín Buades

© **Del texto: los autores**

© **De esta edición:**

Universidad de Alicante

Vicerrectorado de Estudios, Formación y Calidad

Instituto de Ciencias de la Educación (ICE)

ISBN: 978-84-697-0709-8

Revisión y maquetación: Neus Pellín Buades

Laboratorio virtual basado en MATLAB para la docencia de Acústica en el Grado de Telecomunicación

J. Francés Monllor^{(1)(2)*}; S. Bleda Pérez⁽¹⁾⁽²⁾; B. Otero Calviño⁽³⁾; E. M. Calzado Estepa⁽¹⁾⁽²⁾; J. de D. González Ruiz⁽¹⁾; S. Heredia Ávalos⁽¹⁾⁽²⁾; A. Hernández Prados⁽¹⁾⁽²⁾; A. Hidalgo Otamendi⁽¹⁾; D.I. Méndez Alcaraz⁽¹⁾⁽²⁾; M. S. Yebra Calleja⁽¹⁾⁽²⁾; J. Vera Guarinos⁽¹⁾⁽²⁾;

(1) *Departamento de Física, Ingeniería de Sistemas y Teoría de la Señal
Universidad de Alicante*

(2) *Instituto Universitario de Física Aplicada a las Ciencias y Tecnologías
Universidad de Alicante*

(3) *Dpt. d'Arquitectura de Computadors
Universitat Politècnica de Catalunya*

jfmonllor@ua.es

RESUMEN

En este trabajo se presentan los primeros resultados derivados del desarrollo de un laboratorio virtual para el estudio de las ondas mecánicas en barras con diferentes condiciones de contorno. La aplicación permite visualizar tanto las ondas dentro del sólido como las ondas sonoras producidas por la barra sometida a un impacto. Esta aplicación pretende ser la base de un laboratorio virtual completo que cubra diferentes situaciones didácticas dentro del campo de la Acústica. La interfaz gráfica se ha realizado en MATLAB para que el alumno pueda configurar la experiencia sin tener conocimientos avanzados de programación. La herramienta permite variar de forma sencilla los parámetros físicos de las barras a simular, así como el tipo de fijación en la barra y el punto de excitación. La aplicación representa en tiempo real la distribución de velocidades en el interior de la barra y en aire que la rodea, permitiendo al estudiante identificar de forma visual los diferentes fenómenos que se producen en estos sistemas: ondas longitudinales, ondas transversales y ondas superficiales Love y/o Rayleigh. Se pretende distribuir la herramienta de forma libre a los estudiantes en el siguiente curso académico para evaluar el impacto que este tipo de estrategias tienen en el proceso de enseñanza-aprendizaje.

Palabras clave: Aplicaciones interactivas, Acústica, GUI MATLAB, laboratorio virtual, análisis barras.

1. INTRODUCCIÓN

La generación de aplicaciones interactivas o laboratorios virtuales es una alternativa didáctica y fructífera que permite transferir de forma directa los últimos avances en la ciencia y el desarrollo científico al aula universitaria. En este punto se va a situar el contexto en el cual se enmarca el problema a resolver, para ello se realizará un breve resumen de la bibliografía a la cual se ha recurrido así como los objetivos o propósitos principales del trabajo aquí mostrado.

1.1 Problema/cuestión.

La incorporación de nuevas tecnologías en la educación es un hecho constatado y bien conocido por todos en la actualidad. En este trabajo se muestran los resultados derivados de la incorporación de herramientas de análisis avanzado a aplicaciones interactivas que ayuden a los estudiantes a entender fenómenos físicos complejos.

La estrategia utilizada ha sido la de adaptar una aplicación que implementa un algoritmo complejo como puede ser el método de las Diferencias Finitas en el Dominio del Tiempo (DFDT) para ser utilizada por los estudiantes de forma sencilla. Para ello, se ha recurrido a la implementación de una *Graphical User Interface* (GUI) utilizando el lenguaje de programación MATLAB. El diseño se ha realizado con la finalidad de que el estudiante no tenga que conocer el método numérico y que tampoco sea necesario el que tenga un conocimiento avanzado en lenguajes de programación.

1.2 Revisión de la literatura.

Existen numerosos precedentes en el campo de los laboratorios virtuales y su aplicación a la docencia [Aliane, N. (2010); Max, L. *et al.* (2009); Nonclercq, A. *et al.* (2010); Wolf, T. (2010); Chen, W.-F. *et al.* (2008)]. Por otra parte, el trabajo realizado por Francés *et al.* [Francés *et al.* (2012)] puede considerarse una primera incursión por parte de los autores de este trabajo en el desarrollo de aplicaciones interactivas enfocadas a la docencia y la investigación educativa. En este trabajo, se realizó una interfaz gráfica basada en MATLAB para el caso del análisis de la difracción en redes de apertura, donde se aplicó el método DFDT para realizar el estudio de las ondas electromagnéticas, permitiendo de esta forma realizar el estudio de la luz en su vertiente ondulatoria. A raíz de dicha experiencia, se ha considerado interesante reproducir los mismos pasos, pero en este caso enfocando la temática en el análisis de ondas mecánicas en sólidos y fluidos y su interacción en barras sólidas.

El objetivo en todos los casos de este tipo de aplicaciones es la de crear una interfaz de fácil manejo para los estudiantes, los cuales puedan interactuar con la configuración del método numérico subyacente a la aplicación sin que este proceso sea complejo y tedioso.

1.3 Propósito.

El objetivo de esta interfaz (a partir de ahora denominada AcoWaveSim) es la de constituir un puente entre el alumno y la configuración del método DFDT. Para ello, es necesario tener en cuenta algunas consideraciones. El coste temporal y la complejidad del método DFDT son considerables, por lo que su implementación en lenguaje MATLAB no es una opción de garantías. Sin embargo, la creación de interfaces y el post-procesado de la información resulta más sencillo de realizar si utilizamos MATLAB como herramienta de análisis, por lo tanto para poder hacer esto es necesario crear un nexo entre la GUI y el método DFDT. Para ello, se ha seguido la misma estrategia que en. [Francés *et al.* (2012)]. La aplicación DFDT se ha programado en lenguaje C++ [Jiménez, (2009); Francés *et al.* (2013)], obteniendo una aplicación ejecutable que es invocada (con los convenientes parámetros de entrada) mediante la GUI en MATLAB. Este proceso debe hacerse de forma transparente al usuario y de forma sencilla, tal y como se presenta a continuación.

2. DESARROLLO DE LA CUESTIÓN PLANTEADA

A continuación se detallan y presentan las vicisitudes en el desarrollo de la aplicación AcoWaveSim, así como los objetivos planteados en el diseño de la misma.

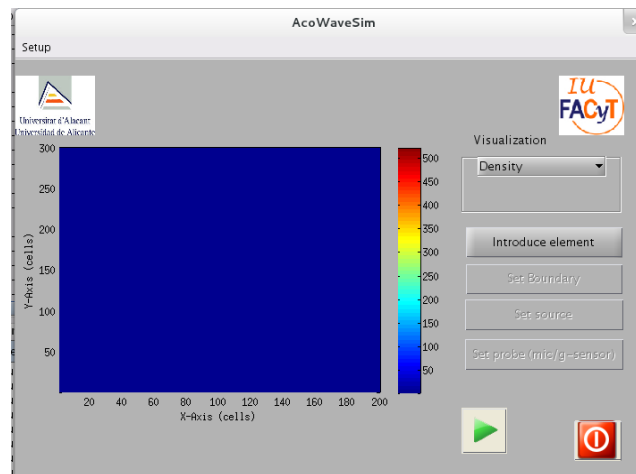
2.1 Objetivos

Tal y como se ha mencionado anteriormente existen dos objetivos principales en el diseño y creación de AcoWaveSim:

- Unificar en un mismo entorno la aplicación DFDT implementada en C++ en la GUI.
- Crear una GUI amigable y fácil de manejar para configurar los parámetros de entrada del código ejecutable encargado de implementar y resolver el problema vibratorio.

Cabe destacar que la aplicación aquí mostrada se encuentra en su versión inicial y se espera añadir numerosas aplicaciones y complementos para extender su aplicabilidad a diferentes aspectos didácticos. A continuación se muestra el aspecto inicial de la aplicación:

Figura 1. Pantalla principal de AcoWaveSim



Tal y como se puede apreciar en la figura 1, se muestra una gráfica que permitirá visualizar la variación temporal de las magnitudes del campo vibratorio (velocidades de partículas y tensiones). En el menú desplegable la densidad es la opción por defecto, sin embargo el usuario puede seleccionar las velocidades o tensiones según su criterio. El objetivo es que a partir de una aplicación sencilla como la mostrada en la figura 1, el usuario sea capaz de extraer información relevante a partir del lanzamiento del programa que se encarga de realizar el cálculo de las velocidades y las tensiones en función del tiempo.

2.2. Método.

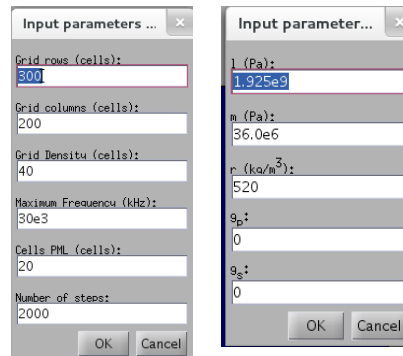
Para poder llevar a cabo esta tarea se ha partido de un código DFDT ya testado y completamente desarrollado [Jiménez, N. (2009); Francés *et al.* (2013)]. Este código presenta las siguientes características:

- Implementación DFDT de la ley de Hooke y de la segunda ley de Newton.
- Simplificación para el caso bidimensional del problema.
- Capacidad para incorporar coeficientes de pérdidas diferentes a las componentes de tensión longitudinal y transversal.
- Capacidad para incorporar medios porosos.
- Implementación optimizada en microprocesadores con instrucciones SSE y AVX.
- Implementación en GPUs CUDA.
- Compatible con OMPI para la ejecución del algoritmo en múltiples computadoras.

En la actualidad la GUI implementada es capaz de configurar los tres primeros puntos. El resto de características se mantienen en su vertiente por defecto a la espera de ser configuradas por la GUI en futuras versiones.

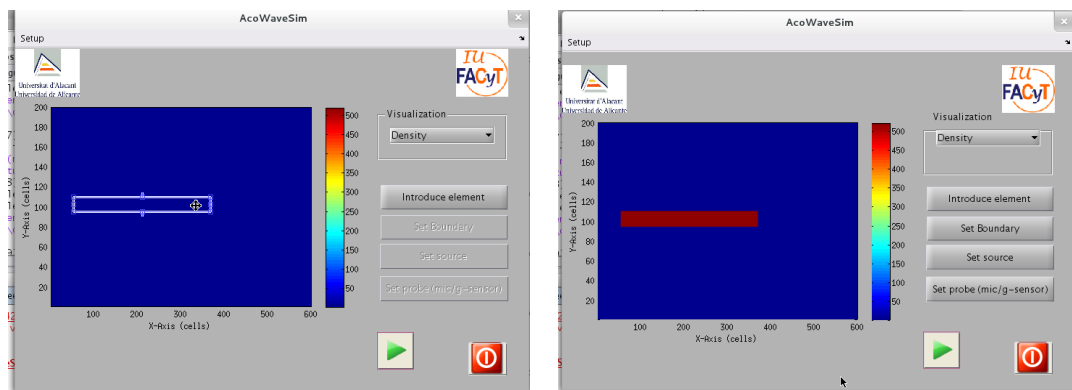
La primera vez que la aplicación AcoWaveSim es ejecutada, se lanza de forma automática la configuración inicial del programa (véase *Set up* en el menú situado en la parte superior izquierda de la ventana de la figura 1). Una vista de estos parámetros se recoge en la figura 2.

Figura 2. Menú de configuración de la malla de simulación del material a incluir.



En esta etapa se fijan ciertos parámetros como pueden ser el tamaño de la malla a simular (en celdas), el número de puntos por longitud de onda y el número de pasos temporales que se desean simular. A priori todos estos parámetros pueden no ser fáciles de configurar. Sin embargo, los valores por defecto que proporciona la aplicación son correctos para un gran abanico de casos prácticos. Una buena alternativa es dejar dichos parámetros inalterados y realizar los primeros estudios con estos parámetros para posteriormente ir refinándolos en función de los intereses de cada caso.

Figura 3. Definición de la barra en AcoWaveSim.

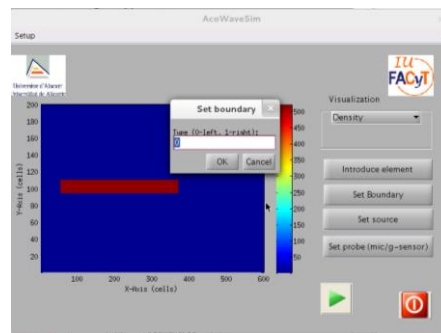


En la figura 3 se muestra el proceso para definir de forma interactiva el tamaño de la barra a simular. Para ello basta hacer clic en el botón *Introduce element* y definir la forma de la barra a introducir en el medio (aire).

A continuación tres botones adicionales se activan una vez finalizada la incorporación de la barra (véase la parte derecha de la figura 3). En concreto, se pueden realizar las siguientes tareas:

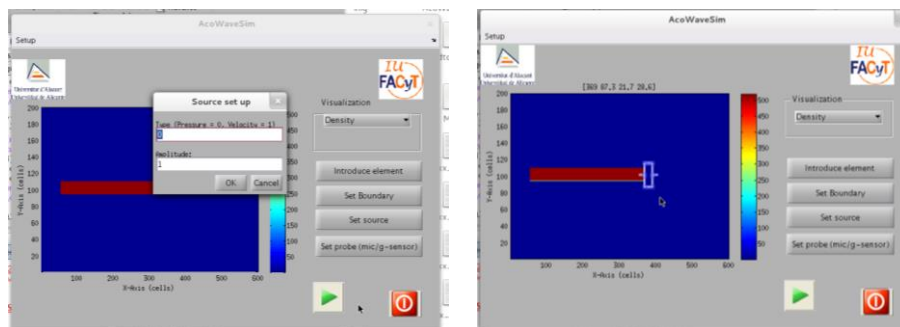
- Aplicar condiciones de contorno fijas: Haciendo clic en este botón (*Set boundary*), el usuario podrá fijar la barra en su extremo izquierdo (valor 0), derecho (valor 1) o en cualquier lugar de la barra (entre 0 y 1, por ejemplo 0,5 sería para una barra fija en punto central). Para ello el usuario deberá de configurar los campos mostrados en la siguiente ventana de diálogo:

Figura 4. Pantalla principal de AcoWaveSim



- Seguidamente se puede incorporar el área donde se va a introducir la excitación (*Set source*). De forma similar al caso anterior, se despliega una ventana de diálogo que permite indicar el tipo de excitación que se desea aplicar: de velocidad o tensión. Así como su amplitud. Por defecto se excitará en el campo de la tensión (proporcional a la presión sonora) con una amplitud unitaria.

Figura 5. Definición de la fuente de excitación en AcoWaveSim.



- El último botón activado permite definir tantos puntos como se desee a lo largo de la malla (véase figura 6). Estos puntos contendrán de forma virtual un micrófono o un sensor de movimiento (acelerómetro) si son establecidos en un fluido (aire) o en el interior de la barra, respectivamente. De esta forma, el usuario puede estudiar a *posteriori* la evolución temporal de las magnitudes involucradas.

Figura 6. Definición de los micrófonos/acelerómetros en AcoWaveSim.

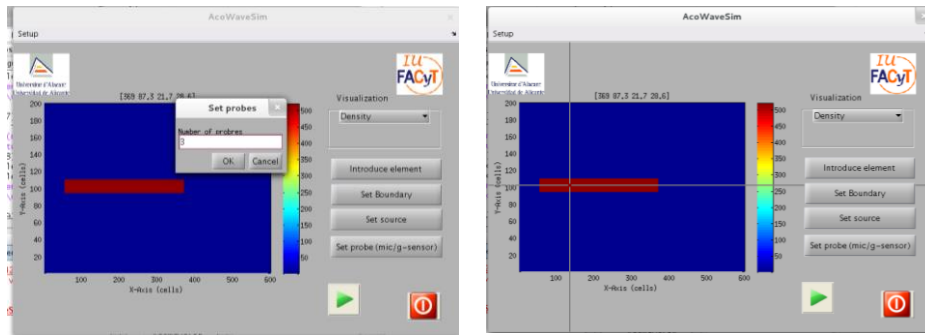


Figura 7. Secuencia de la representación del campo vectorial de velocidad en función del tiempo.

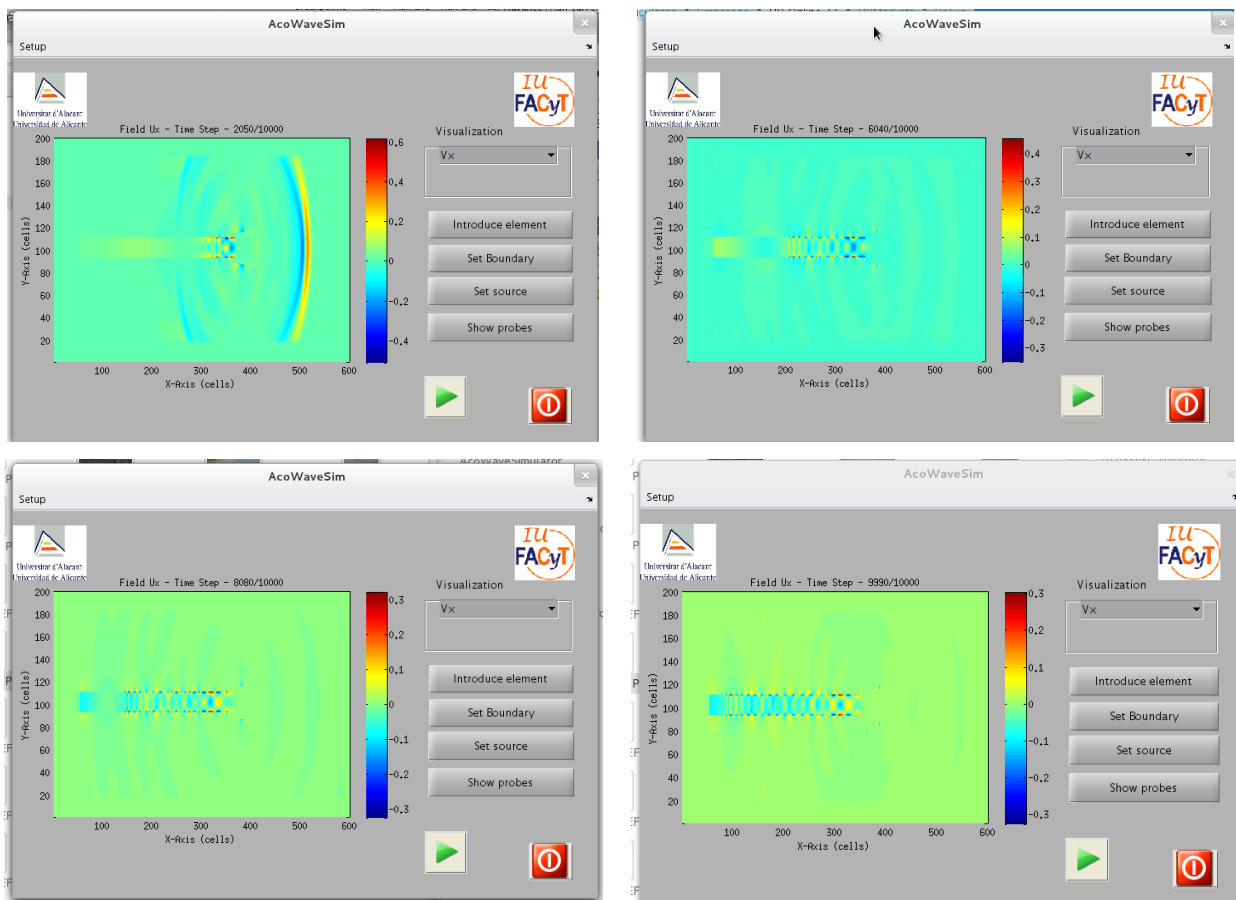
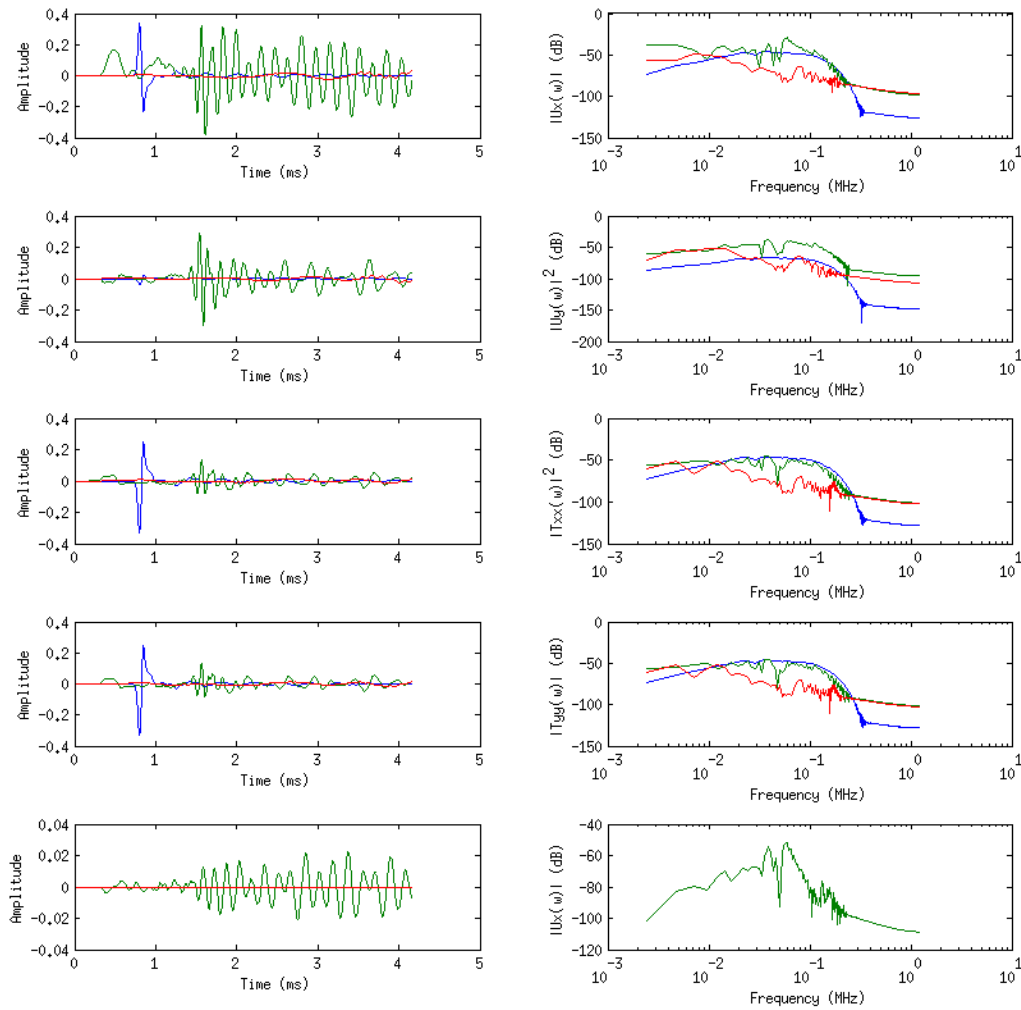


Figura 8. Resultados mostrados en una figura extra de los puntos de monitorización



Finalmente, el usuario podrá presionar el botón de *Play* con el icono verde y comenzar la simulación. El usuario deberá seleccionar en el menú desplegable qué componente quiere visualizar. La figura 7 muestra una secuencia de la simulación realizada para monitorizar la evolución de la velocidad de las partículas en el eje x . En dicha secuencia se puede identificar de forma sencilla la generación de diferentes ondas a lo largo del medio de simulación. En la captura situada en la esquina superior izquierda se puede ver en el aire un gran frente de ondas de tipo longitudinal. Dicho frente se propaga rápidamente a lo largo del fluido. De forma análoga, puede apreciarse la generación de ondas longitudinales en el interior de la barra viajando mucho más rápido y con una longitud de onda mucho mayor (debido a la diferencia de velocidad de propagación en el medio sólido). Otro tipo de ondas más lentas como pueden ser las Rayleigh y/o Love se pueden identificar en todas las capturas a lo largo del contorno

de la barra. Estas ondas se propagan a velocidades relativamente inferiores, sin embargo contribuyen en gran forma en la radiación sonora.

En la figura 8 se muestran un conjunto de gráficas relacionadas con los puntos de captura anteriormente definidos. La columna de la izquierda muestra la magnitud en cuestión para las velocidades en x e y , y las tensiones longitudinales xx , xy y transversales xy (de arriba abajo). En la columna de la derecha se ilustra el espectro de dichas evoluciones temporales con el fin de encontrar tendencias como pueden ser frecuencias de resonancia.

3. CONCLUSIONES

En este trabajo se ha mostrado el proceso de diseño e implementación de una GUI basada en MATLAB, que permite integrar el método DFDT previamente implementado en C++ en una única aplicación. El objetivo principal de esta aplicación es poder proporcionar al estudiante de Acústica en el Grado de Ingeniería en Telecomunicación la posibilidad de estudiar desde otro punto de vista los fenómenos de vibraciones en barras con diferentes puntos de anclaje o condiciones de contorno. Se ha intentado abstraer el experimento para facilitar al usuario la introducción de parámetros y evitar tener que definir o explicar el método numérico subyacente.

Actualmente, la versión implementada está limitada en cuanto a prestaciones y se espera de cara al próximo año académico mejorar y ampliar las características de la misma. Algunas de las opciones que se desean añadir a esta aplicación son las siguientes:

- Adaptar la herramienta para el estudio de barreras acústicas.
- Permitir la configuración avanzada de las características de optimización computacional por parte del usuario.
- Aumentar la capacidad de post-proceso para poder realizar operaciones sobre los datos generados por la aplicación.
- Incorporar un menú de archivo para poder guardar simulaciones y procesados de los mismos.

Finalmente, se puede concluir que la aplicación presenta interesantes características para presentar al estudiante un ejemplo de transferencia directa del campo de la investigación a la docencia. Además, cabe destacar que este tipo de alternativas potencia el desarrollo de las competencias transversales presentes en los títulos de Grado.

Los autores desean agradecer la financiación de la Universidad de Alicante vía los proyectos GITE-09006-UA, GITE-09014-UA, y al ICE de la Universidad de Alicante a través de la convocatoria de Proyectos de Redes 2013-2014 y su soporte a la red 3092.

4. REFERENCIAS BIBLIOGRÁFICAS

- Aliane, N. (2010). A MATLAB/Simulink-based interactive module for servo systems learning, *IEEE Transactions on Education*, 53 (2), pp. 163-271.
- Chen, W.-F. & Wu, W.-H. & Su, T.-J. (2008). Assessing virtual laboratories in a digital-filter design course: An experimental study. *IEEE Transactions on Education*, 51 (1), pp. 10-16.
- Francés, J. & Bleda, S. & Márquez, A. & Neipp, C. & Gallego, A. & Otero B. & Beléndez A. (2013). Performance analysis of SSE and AVX instructions in multi-core CPUs and GPU computing on FDTD scheme for solid and fluid vibration problems. *Journal of Supercomputing*, Early access (Dic-2013), pp. 1-15.
- Francés, J. & Pérez-Molina, M. & Bleda, S. & Fernández E. & Neipp, C. & Beléndez, A. (2012). Educational Software for Interference and Optical Diffraction Analysis in Fresnel and Fraunhofer Regions Based on MATLAB GUIs and the FDTD Method. *IEEE Transactions on Education*, 55 (1), pp. 118-125.
- Jiménez González, N. (2009). *Simulación de tejidos vegetales mediante diferencias finitas*. Tesina de Máster: Universidad Politécnica de Valencia.
- Max, L. & Thiringer, T. & Underland, T. & Karlsson, R. (2009). Power electronics design laboratory exercise for final-year M.Sc. Students, *IEEE Transactions on Education*, 52 (4), pp. 524-531.
- Nonclercq, A. & Biest, V. B. & De Cuyper, K. & Leroy, E. (2010). Problem-based learning in instrumentation: Synergism of real and virtual modular acquisition chains, *IEEE Transactions on Education*, 53 (2), pp. 234-242.
- Wolf, T. (2010). Assessing student learning in a virtual laboratory environment, *IEEE Transactions on Education*, 53 (2), pp. 216-222.