

UNIVERSIDAD POLITECNICA DE VALENCIA

ESCUELA POLITECNICA SUPERIOR DE GANDIA

Master en Ingeniería Acústica



UNIVERSIDAD
POLITECNICA
DE VALENCIA



ESCUELA POLITECNICA
SUPERIOR DE GANDIA

“Obtención de imágenes elastográficas en 2 dimensiones”

TRABAJO FINAL DE MASTER

Autor/a:
Daniel Serra Freese

Tutor/a:
Francisco Camarena Femenia

GANDIA, 2016

[Esta página ha sido intencionalmente dejada en blanco]

Obtención de imágenes elastográficas en 2 dimensiones

Autor:

Daniel Serra Freese

Tutor:

D. Francisco Camarena Femenia

RESUMEN

La técnica elastográfica es la obtención de imágenes que representan las características elásticas de un medio. En este caso, las imágenes se obtienen a partir del análisis de la deformación de las señales ultrasónicas obtenidas de un objeto o cuerpo sometido a una deformación uniaxial estática.

El presente trabajo desarrolla un marco completo de trabajo que va desde la construcción de un modelo numérico, su deformación, el estudio sonográfico y el postprocesado con la finalidad de obtener imágenes elastográficas en dos dimensiones. El marco de trabajo se desarrolla mediante scripts en MATLAB y herramientas para la obtención de imágenes sonográficas como el programa FIELD II basado también en MATLAB.

El objetivo es desarrollar un entorno modular que permita la simulación y obtención de imágenes elastográficas. Cada uno de estos módulos o scripts es fácilmente personalizable, lo que permite adaptarse a diferentes necesidades o introducir mejoras.

Palabras clave: elastografía, simulación, phantom, ultrasonidos, FIELD II

ABSTRACT

Elastographic imaging is the representation of the elastic characteristics of a body or tissue. In the present work, the elastographic images are obtained from the analysis of ultrasonic signals from an object or body under a static uniaxial deformation.

This work develops a comprehensive framework that goes from the construction of a numerical model, its deformation, the sonographic study and postprocessing in order to obtain an elastographic image. The framework is developed using MATLAB scripts and tools for obtaining sonographic images as the FIELD II program, also based on MATLAB.

The aim is to develop a modular environment that allows the simulation of phantoms or models and obtain elastographic images. Each of these modules or scripts are easily customizable, allowing adaptation to different needs or improvements.

Keywords: elastography, simulation, phantom, ultrasound, FIELD II

ÍNDICE DE CONTENIDOS

CAPÍTULO 1. INTRODUCCIÓN	6
1.1 Introducción a la elastografía.....	6
1.2 Motivación y objetivos del trabajo	8
1.3 Organización del trabajo	10
CAPÍTULO 2. FUNDAMENTOS DE LA ELASTOGRAFÍA	11
2.1 Fundamentos teóricos	11
2.1.1. Ejemplo de tres muelles	13
2.2 Detección de la deformación o desplazamiento en señales.....	14
2.2.1. Método del block matching para imágenes.....	15
2.2.2. Correspondencia de bloques en elastografía.....	16
2.2.3. Influencia del Tamaño de kernel.....	18
2.2.4. Detección de la deformación	19
2.3 Función de correlación cruzada en una dimensión	21
2.3.1. Fundamentos de la correlación cruzada	21
2.3.2. Algoritmo de detección desplazamiento mediante la FCC	22
CAPÍTULO 3. ELABORACIÓN DE UN MODELO MEDIANTE EL MÉTODO DE ELEMENTOS FINITOS (FEM)	25
3.1 El método de los elementos finitos.....	25
3.1.1. Introducción al método de elementos finitos.....	25
3.1.2. Resortes y barras como elementos elásticos	25
3.1.3. Condiciones de contorno, equilibrio en los nodos.....	26
3.1.4. Pasos básicos para el cálculo FEM.....	27
3.1.9. Cálculo FEM en MATLAB	27
CAPÍTULO 4. OBTENCIÓN DEL SONOGRAMA	29
CAPÍTULO 4.	29
4.1 Introducción a FIELD II.....	29
4.2 Organización del programa FIELD II	29

4.3	Método de simulación utilizado por Field II.....	30
4.3.1.	La respuesta al impulso espacial	30
4.3.2.	Simulación	31
4.4	Transductor simulado	32
CAPÍTULO 5. RESULTADOS Y DISCUSIÓN.....		33
5.1	Creación y deformación de phantoms.....	35
5.2	Obtención del sonograma.....	39
5.3	Validación de elab_elastograma	42
5.4	Conclusiones	45
ANEXOS		48
A.1	Función elab_phantom	49
B.1	Función elab_sonograma	52
C.1	Función elab_elastograma	55
BIBLIOGRAFÍA.....		57

LISTADO DE ILUSTRACIONES Y FIGURAS

Ilustración 1: Captura de pantalla de Toshiba Accuvix A30.	6
Ilustración 2: Proceso de obtención del elastograma bidimensional.	9
Ilustración 3: Sistema de tres muelles de la misma elasticidad.....	13
Ilustración 4: Sistema de tres muelles de diferente elasticidad.	14
Ilustración 5: Algoritmo básico de <i>blockmatching</i>	15
Ilustración 6: Ejemplo de <i>blockmatching</i> en una imagen 2D.....	16
Ilustración 7: Algoritmo de detección de la deformación mediante la FCC.....	23
Ilustración 8: Región de Interés para la FCC.....	24
Ilustración 9: Elemento finito como sucesión de dos nodos.....	25
Ilustración 10: Ecosonda tipo <i>linear array</i>	32
Ilustración 11: Transductor simulado tipo <i>linear array</i>	32
Ilustración 12: Mapa de elasticidad con inclusión cilíndrica.....	35
Ilustración 13: Dispersores del <i>phantom</i> precompresión.....	36
Ilustración 14: Dispersores del <i>phantom</i> postcompresión (compresión 20%).	37
Ilustración 15: Phantom precompresión y postcompresión, vista 3D.....	37
Ilustración 16: Mapa de desplazamiento y deformación.....	38
Ilustración 17: Líneas de radiofrecuencia de pre- y postcompresión.	39
Ilustración 18: Detalle de patrones de líneas de radiofrecuencia pre- y postcompresión.	40
Ilustración 19: Comparación de ecogramas pre- y postcompresión.	40
Ilustración 20: Sonograma con picos de amplitud muy pronunciada.....	41
Ilustración 21: Líneas de radiofrecuencia de pre- y postcompresión.....	42
Ilustración 22: Elastograma bidimensional (izda. sin filtrar espúrios, dcha. sin espúrrereos)....	43

CAPÍTULO 1. INTRODUCCIÓN

1.1 INTRODUCCIÓN A LA ELASTOGRAFÍA

La elastografía es un procedimiento por el cual se obtienen imágenes que muestran de forma cualitativa las propiedades elásticas de un cuerpo. Esta técnica obtiene la deformación que sufre un tejido mediante la comparación de las reflexiones de pulsos ultrasónicos producidos por un cuerpo antes de ser deformado y después de ser deformado.

Esta técnica es complementaria a la ecografía y fue ideada hace más de 25 años por Jonathan Ophir *et al.* [1], pero no es hasta finales de los años noventa y principios del 2000 [2] que aparecen en el mercado las primeras máquinas elastográficas como por ejemplo la *ElastoScan™* [3] de Samsung Medison o *ElastoQ™* de Toshiba [4]. Esta técnica, permite representar las propiedades elásticas de las diferentes zonas de un cuerpo u objeto.

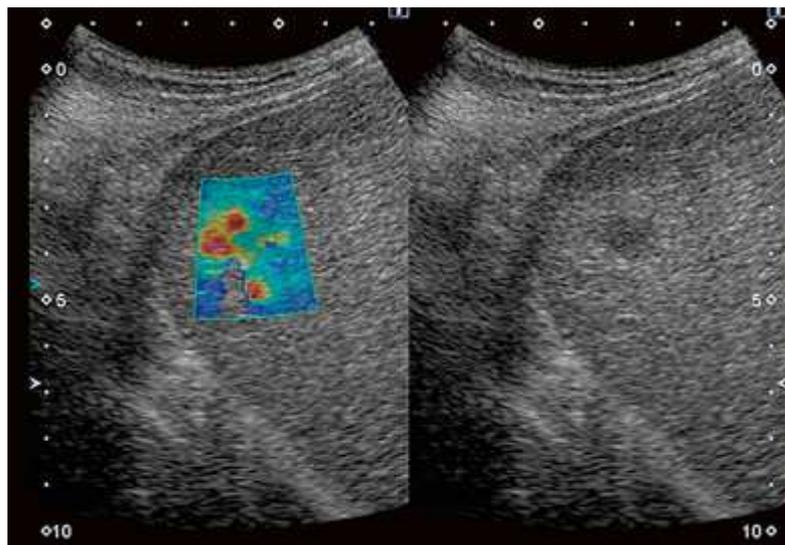


Ilustración 1: Captura de pantalla de Toshiba Accuvix A30.

Dadas las limitaciones computacionales de la época, no ha sido hasta la década de los 90 cuando Jonathan Ophir empezó a obtener las primeras imágenes cuantitativas sobre la deformación y la distribución del módulo elástico en el tejido utilizando aparatos de ultrasonidos [5]. El principio de funcionamiento de la elastografía consiste en la comparación de dos fotogramas obtenidos a partir de un barrido espacial mediante pulsos ultrasónicos del tejido con y sin deformación. Los métodos matemáticos, para la detección del desplazamiento de la señal ya existían pero requieren un gran número de operaciones que a su vez supone un elevado coste computacional, no factible para la época.

Los dispositivos de elastografía dinámica, normalmente y en los casos más sencillos, constan de un transductor de ultrasonidos (5 MHz) acoplado sobre un eje vibrador que genera una vibración a baja frecuencia (50 Hz) y amplitud que provoca una deformación en los tejidos [6]. Las señales recogidas permiten determinar el tiempo que tarda el pulso ultrasónico en propagarse, con lo cual, si se ha producido una deformación, este tiempo también variará respecto del original.

Lo que hace interesante a esta técnica, es que se trata de una técnica de inspección no destructiva, sin radiaciones nocivas y de coste reducido si se compara otras técnicas como la tomografía axial computarizada o la resonancia magnética. Esto es interesante en el campo de la medicina, ya que permite realizar numerosos estudios sin riesgo alguno, incluso pudiéndose utilizar incluso en embarazadas.

En la práctica, hoy en día la forma más común de detectar lesiones en órganos como las mamas o la próstata se sigue realizando mediante la palpación manual, completándose los estudios con biopsias, radiodiagnóstico u otros estudios clínicos. La elastografía permite detectar cambios de elasticidad, como nódulos o fibrosis en zonas poco profundas y posteriormente caracterizar y clasificarlas [7]. En cuanto a la naturaleza de las lesiones, estudios recientes han demostrado que las lesiones pueden ser de 5 a 25 veces más duras que el tejido sano circundante [7]. Esta diferencia de elasticidad será fundamental para poder aplicar las técnicas elastográficas.

Las propiedades de los ultrasonidos hacen que estos puedan viajar varios centímetros por dentro del cuerpo, siendo la limitación principal la longitud de onda y la absorción del medio [8]. Esto hace necesario que se deba llegar a un compromiso entre resolución y profundidad de penetración, haciendo que esta técnica no sea viable para las zonas más profundas del cuerpo humano, como pueden ser los tejidos de riñón e hígado.

Actualmente, la elastografía está sufriendo constantes mejoras debidas a la aplicación de nuevos algoritmos, la mejora en los métodos de captación de las señales ultrasónicas y el aumento de la potencia computacional [1]. También han evolucionado otras técnicas elastográficas basadas en ondas electromagnéticas (rayos X) que permiten obtener resultados similares a las técnicas ultrasónicas [9].

1.2 MOTIVACIÓN Y OBJETIVOS DEL TRABAJO

Este trabajo es una evolución y continuación del Trabajo Final de Carrera con título *Obtención de imágenes elastográficas en una dimensión* [10]. En este trabajo se obtuvieron imágenes elastográficas unidimensionales utilizando un *phantom* realizado con gelatinas de diferentes durezas. Para la obtención de las señales de radiofrecuencia se utilizó una sonda de un único elemento piezoeléctrico, con lo cual únicamente se consiguió un ecograma unidimensional a partir del cual se obtuvo una única línea con la dureza del tejido, es decir, una representación unidimensional de la dureza del tejido.

El objetivo principal de este trabajo es obtener imágenes elastográficas bidimensionales mediante la simulación de un tejido sometido a una compresión, siendo una continuación y ampliación del trabajo inicial.

El hecho de no disponer de una sonda ecográfica de varios elementos, hace necesario utilizar técnicas de simulación numérica. Para ello se construirá un modelo numérico o *phantom* tridimensional al que se pueda aplicar una compresión y que sirva como objeto de estudio para los siguientes pasos.

A continuación se simulará una ecosonda de 128 elementos que servirá para obtener un ecograma bidimensional del modelo anteriormente creado. El programa utilizado para obtener los ecogramas o líneas de radiofrecuencia es Field II, el cual utilizará el *phantom* anteriormente creado. Se obtendrán las líneas de radiofrecuencia (una señal de radiofrecuencia por cada elemento de la ecosonda) de los modelos *phantoms* pre- y postcompresión.

Finalmente, se procesarán las señales, mediante técnicas de identificación de patrones (función de correlación cruzada) para detectar la deformación del *phantom* a partir del desplazamiento en las señales de radiofrecuencia. A partir de la deformación se calculará la dureza del modelo.

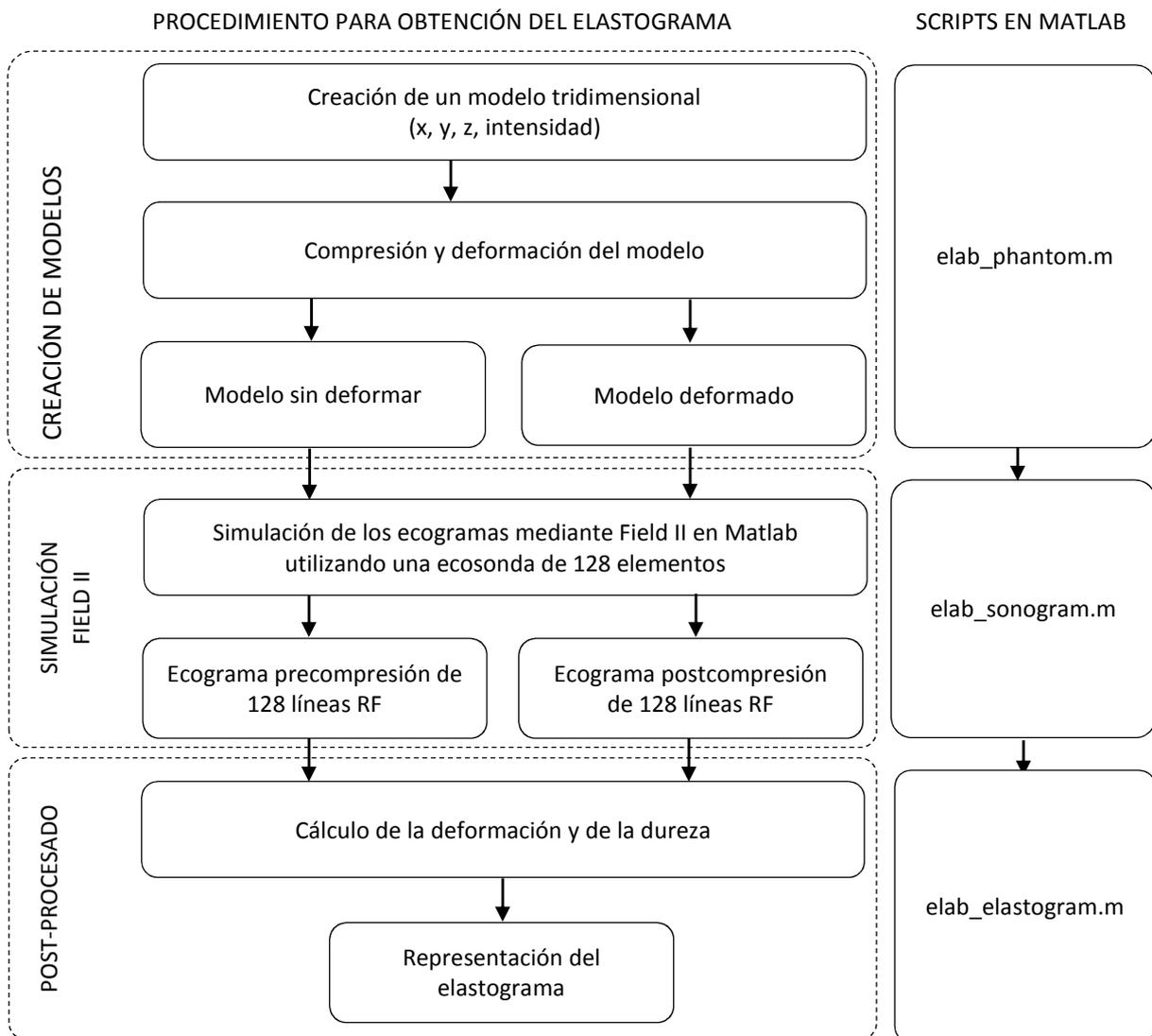


Ilustración 2: Proceso de obtención del elastograma bidimensional.

Resumiendo, para la elaboración del trabajo son necesarias varias tareas, que se proponen como objetivos secundarios del trabajo

1. Elaborar *phantom* y un modelo numérico mediante FEM (*FEM, Finite Element Method*) que sirva como modelo de estudio y al cual se pueda aplicar una deformación
2. Utilizar Field II y MATLAB para simular la propagación de ultrasonidos por el modelo anterior y obtener las señales de radiofrecuencia necesarias.
3. Realizar las tareas de post-procesado para obtener una imagen elastográfica a partir de las señales de radiofrecuencia recogidas.

1.3 ORGANIZACIÓN DEL TRABAJO

El trabajo está dividido en dos bloques, el primer bloque está formado por los conceptos teóricos que se han aplicado en el trabajo y el segundo bloque, de los resultados obtenidos en el trabajo.

El primer bloque consta del presente capítulo y tres capítulos de los fundamentos teóricos que han servido de base para la elaboración del trabajo.

Capítulo 2. Fundamentos de la elastografía. Se explican brevemente los fundamentos de la elastografía y el método de identificación de patrones que permite determinar la deformación o desplazamiento de señales.

Capítulo 3. Elaboración de un modelo mediante FEM. En este capítulo se explica cómo se ha utilizado el método FEM para deformar el modelo numérico o phantom que sirve como objeto de estudio.

Capítulo 4. Obtención del sonograma. Se explica brevemente el programa Field II y su utilidad en el trabajo.

El segundo bloque está formado por los resultados y la discusión de los mismos. Finalmente, se proponen las mejoras para futuros trabajos

Capítulo 5. Resultados y discusión.

Capítulo 6. Mejoras y propuestas para futuros trabajos.

Finalmente, en el Anexo 1 se adjuntan los programas de MATLAB debidamente documentados y se detalla el funcionamiento de los mismos.

CAPÍTULO 2. FUNDAMENTOS DE LA ELASTOGRAFÍA

2.1 FUNDAMENTOS TEÓRICOS

Varios años atrás, algunos autores crearon un nuevo método al que nombraron elastografía para la obtención de imágenes de deformación y del módulo de Young de tejidos. La técnica elastográfica tiene algunas características particulares que se describen a continuación.

La deformación aplicada al tejido no es vibratoria, más bien quasi-estática. Esto tiende a evitar problemas relacionados con reflexiones, ondas estacionarias y modos que podrían interferir con la calidad de la imagen.

La deformación quasi-estática uniaxial reduce la complejidad de la ecuación unidimensional de ondas con un movimiento forzado [11].

$$M \frac{d^2 x}{dt^2} + R \frac{dx}{dt} + Kx = F_0 e^{j\omega t} \quad (2.1)$$

Esta ecuación contiene términos controlados debidos a la inercia (M), la viscosidad (R) y la rigidez (K), y donde x es el desplazamiento, F_0 es la fuerza y ω la frecuencia angular. Si simplificamos la ecuación, para un caso estático ($\omega = 0$), podemos extraer el parámetro de la rigidez en función del desplazamiento de la forma $Kx = F_0$. Dado que x y t son constantes, los términos de la aceleración y de la velocidad desaparecen. Para el caso continuo la ecuación equivalente es $\varepsilon E = \sigma$, donde ε es el módulo elástico, E es la tensión y σ es la fuerza aplicada [12].

Las deformaciones que se aplican normalmente en tejidos son muy pequeñas, del orden de 0,01 (equivalente al 1% de la deformación). Estas deformaciones son lo suficientemente pequeñas para mantener la ecuación de Hooke en el rango lineal. También se mantienen estas deformaciones pequeñas para mantener la distorsión de las señales desplazadas en el tiempo de los ecos en un mínimo debido al ruido de decorrelación en el elastograma. Si se pueden realizar las correcciones pertinentes, se puede aumentar la deformación para ganar contraste en el elastograma [5].

Cuando un medio elástico se comprime por una deformación uniaxial, todos los puntos del medio experimentan un nivel resultante de tensión a lo largo del eje de deformación. Si uno o

varios elementos tienen un parámetro de rigidez diferente, la deformación resultante será mayor o menor; un elemento con mayor rigidez sufrirá una menor deformación que un elemento con una menor rigidez [13].

Esto se consigue adquiriendo un conjunto de señales digitalizadas de radiofrecuencia (RF) de los ecos de los tejidos de la región de interés (*ROI, Region of Interest*) deformando el tejido con el transductor (o con una combinación transductor/compresor) a lo largo del eje de radiación sobre un orden de 0.01 (cerca de 1 por ciento de la profundidad del tejido) y adquiriendo un segundo conjunto de señales postcompresión. Se puede estimar la diferencia de llegada entre los ecos de precompresión y postcompresión mediante la utilización de algoritmos de correlación [13].

Cuando se aplica una presión muy ligera (produciendo una deformación del 1 % sobre el tamaño total del modelo) al tejido con una sonda en la dirección del haz, la mayoría del desplazamiento será en la dirección de la propagación del pulso de ultrasonido. Esto nos permite aproximar el cálculo de la deformación a un modelo n de muelles en 1D. En primer lugar se calcula el desplazamiento $\delta(z)$ en cada punto $n(z)$ de la dirección del haz (z). Este desplazamiento se a partir del cálculo del desplazamiento de las señales de pre- y postcompresión. A continuación, la deformación se calcula como el diferencial del desplazamiento, que es, el ratio de diferencia en el desplazamiento entre dos puntos [2].

$$\varepsilon = \frac{d\delta}{dz} \rightarrow \frac{\delta_{post} - \delta_{pre}}{L} \quad (2.2)$$

2.1.1. EJEMPLO DE TRES MUELLES

Para entender el cálculo de la deformación, el ejemplo más utilizado es el ejemplo de los tres muelles. En un primer caso consideraremos un sistema de tres muelles, sin la aplicación de fuerza alguna, y con la misma constante elástica (k) que representarán los módulos elásticos de diferentes regiones del tejido. Además consideraremos los muelles como cilindros de una longitud L .

Al aplicar una fuerza, se producirá una deformación sobre la longitud de todos los muelles, esta será proporcional e igual para cada muelle. Este ejemplo representa muy bien el comportamiento de un material elástico, homogéneo e isotrópico.

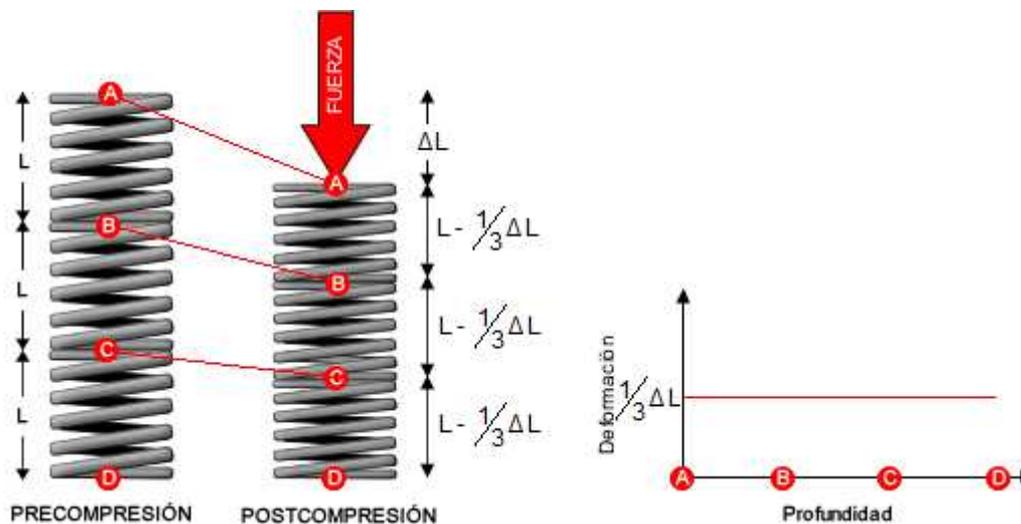


Ilustración 3: Sistema de tres muelles de la misma elasticidad

Ahora si se aumenta la constante elástica de un muelle y volvemos a aplicar la fuerza, la deformación ya no será igual para todos los muelles. Los muelles con menor constante elástica se habrán deformado más que el muelle con la mayor constante elástica. Este ejemplo es válido para materiales elásticos, heterogéneos e isotrópicos.

Si se mide la deformación de cada muelle (incremento de longitud) se podrá obtener información sobre la elasticidad de cada muelle (conociendo la deformación total). Si se extrapola este ejemplo al ámbito natural se observa que el comportamiento de la mayoría de los tejidos animales es muy similar.

En el siguiente caso se substituirá uno de los muelles por otro más duro o casi incompresible. Tras aplicar la fuerza y obtener el diagrama de deformación observaremos que la deformación en la zona comprimida es cero o casi cero. Esta información de deformación nos dará la información necesaria para realizar el elastograma.

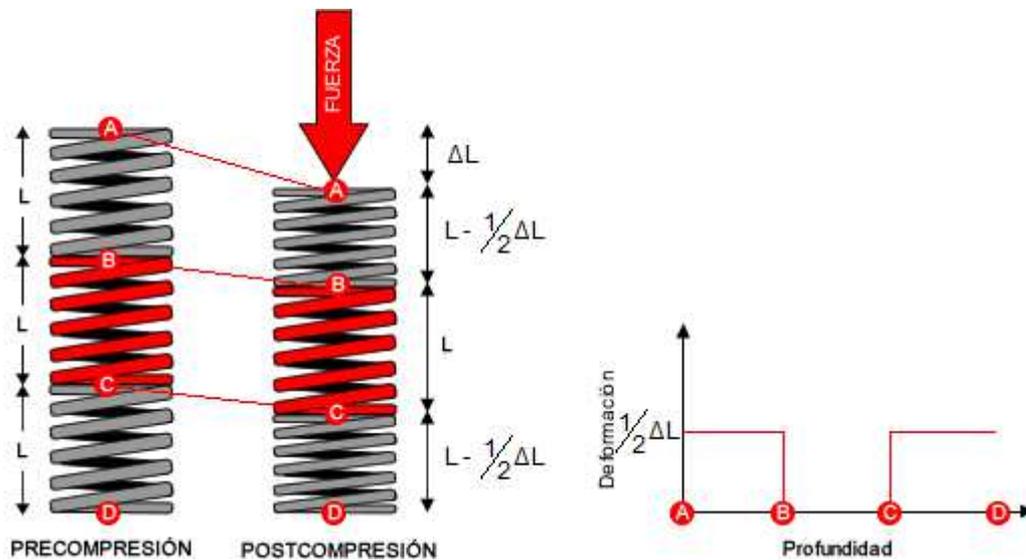


Ilustración 4: Sistema de tres muelles de diferente elasticidad.

2.2 DETECCIÓN DE LA DEFORMACIÓN O DESPLAZAMIENTO EN SEÑALES

La mayoría de los algoritmos para el procesamiento de elastogramas se utilizan en aplicaciones del tratamiento digital de imágenes. Los algoritmos de detección de desplazamientos son muy utilizados en las compresión de video (p. ej. Compresión MPEG) para detectar el desplazamiento de bloques y no tener que redefinirlos otra vez. Otro ámbito de aplicación del *blockmatching* es en la detección de patrones de señal como son señales de radar. En este capítulo explicaremos la técnica utilizada para determinar el desplazamiento de las señales radiofrecuencia y representar los mapas de deformación.

2.2.1. MÉTODO DEL BLOCK MATCHING PARA IMÁGENES

En nuestro caso, esta técnica ha sido adaptada para detectar el movimiento en un tejido cuando es sometido a una pequeña presión. La técnica de *blockmatching* (o correspondencia de bloques) se basa en algoritmos de búsqueda, comparando un patrón (también llamado *kernel*) con otro patrón y estableciendo si se corresponden o no. La función utilizada en este trabajo es la Función de Correlación Cruzada o FCC. En este caso los vectores de movimientos indican cuanto se ha desplazado un bloque de $n \times m$ de una posición i a otra posición j de otra señal.

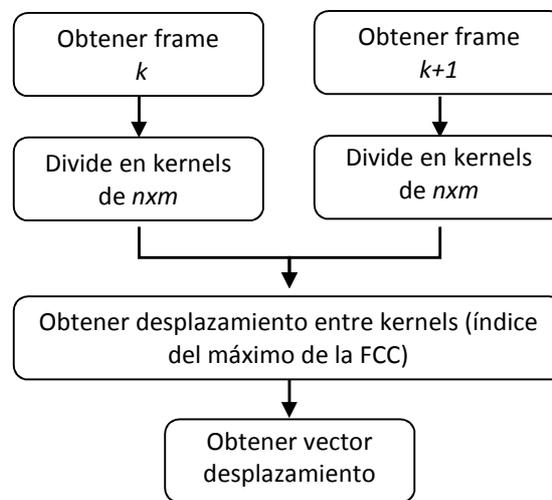


Ilustración 5: Algoritmo básico de *blockmatching*

Una vez obtenidos los vectores de desplazamiento ya se realizará con ellos las operaciones oportunas. Normalmente se establece una región de interés (*ROI, Region of Interest*) para evitar ambigüedades entre dos patrones y acelerar la búsqueda. Existen numerosas funciones para calcular el desplazamiento o reconocer patrones vectoriales o matriciales:

- Función de correlación cruzada normalizada (NCC, Normalized Cross Correlation).
- Suma de diferencias al cuadrado (SSD, Sum of Squared Differences).
- Suma de diferencias absolutas (SAD, Sum of Absolute Differences).
- Pel Difference Classification (PDC)

En la siguiente figura se puede observar el proceso de identificación de patrones mediante la técnica de *blockmatching*. La imagen de entrada es dividida en varias regiones del cual se toma un bloque o *kernel*. El siguiente paso consiste en identificar la nueva posición del *kernel*

mediante alguna de las funciones anteriores. Finalmente se obtiene la nueva posición del bloque y con ello el vector de desplazamiento.

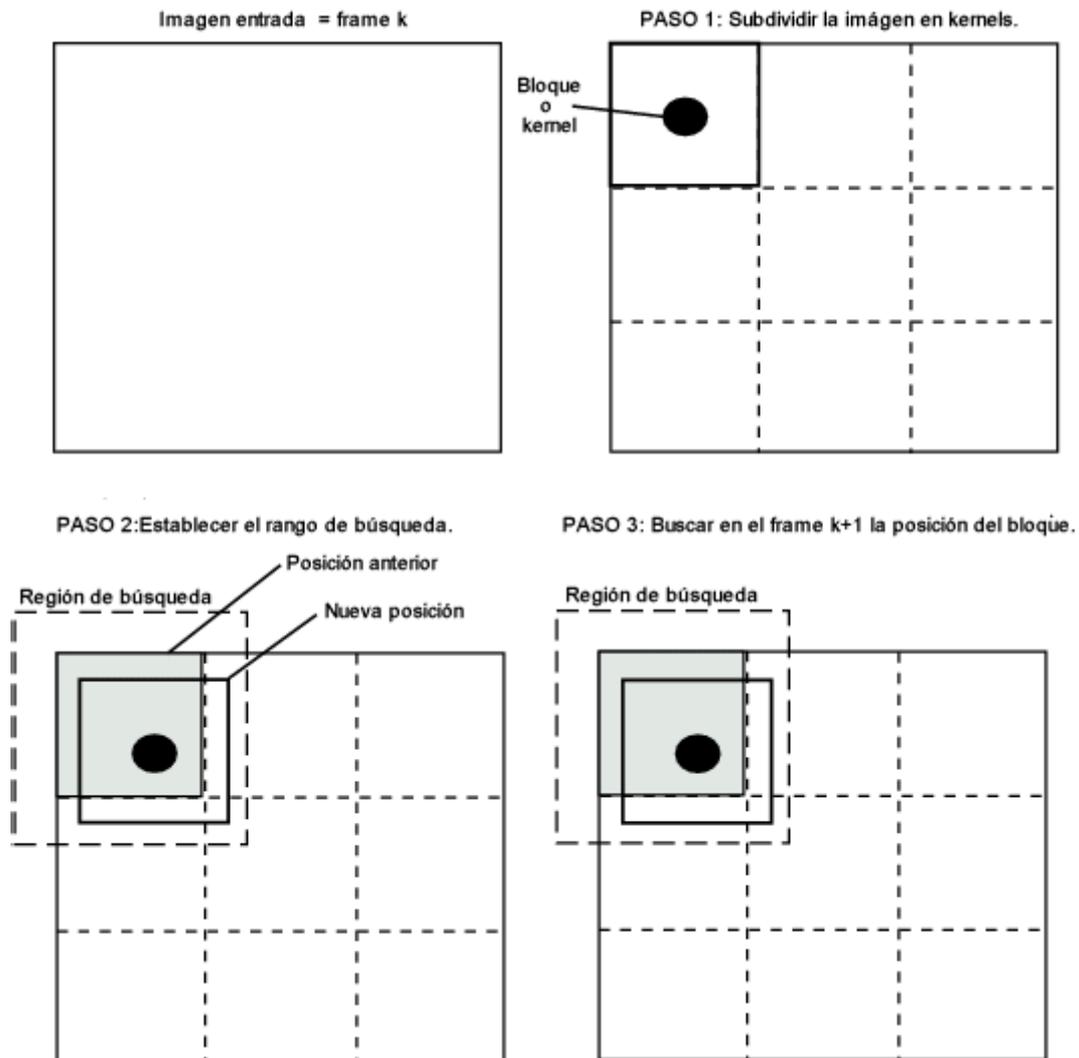


Ilustración 6: Ejemplo de *blockmatching* en una imagen 2D

2.2.2. CORRESPONDENCIA DE BLOQUES EN ELASTOGRAFÍA

Para determinar el desplazamiento de los componentes de dos mapas de valores será necesario al menos de un patrón y de una zona donde buscar (ver punto anterior). Para nuestro caso, se hace necesario que obtengamos como mínimo dos señales, una antes de producirse el desplazamiento (precompresión) o deformación y otra después de que esta se produzca (postcompresión).

La detección del desplazamiento será más fácil cuanto más se parezcan las señales entre sí. Si las dos señales han variado (desplazado o deformado) mucho los resultados devueltos por los métodos de *blockmatching* pueden ser erróneos y producir ambigüedades. Esto es una limitación que se impone a la hora de capturar la señal postcompresión, normalmente se aconseja que este valor no supere el 5% [1]. A mayor deformación, mayores serán los errores de localización.

Como se ha comentado anteriormente, las señales se dividirán en bloques o *kernels* (núcleos, ventanas o patrones) y se realizará la comparación entre los dos bloques. Si la forma de onda en un *kernel* es única, esta comparación ofrece una medida de desplazamiento igualmente única. Si la región de interés o de búsqueda es demasiado extensa puede que existan patrones más parecidos que el que realmente debería haber sido detectado. Para este caso, en nuestro trabajo hemos descartado posibles ambigüedades mediante un enventanado de *Hanning*. En el capítulo de discusión de este trabajo, se exponen los resultados obtenidos para los diferentes tamaños de *kernels*.

Dado que en nuestro caso disponemos una señal dependiente de una sola variable $f(t)$ extrapolaremos el método de *blockmatching* a una dimensión. La posibilidad de que se produzcan ambigüedades en una dimensión es mucho mayor que en dos, ya que la unicidad de los conjuntos es menor; p. ej. un pulso compuesto por una señal senoide se repite cada T instantes, si el *kernel* es mayor que T entonces el desplazamiento obtenido será erróneo.

Suponiendo que estamos ante una señal de postcompresión, con una deformación muy pequeña (como máximo 5%), si tomamos una muestra de la señal precompresión y buscamos el mismo patrón en la señal postcompresión, sabemos que bastará con buscar en las cercanías. A esta zona la llamaremos Región de Interés (ROI, *Region of Interest*). Esta suposición ya nos reduce de manera importante el número de operaciones a realizar y los errores producidos por coincidencias con zonas muy lejanas que no tienen ninguna relación con el patrón pero que se parecen.

En la bibliografía consultada mencionan métodos para reducir el coste computacional y obtener imágenes a tiempo real [1]. Algunas de ellas consisten en múltiples resoluciones o en la interpolación de los resultados. En nuestro caso, el coste computacional no es importante ya

que se trata de una aplicación que no trabaja en tiempo real y con una cantidad muy pequeña de información.

2.2.3. INFLUENCIA DEL TAMAÑO DE KERNEL

Escoger el tamaño adecuado del *kernel* no es una tarea trivial. En general *kernels* grandes son menos sensibles a ruido, mientras que *kernels* pequeños ofrecen mejores contornos entre estructuras con diferentes elasticidades. Además, cuantos mayores sean el *kernel* y el rango de búsqueda, también será mayor el coste computacional. Uno de los problemas más comunes es cuando surgen ambigüedades producidas por mismas estructuras acústicas.

Un ejemplo para ilustrar el uso del tamaño del *kernel* es en la compresión de videos. En algunos casos cuando hay poco contraste entre zonas, a la hora de comprimir, las zonas se codifican y erróneamente se les atribuye un falso vector de movimiento. El resultado es una imagen de video donde se pueden ver zonas que no se mueven aun habiendo movimiento. No se debe confundir esto con el término de pixelar.

Algunas consideraciones sobre el tamaño de los *kernels*:

- Debemos escoger un tamaño de *kernel* que se ajuste al tamaño de los *successos* o patrones identificables.
- Debemos escoger entre un tamaño grande que produzca pocas ambigüedades y un tamaño pequeño que ofrezca buena resolución.
- Un *kernel* no debe abarcar más de dos sucesos similares (*kernel* demasiado grande) ya que puede presentar ambigüedades al compararse con otro *kernel*.
- Aunque normalmente se utilicen *kernels* del mismo tamaño, para el principio de la señal de compresión basta con utilizar un *kernel* pequeño, pero al final se debe utilizar uno más grande que abarque todo el desplazamiento de la señal.
- No hace falta detectar el desplazamiento de todos los puntos, ya que este no presentará discontinuidades y se puede interpolar.

2.2.4. DETECCIÓN DE LA DEFORMACIÓN

La principal dificultad es detectar correctamente el desplazamiento de los ecos de la señal de post-deformación respecto a los ecos de la señal de pre-deformación. Este es uno de los pasos más importantes ya que si se falla aquí no se podrá obtener la deformación del tejido ni representar el elastograma.

En la sección 2.2 *Detección de la deformación o desplazamiento en señales* habíamos hablado de la elasticidad y sus propiedades que venía definida como, utilizando los términos anglosajones:

$$E = \frac{\text{tensile stress}}{\text{tensile strain}} = \frac{\sigma}{\varepsilon} \quad (4.1)$$

Para completar la ecuación que define el módulo de Young definiremos los parámetros de presión y deformación unitaria representados por las letras σ y ε respectivamente.

$$E = \frac{\sigma}{\varepsilon} = \frac{\frac{F}{A}}{\frac{\Delta l}{l_0}} \quad (4.2)$$

De las señales de pre- y postcompresión, se puede obtener el incremento de tiempo entre reflexiones, el cual nos permiten estimar la deformación. Para simplificar los cálculos y poder medir la deformación estableceremos como constante la velocidad en los diferentes medios, representada por el símbolo c , quedando así la deformación en función del tiempo.

$$\Delta l = l_{ini} - l_{fin} \quad (4.3)$$

$$l = c \cdot t \quad (4.4)$$

Combinando las expresiones 4.3 y 4.4 obtenemos la siguiente relación:

$$\Delta l = c \cdot t_{fin} - c \cdot t_{ini} = c \cdot (t_{fin} - t_{ini}) = c \cdot \Delta t \quad (4.5)$$

Ahora si combinamos las ecuaciones 4.2 y 4.5 obtenemos que el módulo de Young en función de Δt .

$$E = \frac{\sigma}{\varepsilon} = \frac{\frac{F}{A}}{\frac{c \cdot \Delta t}{l_0}} \quad (4.6)$$

Dado que las señales pre y postcompresión han sido digitalizadas deberemos hablar de tiempo discreto (muestras) y no de tiempo continuo (segundos). Conociendo la frecuencia de muestreo podremos pasar fácilmente del dominio discreto al dominio continuo.

$$f_s \left[\frac{\text{muestras}}{\text{segundo}} \right] \rightarrow T_s = f_s^{-1} \left[\frac{\text{segundos}}{\text{muestra}} \right] \quad (4.7)$$

$$t = nT_s \quad (4.8)$$

Reagrupando las ecuaciones, podemos establecer una relación entre el dominio discreto y el dominio analógico para obtener Δt .

$$\Delta t = T_s \Delta n \quad (4.9)$$

Finalmente, substituyendo la ecuación 4.9 en la ecuación 4.6 obtenemos la siguiente ecuación que establece el módulo de Young en función del incremento, o desplazamiento, de muestras.

$$E = \frac{\sigma}{\varepsilon} = \frac{\frac{F}{A}}{\frac{c \cdot T_s \cdot \Delta n}{l_0}} \quad (4.10)$$

Para calcular el desplazamiento de muestras partiremos de la señal que hemos obtenido de la interpolación de los puntos válidos, resultado de la función de correlación cruzada. Obtendremos el incremento entre dos puntos sucesivos. En otras palabras, hay que encontrar la pendiente de las rectas de los desplazamientos. Definiremos Δn como:

$$\Delta n = \frac{d(\text{deformacion acumulada})}{dt} \quad (4.11)$$

Para este trabajo se han probado dos métodos diferentes, uno el uso del factor de correlación y el otro el uso de la función de correlación cruzada. Los resultados son muy parecidos pero la función de correlación cruzada presenta algunas mejoras frente al factor de correlación.

2.3 FUNCIÓN DE CORRELACIÓN CRUZADA EN UNA DIMENSIÓN

La función de correlación cruzada normalizada es una operación bastante común en el mundo del tratamiento de señales e imágenes. Muchas de las bibliografías que hemos consultado sobre elastografía han obviado este paso y directamente se centran sobre el tema principal de su estudio.

2.3.1. FUNDAMENTOS DE LA CORRELACIÓN CRUZADA

Consideremos dos procesos conjuntamente estacionarios $x[n]$ y $y[n]$, para $n = \pm 1, \pm 2, \dots$. La covarianza cruzada de orden k entre $x[n]$ y $y[n]$, está definida como:

$$\gamma_{xy}[k] = \sum \left((x[n] - \mu_x) \cdot (y[n+k] - \mu_y) \right) \quad (2.3)$$

para $k = \pm 1, \pm 2, \dots$

Como función de k , $\gamma_{xy}[k]$ es llamada Función de Covarianza Cruzada entre $x[n]$ y $y[n]$.

La estandarización de $\gamma_{xy}[k]$ produce la Función de Correlación Cruzada (FCC)

$$\rho_{xy}[k] = \frac{\sum \left((x[n] - \mu_x) \cdot (y[n+k] - \mu_y) \right)}{\sigma_x \sigma_y} \quad (2.4)$$

para $k = \pm 1, \pm 2, \dots$, donde σ_x y σ_y son las desviaciones estándar para los procesos $x[n]$ y $y[n]$

La FCC mide no solamente la fortaleza de la relación, sino también su dirección o desplazamiento respecto el origen. Por esta razón, es importante examinar la FCC tanto para valores positivos de k como para los negativos. Para valores negativos de k , la FCC describe la influencia lineal de los valores pasados de $y[n]$ sobre $x[n]$. El gráfico de la FCC contra k , llamado correlograma cruzado, es útil para visualizar estas relaciones.

Cuando una serie $x[n]$ o $y[n]$ es ruido blanco y hay normalidad podemos asegurar que tienen correlación cruzada nula. En nuestro caso podremos observar valores de FCC bajos cuando el nivel de señal sea cercano al ruido de fondo.

2.3.2. ALGORITMO DE DETECCIÓN DESPLAZAMIENTO MEDIANTE LA FCC

Usando la función de correlación cruzada se puede crear un algoritmo que permita detectar el desplazamiento temporal entre dos señales. Como se ha comentado anteriormente, la FCC nos devolverá dos informaciones de interés, por un lado el valor de la correlación y el desplazamiento. Dado que posiblemente haya ambigüedades en zonas con poco nivel de señal, aprovechando que tenemos el valor de la correlación, seleccionaremos los puntos de interés de las zonas con alta correlación. Posteriormente haremos una interpolación de estos puntos. El algoritmo se detalla a continuación:

1. Se toma una ventana de tamaño w de la señal precompresión.
2. Se toma una ventana de tamaño w de la señal postcompresión.
3. Se realiza la correlación cruzada entre las dos ventanas.
4. Obtenemos del desplazamiento de la señal dentro de esta ventana y la correlación entre ambas ventanas.
5. Obtenemos el desplazamiento respecto el centro de la vector resultado de la FCC.
6. Se desplazan las ventanas sobre las señales y se repiten los pasos 3 a 5 hasta el final de la señal.

Finalmente obtendremos como resultado un vector, conteniendo los desplazamientos para todas las correlaciones. Idealmente el vector debería ir aumentando progresivamente el valor del desplazamiento sin presentar saltos bruscos (ambigüedades). Como no es posible evitar las ambigüedades procedemos a la interpolación de los puntos que consideramos como válidos, para ello seleccionamos las zonas con mayor valor de correlación y de esas zonas seleccionamos los desplazamientos.

1. Del vector con todas las correlaciones seleccionamos los puntos del vector de desplazamiento que estén situados en zonas que estén por encima de un umbral del valor de correlación.
2. Interpolamos esos puntos linealmente y obtenemos desplazamientos acumulados.

Ahora deberemos realizar la derivada o en el caso discreto la primera diferencia de la señal obtenida por interpolación para obtener el desplazamiento de un punto respecto el anterior.

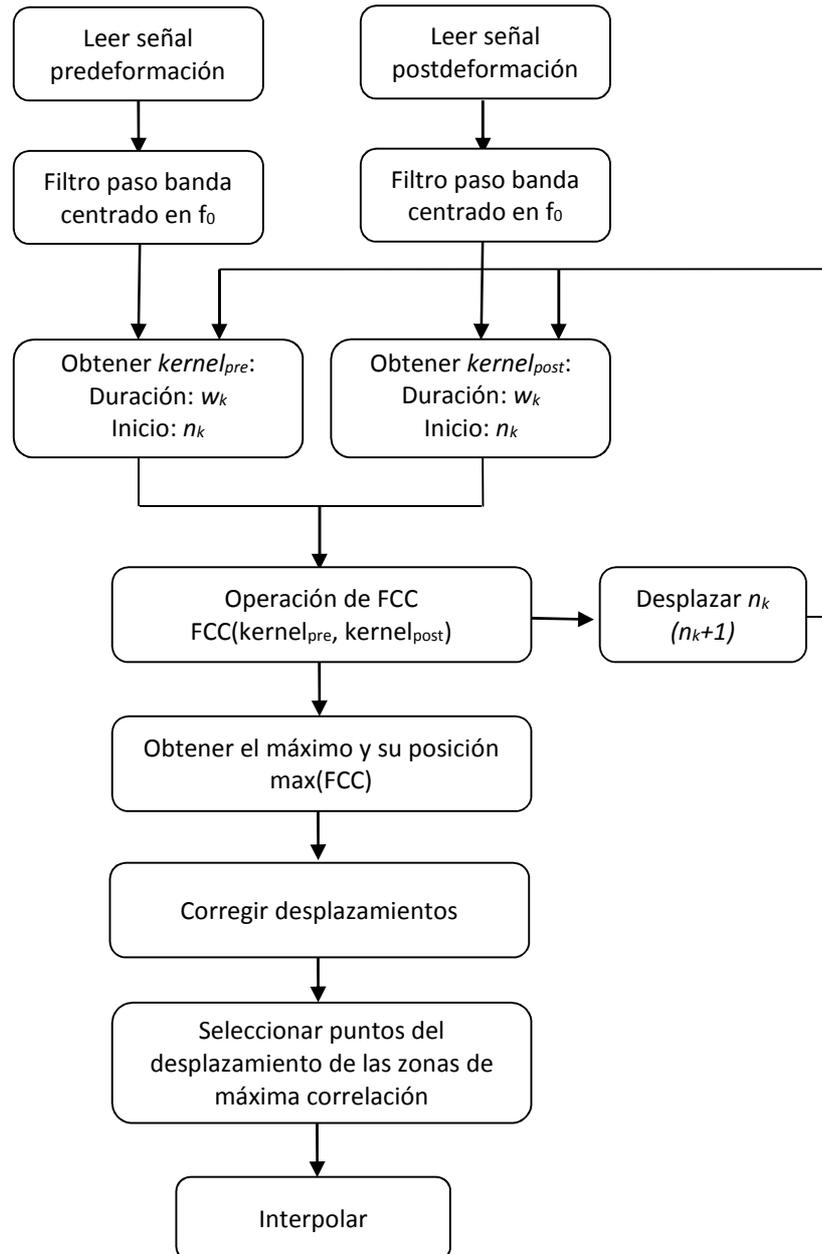


Ilustración 7: Algoritmo de detección de la deformación mediante la FCC.

En la siguiente imagen se puede observar la búsqueda de un patrón postcompresión sobre una señal de precompresión en un una determinada región de interés. La función de correlación cruzada nos dará el valor del desplazamiento (en muestras n) y un valor máximo para cuando las dos señales solapen exactamente.

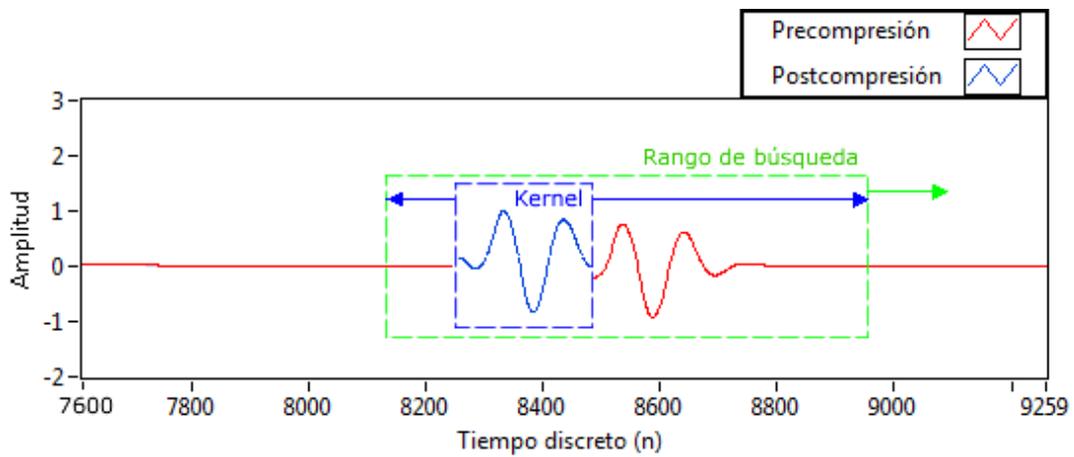


Ilustración 8: Región de Interés para la FCC

Una vez obtenido el desplazamiento acumulado aplicaremos la ecuación 4.11 de la sección 2.2 Detección de la deformación o desplazamiento en señales y obtendremos la el desplazamiento diferencial o la deformación.

CAPÍTULO 3. ELABORACIÓN DE UN MODELO MEDIANTE EL MÉTODO DE ELEMENTOS FINITOS (FEM)

3.1 EL MÉTODO DE LOS ELEMENTOS FINITOS

3.1.1. INTRODUCCIÓN AL MÉTODO DE ELEMENTOS FINITOS

El método de los elementos finitos (FEM), es hoy en día una herramienta muy usada en aplicaciones científicas y de la ingeniería. El método de las diferencias finitas tiene su origen aproximadamente en 1950, con trabajos de Courant, Argyris y Clough [12].

Existen varios libros sobre el método de los elementos finitos, algunos libros recientes tratan sobre la utilización de este método en MATLAB [12]. En esta parte del trabajo se presenta como se ha resuelto el problema de crear un modelo elástico para posteriormente utilizarlo en la obtención de un sonograma.

3.1.2. RESORTES Y BARRAS COMO ELEMENTOS ELÁSTICOS

Consideremos un resorte con dos nodos, dos grados de libertad que corresponden a dos desplazamientos axiales $u_1^{(e)}$ y $u_2^{(e)}$, como se ilustra en la siguiente figura. Supongamos un elemento de longitud L , con una sección constante y área A , con módulo de elasticidad E . El elemento únicamente soporta fuerzas axiales.

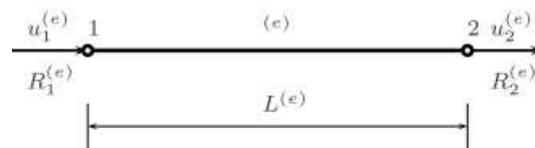


Ilustración 9: Elemento finito como sucesión de dos nodos

El superíndice (e) se refiere a un elemento finito genérico.

La deformación en esta barra se obtiene como

$$\varepsilon = \frac{u_2 - u_1}{L^{(e)}} \quad (3.3)$$

mientras que la tensión en la barra se define por la ley de Hooke como

$$\sigma = E^{(e)} \varepsilon = E^{(e)} \frac{u_2 - u_1}{L^{(e)}} \quad (3.4)$$

La fuerza axial resultante se obtiene como la integración de todas las tensiones a lo largo de la longitud del elemento como

$$N = A^{(e)}\sigma = (EA)^{(e)} \frac{u_2 - u_1}{L^{(e)}} \quad (3.5)$$

Teniendo en cuenta el equilibrio estático de las fuerzas axiales $R_1^{(e)}$ y $R_2^{(e)}$, como

$$R_2^{(e)} = R_1^{(e)} = N(AE)^{(e)} \frac{u_2^{(e)} - u_1^{(e)}}{L^{(e)}} \quad (3.6)$$

Podemos reescribir las ecuaciones de la forma (tomando $k^{(e)} = \frac{EA}{L}$)

$$\mathbf{q}^{(e)} = \begin{Bmatrix} R_1^{(e)} \\ R_2^{(e)} \end{Bmatrix} k^{(e)} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{Bmatrix} u_1^{(e)} \\ u_2^{(e)} \end{Bmatrix} = \mathbf{K}^{(e)} \mathbf{a}^{(e)} \quad (3.7)$$

donde $\mathbf{K}^{(e)}$ es la matriz de rigidez del elemento barra (o muelle), $\mathbf{a}^{(e)}$ es el vector de desplazamiento, y $\mathbf{q}^{(e)}$ representa el vector de fuerzas nodales. Si el elemento sufre la acción de otras fuerzas distribuidas, se requiere que se transformen estas fuerzas en fuerzas nodales como

$$\mathbf{q}^{(e)} = k^{(e)} \begin{bmatrix} 1 & -1 \\ -1 & 2 \end{bmatrix} \begin{Bmatrix} u_1^{(e)} \\ u_2^{(e)} \end{Bmatrix} - \frac{(bl)^{(e)}}{2} \begin{Bmatrix} 1 \\ 1 \end{Bmatrix} = \mathbf{K}^{(e)} \mathbf{a}^{(e)} - \mathbf{f}^{(e)} \quad (3.8)$$

siendo $\mathbf{f}^{(e)}$ el vector de fuerzas nodales equivalente a las fuerzas distribuidas b .

3.1.3. CONDICIONES DE CONTORNO, EQUILIBRIO EN LOS NODOS

En la ecuación 3.6 hemos visto la relación de equilibrio para un elemento, pero también necesitamos obtener las ecuaciones de equilibrio para la estructura. Para esto, necesitaremos ensamblar todas las contribuciones de todos los elementos en un sistema de ecuaciones global. Para hacer esto consideramos que en cada nodo la suma de las fuerzas adyacentes es igual a la fuerza o carga aplicada en este nodo.

Así obtenemos

$$\sum_{e=1}^{n_e} R^{(e)} = R_j^{(e)} \quad (3.9)$$

donde n_e representa el número de elementos de la estructura, produciendo un sistema global de ecuaciones con la forma

$$\begin{bmatrix} K_{11} & \cdots & K_{1n} \\ \vdots & \ddots & \vdots \\ K_{n1} & \cdots & K_{nn} \end{bmatrix} \begin{Bmatrix} u_1 \\ \vdots \\ u_n \end{Bmatrix} = \begin{Bmatrix} f_1 \\ \vdots \\ f_n \end{Bmatrix} \quad (3.10)$$

o de una forma más compacta

$$\mathbf{Ka} = \mathbf{f} \quad (3.11)$$

Aquí K representa el sistema (o estructura) de rigidez, a es el vector de desplazamiento del sistema, y f representa el vector de fuerzas del sistema.

3.1.4. PASOS BÁSICOS PARA EL CÁLCULO FEM

En cualquier problema de elementos finitos los pasos básicos de cálculo y típicos son los siguientes:

1. Definir un conjunto de elementos conectados a nodos.
2. Para cada elemento, calcular la matriz $\mathbf{K}^{(e)}$, y el vector de fuerzas $\mathbf{f}^{(e)}$.
3. Ensamblar la contribución de todos los elementos en un sistema global $\mathbf{Ka} = \mathbf{f}$.
4. Modificar el sistema global imponiendo las condiciones de contorno de desplazamiento.
5. Resolver el sistema global y obtener los desplazamientos a.
6. Para cada elemento, evaluar las deformaciones y tensiones (post-procesado).

3.1.9. CÁLCULO FEM EN MATLAB

Para crear el *phantom* en MATLAB partiremos de la matriz de rigidez que hemos creado a partir de un mapa de bits en escala de grises. En nuestro caso se trata de una compresión uniaxial y estática, que podríamos traducir como la compresión o deformación de un conjunto de barras (o muelles). Así trataremos cada columna de la matriz de rigidez como un sistema de muelles con sus constantes elásticas definidas en esta columna. Si disponemos de una matriz de rigidez de la forma

$$\mathbf{K} = \begin{bmatrix} k_{11} & \cdots & k_{1j} \\ \vdots & \ddots & \vdots \\ k_{i1} & \cdots & k_{ij} \end{bmatrix} \quad (3.12)$$

tendremos j sistemas formados por i elementos (o resortes). Para cada sistema tendremos $i+1$ nodos, de los cuales dos serán fijos o bloqueados.

Ahora, como paso previo deberemos calcular la fuerza necesaria para producir una deformación a la que hemos especificado como variable de entrada. A partir de la ley de Hooke obtenemos que

$$F_T = k_{eq} \Delta x \quad (3.13)$$

donde F_T es la fuerza total que del sistema, k_{eq} es la constante elástica equivalente de la asociación de i resortes en serie, y Δx es la compresión aplicada al modelo.

El siguiente paso es definir las condiciones de contorno para este sistema, en este caso definiremos las fuerzas que actúan sobre los nodos. Como se trata de un sistema estático sabemos que, por convención nuestra, la fuerza aplicada sobre el último nodo es

$$F_i = -F_T \quad (3.14)$$

donde el signo negativo implica una compresión.

Una vez definidas las fuerzas, definiremos las siguientes condiciones de contorno. En este caso serán la libertad de movimiento de cada nodo. El único nodo que no desplazará será el primer nodo, o nodo origen; los demás nodos estarán libres.

Finalmente, componemos el sistema de ecuaciones y resolvemos para la variable que nos interesa, en este caso la deformación o $\Delta x^{(e)}$

Se debe indicar que en este caso hemos tratado todos los elementos como elementos de longitud de unitaria. Con lo cual es necesario realizar el escalado antes y después de introducir los parámetros de compresión u obtener la deformación para cada elemento.

CAPÍTULO 4. OBTENCIÓN DEL SONOGRAMA

4.1 INTRODUCCIÓN A FIELD II

Los *scripts* para el trabajo han sido desarrollados con la versión 3.20 del 19 de noviembre de 2010 del programa Field II. Esta versión del programa se ejecuta en MATLAB 7.1 y puede simular todo tipo de transductores de ultrasonidos y las imágenes asociadas. El enfoque y la apodización de los transductores se pueden controlar de forma dinámica, y es, por tanto, posible simular todo tipo de sistemas de imágenes por ultrasonido. La última versión de Field II también puede utilizarse para obtener imágenes de apertura sintética [12].

La guía o manual de Field II [12] pretende ser una presentación de las rutinas disponibles en la actualidad. Se incluyen unos pocos ejemplos y da una pequeña cantidad de información de fondo.

Los ejecutables del programa pueden ser descargados desde el sitio Web del programa: <http://server.elektro.dtu.dk/www/jaj/field/>

4.2 ORGANIZACIÓN DEL PROGRAMA FIELD II

El programa FIELD II consta de un programa en C y un número de funciones .m o funciones en MATLAB a las que llama a este programa. Todos los cálculos se llevan a cabo por el programa en C, y todos los datos se mantienen por programa de C. FIELD II utiliza tres conjuntos o tipos de funciones para MATLAB:

- i. Funciones para inicializar el programa. Estas funciones están precedidas por el prefijo *field_*.
- ii. Funciones para definir y manipular transductores. Estas funciones están precedidas por el prefijo *xdc_*.
- iii. Funciones para el cálculo del campo acústico. Estas funciones están precedidas por el prefijo *calc_*.
- iv. Otras funciones para manipulación de parámetros. Se pueden manipular parámetros como la velocidad del sonido, la frecuencia de muestreo y apodización de transductores entre otros.

Se dispone de una ayuda para el uso de cada función, para ello se deberá escribir:

```
>> help <nombre de la función>
```

Además de la información en el manual de FIELD II, se dispone de ejemplos muy útiles en la web del programa. Los programas realizados para este trabajo son adaptaciones de alguno de estos ejemplos.

4.3 MÉTODO DE SIMULACIÓN UTILIZADO POR FIELD II

4.3.1. LA RESPUESTA AL IMPULSO ESPACIAL

El sistema del programa FIELD II utiliza el concepto de respuesta al impulso espacial desarrollado por Tupholme y Stepanishen en una serie de documentos y resumido por Jensen [14]. El enfoque se basa en la teoría de sistemas lineales para encontrar el campo acústico tanto para series de pulsos como para el caso de ondas continuas. Esto se hace a través de la respuesta al impulso espacial. Esta respuesta proporciona el campo acústico en un punto específico en el espacio como función del tiempo, cuando el transductor se excita por un impulso de Dirac. El campo para cualquier tipo de excitación a continuación, se puede encontrar por convolución de la respuesta al impulso espacial con la función de excitación. La respuesta al impulso variará como una función de la posición relativa al transductor, por lo tanto la respuesta el nombre de impulso espacial.

La respuesta recibida de una esfera oscilante pequeña se puede encontrar por reciprocidad acústica. La respuesta espacial al impuso respuesta es igual a la respuesta recibida por una onda esférica emitida por un punto. La respuesta total recibida por un pulso puede ser por tanto calculada como la convolución de la función de excitación del transductor con la respuesta al impulso espacial de la apertura de emisión. Si además se tiene en cuenta la función de transferencia electromecánica se puede llegar a la señal de tensión recibida.

Una explicación y demostración rigurosa de esto puede encontrarse en [15].

Se puede utilizar cualquier excitación ya que se está utilizando la teoría de sistemas lineales. El resultado para el caso de una onda continua se encuentra por la Transformada de Fourier de la respuesta espacial al impulso para la frecuencia dada.

4.3.2. SIMULACIÓN

Un gran número de diferentes autores han calculado la respuesta al impulso espacial de geometrías diferentes transductores. Pero en general, es difícil de calcular una solución, y especialmente si se tiene en cuenta la apodización del transductor. Aquí la superficie del transductor no vibra como un pistón, por ejemplo los bordes podrían vibrar menos que el centro del transductor. El programa de simulación resuelve este problema dividiendo la superficie del transductor en cuadrados o triángulos según el caso y la suma de la respuesta de estos para producir la respuesta. De esta manera se puede simular cualquier tipo de geometría para de transductor.

El tiempo de una simulación es también motivo de gran preocupación. Como los cuadrados de la apertura del transductor son pequeños es recomendable utilizar la aproximación de campo lejano, haciendo más simple la simulación. Otra forma para mantener el tiempo de simulación bajo es utilizando una frecuencia de muestreo baja. A menudo, las respuestas impulsivas espaciales se calculan utilizando las frecuencias de muestreo en el intervalo GHz debido a las discontinuidades abruptas de las respuestas. Estas discontinuidades se manejan en FIELD II de forma precisa manteniendo de posición del tiempo de las respuestas y además utiliza la respuesta al impulso espacial integrado como un paso intermedio en los cálculos. De esta manera no se pierde energía de la respuesta lo cual es mucho más importante que tener la forma exacta de la respuesta del impulso espacial. Por todo esto, el programa se recomienda utilizar una frecuencia de muestreo de 100 MHz para cálculos aproximados que no una frecuencia de muestreo de 1 GHz para obtener la respuesta exacta.

Según la teoría bastaría una frecuencia dos veces superior a la frecuencia máxima (Teorema de Nyquist) pero como estamos tratando con señales impulsivas de un gran ancho de banda es recomendable utilizar un frecuencia de muestreo más elevada.

4.4 TRANSDUCTOR SIMULADO

Como apertura de emisión se simula un transductor tipo *linear array* de 192 elementos con 64 elementos activos para cada línea. Se utiliza el mismo transductor como apertura de recepción.



Ilustración 10: Ecosonda tipo *linear array*.

Con FIELD II se puede simular casi cualquier configuración de transductor. Los retrasos para cada elemento se calculan automáticamente a partir de las leyes focales implementadas en el programa FIELD II, solamente hay que indicar coordenadas para la focalización.

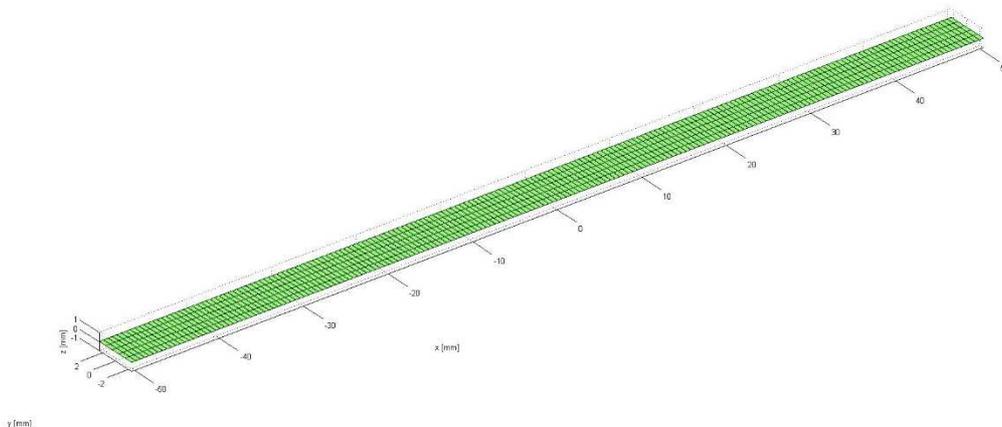


Ilustración 11: Transductor simulado tipo *linear array*.

Este tipo de transductor es uno de los más sencillos que se pueden implementar, además de haberse diseñado para que se igual de ancho que el *phantom*.

CAPÍTULO 5. RESULTADOS Y DISCUSIÓN

En este capítulo se presentan y discuten los resultados obtenidos en los diferentes scripts de MATLAB realizados para elaborar y conseguir los objetivos del presente trabajo.

Los scripts están debidamente comentados, con lo cual es relativamente fácil seguir el procedimiento de cálculo. A continuación se resume brevemente los procedimientos de los scripts.

1. Script en MATLAB elab_init.m. Es un script de inicialización del entorno de trabajo. Sus principales funciones son:

- a. Crear la carpeta del proyecto para guardar los datos del proyecto.
- b. Crear un documento de texto con la información relativa al proyecto (info.txt).
- c. Inicializar la aplicación de Matlab Field II (field_init).
- d. Ejecutar script elab_phantom.m para obtener los *phantoms* de pre- y postcompresión.
- e. Ejecutar el scrip elab_sonograma.m para obtener las líneas de radiofrecuencia de pre- y postcompresión.

2. Script en MATLAB elab_phantom.m. Crea el phantom precompresión y mediante la técnica de los elementos finitos (FEM) lo deforma para crear un *phantom* postcompresión.

- a. Leer los archivos de geometría del modelo (mapa_elastico.bmp) y el mapa de amplitudes de las intensidades de los dispersores (mapa_amplitudes.bmp).
- b. Definir las dimensiones $[lx, ly, lz]$ y resoluciones del modelo tridimensional $[dx, dy, dz]$.
- c. Sembrar dispersores y obtiene sus coordenadas $[x, y, z]$.
- d. Crear modelo de elementos finitos y aplicar compresión para obtener el modelo postcompresión. Se calculan las nuevas coordenadas de $[x, y, z]$.
- e. Representar los modelos para comprobación.
- f. Crea los archivos que contienen los datos para los siguientes pasos de cálculo.
- g. Añade la información al documento de texto (info.txt).

3. Script en MATLAB elab_sonograma.m. Este script ejecuta la simulación de un transductor de ultrasonidos sobre los modelos creados en los pasos anteriores (pre- y postcompresión).

- a. Establece los datos y variables para la simulación mediante Field II.
- b. Carga datos de los *phantoms* almacenados en formato mfile.
- c. Configura el transductor y define los elementos con las funciones de Field II.
- d. Calcula las líneas de radiofrecuencia para ambos modelos pre- y postcompresión.
- e. Guarda cada línea para cálculos posteriores (representación y elastograma).
- f. Representa las señales y sonogramas obtenidos.
- g. Crea los archivos que contienen los datos para los siguientes pasos de cálculo.
- h. Añade la información al documento de texto (info.txt).

4. Script en MATLAB elab_elastograma.m. Este script realiza todo el postprocesado para obtener los elastogramas a partir de las señales de radiofrecuencia calculadas anteriormente de las señales de pre- y postcompresión.

- a. Recortarlas señales al mismo tamaño.
- b. Localizar desplazamientos.
- c. Calcular la deformación.
- d. Mejorar y filtrar los resultados.
- e. Representar resultados.

Para que los scripts funcionen correctamente se necesitan los siguientes archivos (incluidos en los documentos anexos) pero que se pueden obtener de las siguientes fuentes:

Field II. Programa de simulación de transductores acústicos (<http://field-ii.dk/>)

Matlab Statistics Toolbox. Contiene la Función de Correlación Cruzada.

Antes de ejecutar los scripts se deben añadir las rutas al directorio de trabajo de MATLAB con la función de *Set Path...* (File/Set Path...) y establecer las rutas de guardado de los proyectos en el script *elab_init.m*.

5.1 CREACIÓN Y DEFORMACIÓN DE PHANTOMS

Mediante el script de Matlab `elab_phantom.m` se obtienen los *phantoms* pre- y postcompresión que sirven como fuente de datos para los siguientes pasos (obtención del sonograma). Durante la ejecución del script se obtienen diferentes imágenes o plots que permiten el control visual de los *phantoms* creados, aunque para pequeñas deformaciones (del orden de 2%) no se puede apreciar deformación.

Para las pruebas se han creado varios *phantoms* de geometrías sencillas, de las cuales se conoce previamente cómo se comportará el modelo una vez sea sometido a una deformación. En este caso se crea el *phantom* de una inclusión cilíndrica al cual se someterá a una deformación uniaxial del 20% (2 cm sobre 10 cm totales del modelo).

Como mapa elástico se utiliza una matriz bidimensional cargada desde una imagen BMP (BMP, Bitmap de Windows), que por su sencillez nos permite dibujar formas rápidamente. No se ha considerado la normalización de los valores o conversión a unidades elásticas reales del archivo BMP ya que trabajamos con datos relativos o proporciones. Por ejemplo, en la imagen el área blanda tiene un valor de 28 unidades frente a las 178 unidades de la inclusión cilíndrica. Esto da una proporción de 6,35 a 1 ($178/28$), es decir, la inclusión cilíndrica es 6,35 veces más dura que el resto.

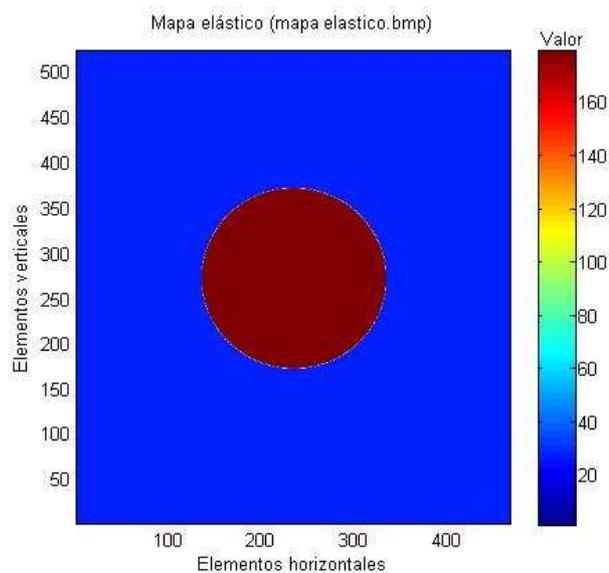


Ilustración 12: Mapa de elasticidad con inclusión cilíndrica

La siguiente imagen muestra una vista bidimensional del modelo de precompresión, formado por 50.000 dispersores. En este caso la escala de colores muestra la intensidad de los dispersores obtenidos del mapa de intensidades o amplitudes. En este caso, el mapa de elasticidad y el mapa de amplitudes es la misma imagen, con el fin de conseguir una representación más sencilla. Así, la inclusión se muestra en color rojo ya que tiene mayor amplitud que el resto del modelo, característica definida en la imagen mapa_amplitudes.bmp.

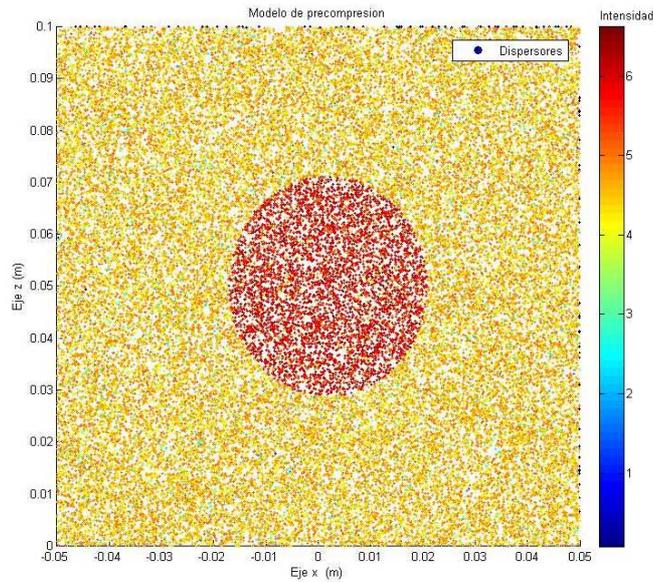


Ilustración 13: Dispersores del *phantom* precompresión.

En la imagen de postcompresión se muestra el phantom comprimido un 20%, que equivale a 2 centímetros sobre los 10 cm totales del modelo. En este caso la compresión aplicada es solo para fines ilustrativos, en la práctica la compresión estará entre el 1 y el 2% y en ningún caso superará el 5%.

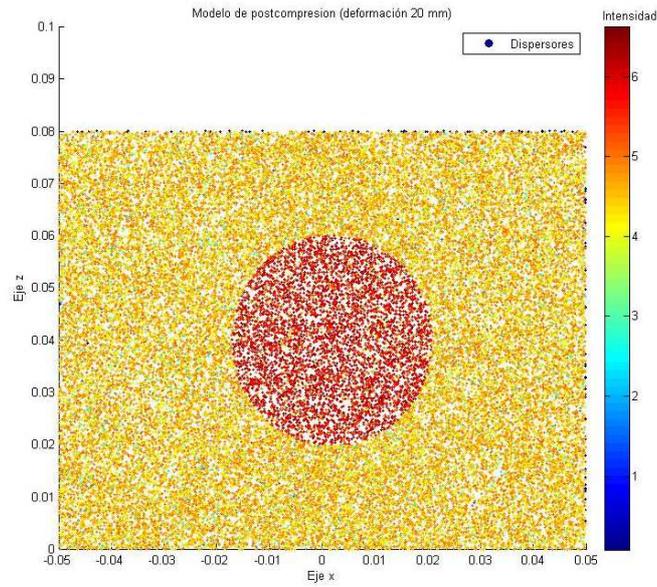


Ilustración 14: Dispersores del *phantom* postcompresión (compresión 20%).

Se puede observar que en las áreas de menor elasticidad han sufrido una mayor compresión como era de esperar, mientras que la zona de mayor dureza se mantiene constante.

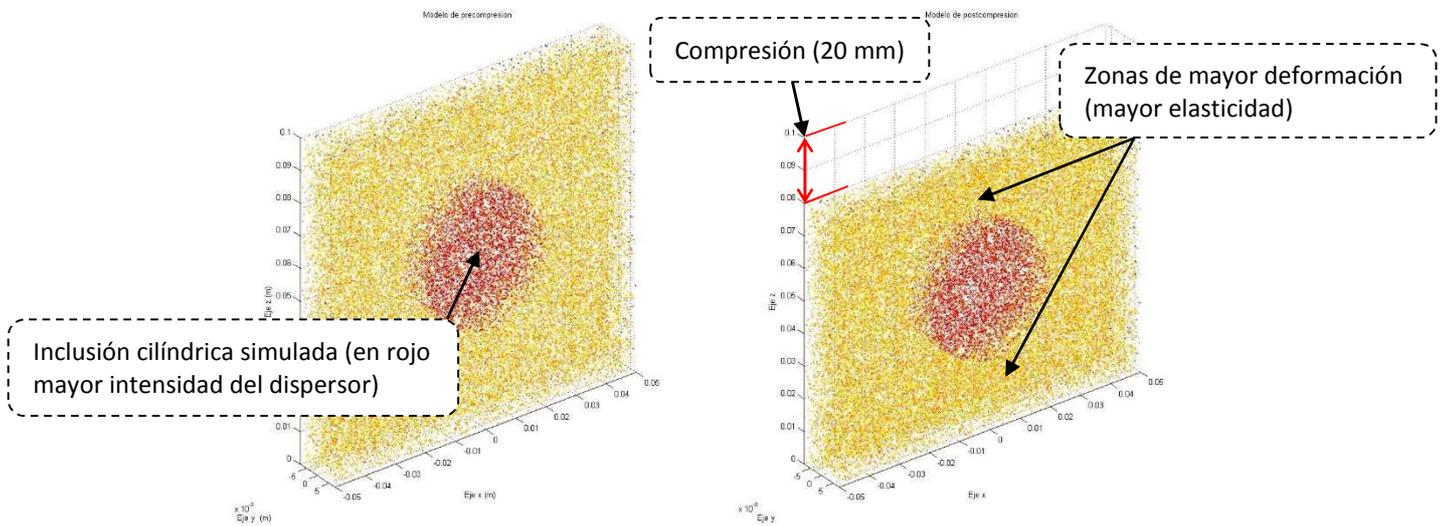


Ilustración 15: Phantom precompresión y postcompresión, vista 3D.

En las siguientes imágenes se pueden ver los valores del Mapa de desplazamiento (imagen izquierda) y deformación (imagen izquierda). En la parte de debajo de la imagen de la izquierda se puede observar que existe mayor desplazamiento que en la parte superior, exactamente $-20 \cdot 10^{-3}$ m, que equivale a los 2 cm de compresión aplicada al modelo. Así se puede interpretar que la parte superior se ha desplazado un total de 2 cm mientras que la base no se ha desplazado (los nodos bloqueados)

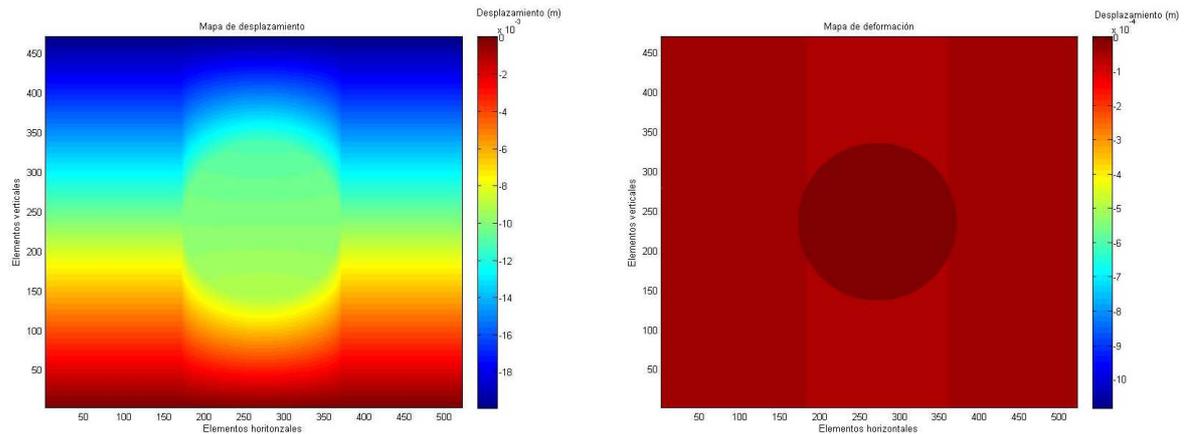


Ilustración 16: Mapa de desplazamiento y deformación.

En el Mapa de deformación (imagen derecha) se obtiene a partir de la primera diferencia de la imagen de desplazamiento. Se observa como en el lado izquierdo de la figura de deformación esta es constante. En la zona de la inclusión cilíndrica se pueden observar dos zonas, por un lado la zona de la inclusión cilíndrica con valores menores, es decir, se ha deformado menos que el resto. Por otro lado, las zonas que están por encima y debajo de la inclusión cilíndrica, estas se han tenido que deformar más para compensar la poca deformación de la inclusión. Para simplificar, podríamos decir que la parte izquierda de la imagen equivaldría al modelo de tres muelles con una misma constante elástica (todos los muelles se han deformado lo mismo) mientras que la parte central de la imagen equivaldría al ejemplo de los tres muelles con diferente elasticidad (el muelle central es más duro y por lo tanto se ha deformado menos).

Hay que recordar, que las imágenes anteriores únicamente se refieren a la creación del *phantom* y se utilizan para control. A continuación se procederá a calcular las líneas de radiofrecuencia y los sonogramas asociados mediante el matscript `elab_sonograma`.

5.2 OBTENCIÓN DEL SONOGRAMA

Una vez se han obtenido los modelos de *phantom* (coordenadas espaciales y amplitud de los dispersores) se procede al cálculo de las líneas de radiofrecuencia y a componer un sonograma.

Una vez terminada la simulación y obtenidas las líneas de radiofrecuencia, el programa las superpone para poder observar si se ha producido un desplazamiento temporal en las señales. En la siguiente imagen, simulación de inclusión cilíndrica, se puede observar claramente la forma circular en la zona central. En las líneas periféricas, líneas 0 a 30 y 110 a 128 aproximadamente, se pueden observar artefactos. En este caso se aprecia lo que se conoce como *side lobe effect* [16], produciendo una de imagen espejo de la imagen real.

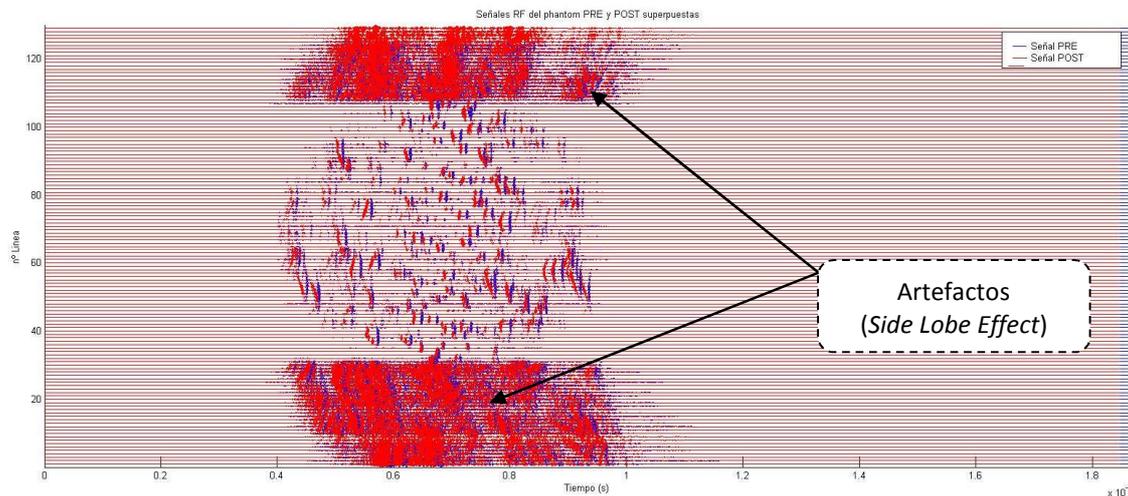


Ilustración 17: Líneas de radiofrecuencia de pre- y postcompresión.

Al hacer un aumento de las líneas central, las que contienen información de la inclusión cilíndrica, se puede observar como para las mismas líneas de radiofrecuencia se reproducen los mismos patrones en las imágenes de pre- y postcompresión, únicamente desplazados.

Este desplazamiento entre patrones es el que se deberá reconocer y cuantificar en los siguientes pasos del tratamiento de la señal.

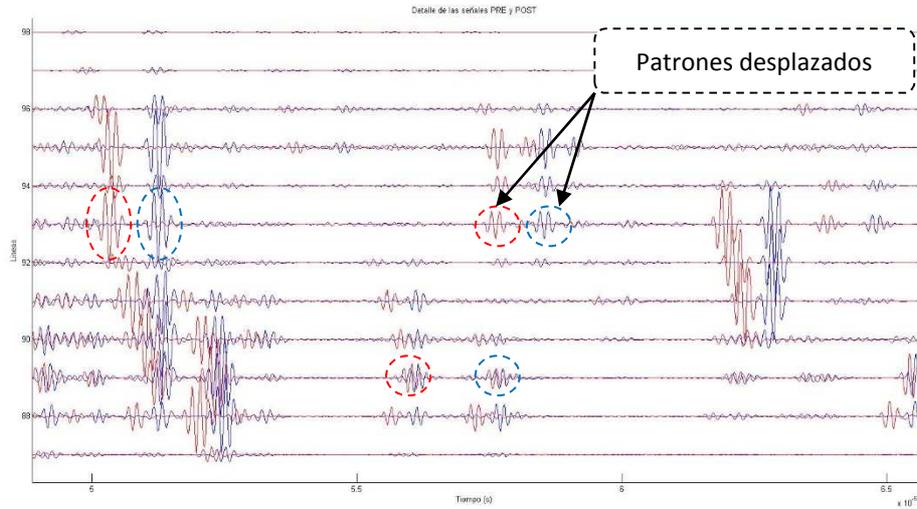


Ilustración 18: Detalle de patrones de líneas de radiofrecuencia pre- y postcompresión.

Finalmente, se utiliza una representación de la envolvente con compresión logarítmica en escala de grises para simular una imagen elastográfica.

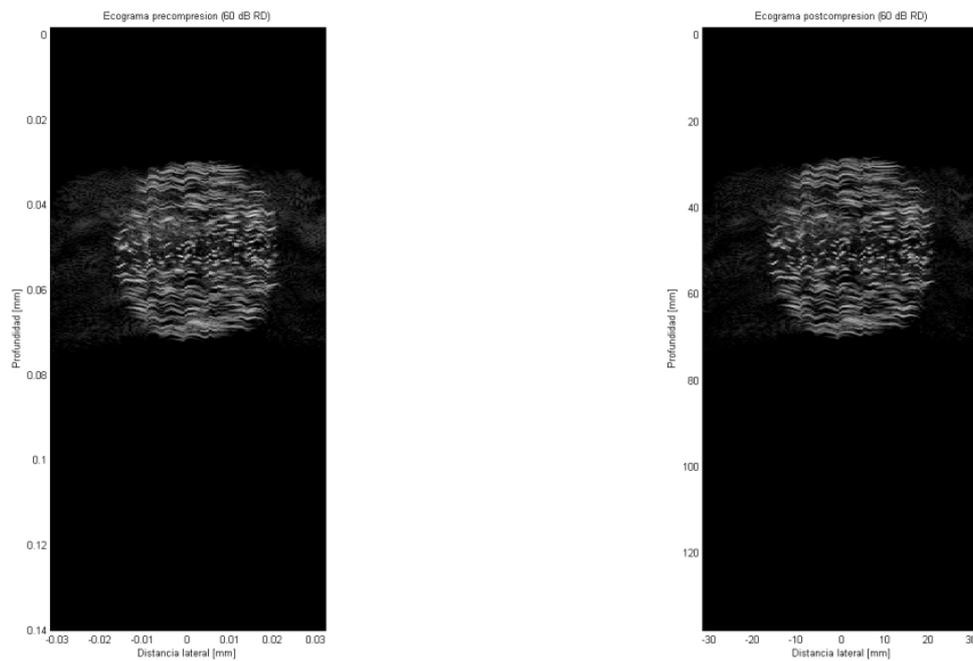


Ilustración 19: Comparación de ecogramas pre- y postcompresión.

Sin un procesado de señal, en este caso, no es apreciable la deformación de la inclusión cilíndrica. Este paso se realiza con el siguiente *matscript* `elab_elastograma.m`.

Durante la comprobación del programa se han obtenido sonogramas con imágenes aparentemente negras, con algún punto blanco. Esto ha ocurrido cuando se han utilizado mapas de amplitudes con ruido aleatorio para obtener buenos dispersores por todo el volumen del phantom. Los picos de amplitud corresponden a reflexiones cercanas a la superficie, debidas a dispersores de gran amplitud.

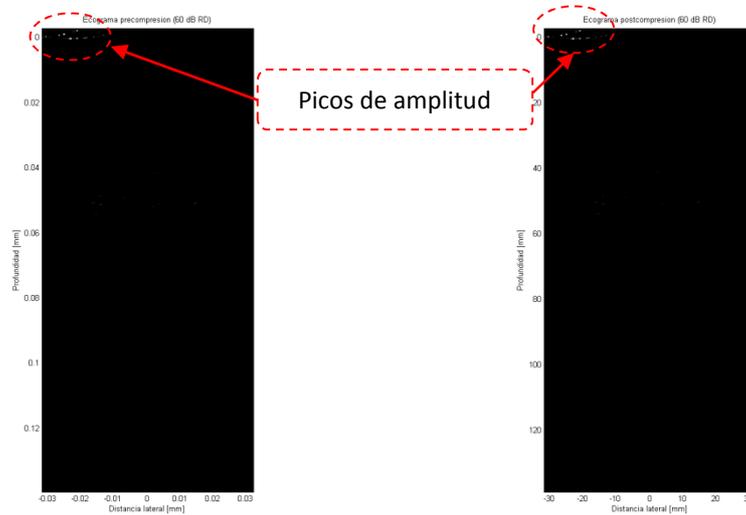


Ilustración 20: Sonograma con picos de amplitud muy pronunciada.

Aun obteniendo un sonograma que aparentemente no parece válido, si se observa la información de las líneas de radiofrecuencia, se puede observar que existen reflexiones a lo largo de toda la profundidad del phantom. Además se pueden reconocer fácilmente los ecos desplazados de una señal a otra (en rojo la señal postcompresión).

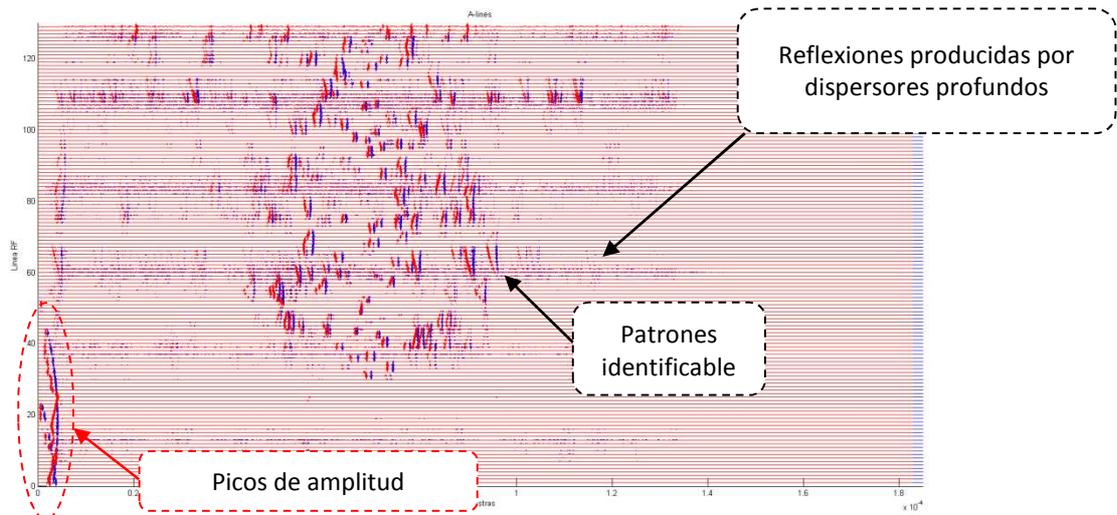


Ilustración 21: Líneas de radiofrecuencia de pre- y postcompresión.

Si se producen reflexiones cercanas y de gran magnitud en la superficie, las señales de las reflexiones que siguen en profundidad son pobres o casi inexistentes. La imagen anterior corresponde con los sonograma con los picos de amplitud mostrados anteriormente.

Una de las partes más difíciles del trabajado ha sido encontrar una buena relación de dispersores para obtener un buen sonograma. Como norma, si se utiliza mayor cantidad de dispersores, menor debe ser la amplitud de estos. Para las simulaciones se han utilizado entre 5.000 y 10.000 dispersores.

5.3 VALIDACIÓN DE ELAB_ELASTOGRAMA

Una vez obtenido las líneas de radiofrecuencia, se procede a realizar el cálculo del elastograma. El cálculo se realiza línea por línea, componiendo una matriz para su representación en dos dimensiones, que será el elastograma.

Esta parte es la más crítica, ya que hasta aquí los datos obtenidos hasta este punto deben ser de buena calidad.

En las siguientes imágenes se pueden observar los elastogramas obtenidos. En la imagen de la izquierda, se muestran en escala de color la magnitud del desplazamiento. En la imagen se pueden observar puntos de una amplitud mucho mayor, que llamaremos espúrio.

Se define como espúrios aquellos valores anómalos que destacan sobre los valores cercanos. Estos valores normalmente se producen cuando la función de correlación cruzada da un máximo para la comparación de dos patrones cuando realmente no lo es. Este error en la detección del desplazamiento da como resultado un pico, bien dando un valor menor.

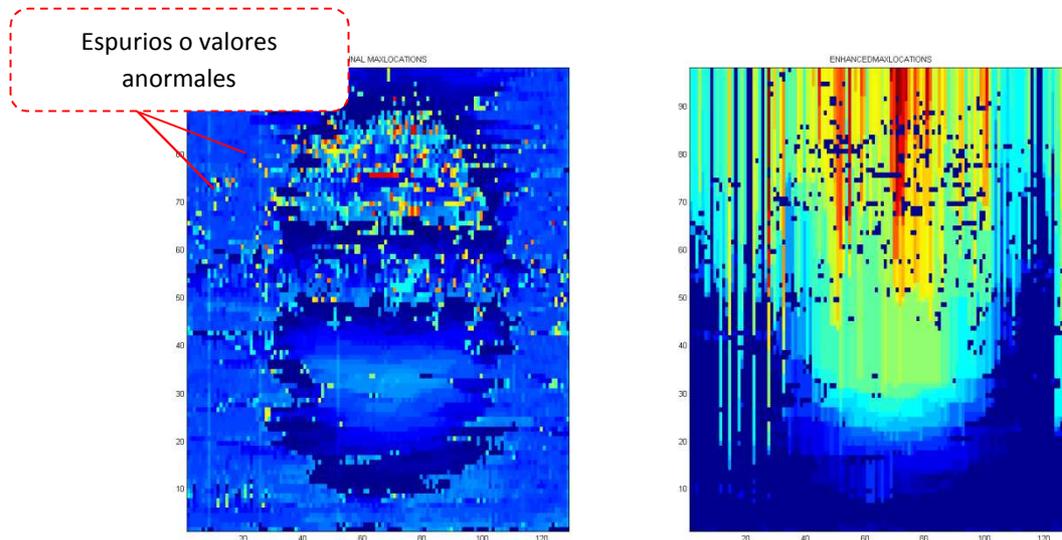


Ilustración 22: Elastograma bidimensional (izda. sin filtrar espúrios, dcha. sin espúrrereos).

Figura: Elastograma bidimensional

Durante las pruebas se han utilizado diferentes técnicas de enventanado (*boxcar windowing*, *hanning*, *hanning* y triangular). La principal dificultad es determinar el ancho de las ventanas (*kernels*) para obtener buenos resultados:

1. Un tamaño de *kernel* excesivamente pequeño hará que para grandes desplazamientos las zonas que se deban corresponder ya no estén en la ventana y por tanto la función de correlación cruzada de ambos *kernels* dé como resultado un espúrios.
2. Un tamaño de *kernel* demasiado grande provocará falsos máximos en zonas periféricas.

Los espúrios periféricos se pueden atenuar volviendo a utilizar un enventanado para enfatizar la zona central.

En la imagen de la derecha se ha probado implementar una corrección de la detección del desplazamiento. Partiendo de la premisa de que si una señal que se comprime, los máximos detectados en el desplazamiento deberán ir aumentando progresivamente y en ningún caso retroceder. Así, se ha implementado que si un máximo solamente podrá ser mayor o igual que el máximo anterior, en ningún caso menor. Si fuese menor, el nuevo máximo será igual al último máximo correcto detectado. Esto ha mejorado por una parte los resultados, pero por

otra también los ha empeorado, ya que si se detecta un máximo muy por encima del valor real, este máximo se mantendrá, lo cual explica las líneas verticales de un mismo color en la imagen de la derecha.

En ambas imágenes se puede observar una zona central de diferente dureza que la zona periférica. Esta es una representación cualitativa de la dureza del *phantom*, que corresponde con el mapa elástico del *phantom*. Así, se ha obtenido a partir de dos imágenes sonográficas, precompresión y postcompresión, una imagen cualitativa en dos dimensiones de la elasticidad del *phantom*.

5.4 CONCLUSIONES

Este trabajo ha servido para desarrollar un entorno de trabajo numérico completo para simular la obtención de imágenes elastográficas y aplicar los conocimientos adquiridos durante los estudios, tanto de métodos numéricos de simulación como de ultrasonidos.

La primera fase de documentación ha sido relativamente dificultosa, dada la amplitud y complejidad de las técnicas elastográficas y ya que la mayoría de los artículos consultados se centran en la mejora de las técnicas elastográficas (correcciones de desplazamiento lateral, sistemas multiresolución, técnicas de mejora de la integridad de señal o identificación de patrones).

Para la parte de la creación de un *phantom* han sido necesarias varias pruebas y modificaciones de los ejemplos encontrados [12], hasta llegar a un modelo que permite la carga de una geometría de forma rápida y que permite aplicar una deformación. Otra de las dificultades encontradas ha sido traspasar los resultados del modelo numérico FEM a un modelo que sirviera como datos de entrada para el simulador de ultrasonidos FIELD II.

El programa FIELD II se ha mostrado como una herramienta muy potente para la simulación de ultrasonidos y relativamente sencilla de utilizar. Una vez se han definido correctamente el modelo de entrada y los parámetros de simulación, se han obtenidos imágenes ecográficas con relativa sencillez. En algunos casos, aparentemente la simulación no ha producido los resultados deseados, pero esto es debido a que existían picos de amplitud que enmascaraban las señales ecográficas producidas por los dispersores de las inclusiones (*Ilustración 20: Sonograma con picos de amplitud muy pronunciada.*). Se sospecha que esto es debido a la naturaleza aleatoria en la magnitud o intensidad de los dispersores. Esto no se ha producido en todas las simulaciones.

La obtención del elastograma ha sido relativamente sencilla, dada la experiencia del trabajo anterior [10]. En este paso, la tarea más ardua ha sido encontrar el valor correcto de los parámetros de postprocesado como pueden ser los tamaños de ventana o *kernel*, los desplazamientos de ventana o filtrados para minimizar o descartar espurios.

Dado que la simulación de la propagación de ultrasonidos mediante FIELD II puede durar varios minutos (dependiendo de la cantidad de dispersores), ha venido bien guardar las señales

resultantes en archivos *.mat*. Esto ha permitido cargar las señales rápidamente y disponer de una copia de seguridad para probar diferentes parámetros de postprocesado.

Mientras que los programas de creación del modelo o *phantom* mediante FEM y simulación de ondas mediante FIELD II han sufrido pocas modificaciones desde su creación no ha sido así con el programa de postprocesado. Para ello se han probado varias técnicas, como la utilización del factor de correlación en las primeras versiones. Se ha descartado el factor de correlación, ya que produce gran cantidad de espurios (muchos patrones de señal son parecidos) y además no da el desplazamiento temporal entre dos señales.

En cuanto al inventariado utilizado y el desplazamiento sobre las señales, se puede optimizar la técnica en dos sentidos:

1. Optimización de la velocidad o reducción del coste computacional.
2. Optimización de búsqueda de patrones y desplazamiento.

Es aquí donde muchos autores realizan sus investigaciones y elaboran trabajos sobre ello dada la dificultad y la variedad de técnicas disponibles.

Aunque el programa final de postprocesado disponga de varias ventanas de visualización, muchas de ellas pueden ser omitidas o ignoradas ya que su única función es controlar y visualizar las señales intermedias resultantes de las operaciones realizadas, como son los correlogramas, que nos indican la calidad o la ubicación de la información útil de las señales de radiofrecuencia.

Referente al uso de MATLAB, podemos decir que ha sido una herramienta poderosa, por un lado, por la gran cantidad de documentación de modelos numéricos FEM utilizados para la creación del phantom y por otro, por la librería o programa FIELD II, que ha permitido la simulación de la propagación de ultrasonidos.

Para futuros trabajos se propone:

1. Revisar el modelo FEM e introducir correcciones ya que para deformaciones extremas (80% sobre el tamaño total) parece arrojar resultados numéricamente incorrectos.
2. Revisar o mejorar la forma para introducir geometrías en los modelos, ya que actualmente se trabaja con un archivo *.bmp* en escala de grises. Revisar cómo se almacena los datos y cuál es la mejor forma de trabajar con estos archivos si procede.

3. Determinar cuál es la cantidad óptima de dispersores en función del volumen de un modelo FEM. Pueden variar de entre 2.000 a 50.000 para un mismo modelo. Determinar la relación entre cantidad de dispersores e intensidad de los mismos y cómo afectan a la simulación con FIELD II y a los resultados obtenidos.
4. Mejorar el método de detección de desplazamiento utilizando otras funciones o algoritmos.
5. Mejorar el sistema de enventanado y de seguimiento del desplazamiento.

ANEXOS

El objetivo del presente anexo es ofrecer una visión de los parámetros de entrada de los diferentes programas o scripts escritos en MATLAB.

Las funciones o scripts son las siguientes:

1. `elab_init`. Genera el espacio de trabajo para guardar la simulación, crea archivo con los datos de la simulación y ejecuta el programa FIELD II.
2. `elab_phantom(N, compresión)`. Genera el modelo mediante FEM para la simulación.
3. `elab_sonograma()`. Carga el modelo generado en `elab_phantom` y procede a la simulación mediante el programa FIELD II. Guarda cada línea de radiofrecuencia obtenida en un archivo `.mat`.
4. `elab_elastograma()`. Carga las líneas de radiofrecuencia generadas en el paso anterior desde el archivo `.mat` y procede al condicionamiento de las señales y al postprocesado para obtener el elastograma.

A.1 FUNCIÓN ELAB_PHANTOM

La función `elab_phantom` se encarga de obtener el modelo para realizar el cálculo del sonograma. Los datos de salida son dos matrices (coordenadas de los dispersores de precompresión y postcompresión) y un vector con las amplitudes de los dispersores.

Llamada a la función:

```
elab_phantom(N, compresion)
```

Parámetros de entrada:

N: Cantidad de dispersores

compresion: Compresión aplicada en

Parámetros o archivo de salida:

Archivo `phantom_data.mat`

`Precompresion`: Coordenadas de los dispersores de precompresión.

`Postcompresion`: Coordenadas de los dispersores de post-compresion

`amp`: Amplitudes de los dispersores, es igual para el modelo de

Una vez creados y guardados los *phantoms* podemos calcular los sonogramas correspondientes. En la función `elab_sonograma` hacemos uso de la herramienta FIELD II para realizar la simulación. Éste es el paso que más tiempo y recursos de nuestro ordenador consume. Describiremos detalladamente los pasos de la función en la sección dedicada a esta.

A.2 FUNCIONAMIENTO DEL PROGRAMA

Con este programa crearemos dos *phantoms* de N dispersores, uno sin deformar y otro phantom comprimido según la compresion indicada en milímetros por el parámetro `compresion`. Las dimensiones del phantom se pueden modificar dentro del archivo o función.

Los datos de entrada son dos imágenes en formato BMP (Mapa de Bits de Windows) que contendrán por un lado información sobre la elasticidad (archivo `mapa_elastico.bmp`) del modelo y por otro lado, información sobre la amplitud de los dispersores (archivo

mapa_amplitudes.bmp). El programa devuelve tres variables relacionadas con los dispersores, posiciones y amplitud para cada modelo.

Deberemos crear dos archivos con extensión BMP. El primer archivo deberá tener el nombre de mapa_elastico.bmp que contendrá las diferentes elasticidades de nuestro modelo. Con un color claro definiremos un material elástico o blando y con un color oscuro definiremos un material poco elástico o duro. No se necesita ninguna herramienta de dibujo profesional, basta crear un archivo en escala de grises en Microsoft Paint u otra herramienta similar. Recomendamos que se edite alguno de los archivos que se entregan con este trabajo (formato adecuado 8 bits, escala de grises).

De la misma manera crearemos otro archivo con las amplitudes de los dispersores, con el nombre de mapa_amplitudes.bmp. Podemos utilizar como recomendación general las siguientes reglas de diseño:

1. Si nuestro modelo tiene muchos dispersores (>10.000) deberemos reducir las amplitudes de forma general y preferiblemente más en zonas cercanas al transductor.
2. Como referencia, para un modelo de 750 cm³ para realizar pruebas recomendamos unos 10.000 dispersores, y para simulaciones de 50.000 a 100.000 dispersores.

Una vez creadas ambas imágenes procederemos a cargarlas y a comprobar que las dimensiones sean correctas. Cargamos las imágenes con la función `bmpread` que posiblemente deberemos incluir en nuestro directorio de trabajo. Una alternativa a esta función podría ser la función `imread`.

A continuación pondremos una referencia espacial a nuestros mapas de datos con las variables `dx`, `dy` y `dz`. Para realizar el cálculo por FEM necesitamos pasar nuestras variables de posición y compresión a variables adimensionales.

Con la variable `markerSize` modificaremos las representaciones de los plots para representar los dibujos. Conviene probar diferentes valores hasta encontrar el que más se ajuste a nuestras necesidades. Esta variable no tiene importancia ya que no modificará nuestros resultados, únicamente afectará a la visualización.

A continuación se realiza la compresión mediante la creación de un modelo FEM y se desplazan todas las posiciones de dispersores del *phantom* de postcompresión.

Finalmente se guardan los datos y se escribe la información necesaria en el archivo info.txt. También se representan con diferentes *plots* los modelos calculados.

B.1 FUNCIÓN ELAB_SONOGRAMA

Con la función `elab_sonograma.m` se obtienen todas las líneas de radiofrecuencia (precompresión y postcompresión) y es donde interviene el programa FIELD II. Cargamos los parámetros de los *phantoms* y realizamos el cálculo de la respuesta al impulso espacial. En esta función describimos la apertura de emisión y de recepción.

Llamada a la función:

```
elab_sonograma()
```

Parámetros o archivo de salida:

Archivo `sonogram_data.mat`

`rfAllLinesAligned_pre`: Todas las líneas simuladas y alineadas temporalmente del phantom de precompresión.

`rfAllLinesAligned_post`: Todas las líneas simuladas y alineadas temporalmente del *phantom* de postcompresión.

Archivo `sonogram_parameters.mat`

`fs`: Frecuencia de muestreo de la señal.

`f0`: Frecuencia central del transductor.

`c`: Velocidad del sonido en el medio.

`no_lines`: Cantidad de líneas obtenidas.

Archivo `sonogram_data.mat`

`rfAllLinesAligned_pre`: Líneas de radiofrecuencia alineadas en origen ($t = 0$ s).

`rfAllLinesAligned_post`: Líneas de radiofrecuencia alineadas en origen ($t = 0$ s).

Nota: Se deben alinear las líneas en origen de tiempo ya que para cada línea de radiofrecuencia y por características de la respuesta del impulso temporal la primera muestra de la señal tiene un origen diferente. Ver ejemplos en Field II.

En este programa realmente no necesitamos realizar modificaciones a menos que deseemos modificar los parámetros del transductor como: cantidad de elementos, altura de elemento, ancho de elemento, respuesta temporal de la apertura de emisión, respuesta temporal de la apertura de recepción. Se recomienda consultar el manual del programa FIELD II.

B.2 FUNCIONAMIENTO DEL PROGRAMA

En la primera parte del programa definimos todas las variables relacionadas con el diseño del transductor. Para entender esta parte recomendamos consultar algún manual sobre el diseño de transductores o la ayuda del programa FIELD II.

A continuación definimos los siguientes elementos necesarios para realizar la simulación, como son las aperturas de emisión y recepción así como sus respectivas respuestas temporales. A continuación se inicializan las diferentes variables necesarias para el cálculo y también se reservan espacios en la memoria para guardar los datos simulados. Todas las líneas calculadas se almacenan en la carpeta `rf_data`. El cálculo de líneas se puede reanudar en cualquier punto ya que el programa comprueba cuál es la última línea calculada.

Una vez calculadas todas las líneas, es necesario que alineemos temporalmente cada línea, ya que los resultados devueltos por FIELD II empiezan en la primera muestra de la respuesta calculada. Para ello componemos una matriz con todas las líneas de radiofrecuencia calculadas y alineamos las líneas según los tiempos.

Como paso previo a todo el procesado, el programa ajusta las señales para que tengan la misma longitud y podamos trabajar sin problemas tamaños de vectores.

El programa trabaja inventanando cada línea de precompresión y postcompresión. Se aplica la función de correlación cruzada (función `xcorr(x, y)`) ventana a ventana y localizamos los máximos. Estos máximos nos dan una posible idea del desplazamiento temporal entre las dos señales.

Una punto importante es la reducción del ruido debido a valores extremos de la señal que hemos obtenido a partir de la función de correlación cruzada. Esto lo conseguimos con ventanas dinámicas que se van ajustando a los resultados obtenidos. La ventana está

compuesta por vector de ceros de longitud variable, una ventana *Hanning* de tamaño fijo y otro vector de ceros de longitud variable. Las funciones de estas ventanas son las siguientes:

- i. En primer lugar descartamos los valores negativos o menores que el máximo actual, ya que supondrían un adelantamiento temporal, que en este caso es imposible a no ser que invirtamos las señales.
- ii. La ventana de *Hanning* se centra en el máximo que hemos obtenido anteriormente. Con esto damos más importancia a valores cercanos al máximo obtenido anteriormente y reducimos la amplitud de posibles picos muy alejados que podrían introducir ruido.

Aplicando estas ventanas deslizantes obtenemos una notable mejora de la detección del desplazamiento y no hace necesarios que debamos fijar un umbral en el nivel de correlación. Con todas estas mejoras obtenemos mejores resultados que los obtenidos en el trabajo *Obtención de imágenes elastográficas en una dimensión*.

Finalmente, presentamos los resultados de diferentes formas.

C.1 FUNCIÓN ELAB_ELASTOGRAMA

Con la función `elab_elab.m` elastograma se obtienen los elastogramas a partir de los datos del sonograma creado en el paso anterior.

Llamada a la función:

```
elab_elastograma()
```

Parámetros o archivo de salida:

Archivo `elastogramInfo.mat`

`corrAmounts`: Indican los valores de los máximos de aplicar la función de correlación cruzada (FCC).

`maxLocations`: Guarda la ubicación de los máximos del pico de la función de correlación cruzada, tanto valores negativos (menores que la mitad de la longitud de ventana), que indican retrasos (no válidos) o valores positivos.

`enhancedLocations`: Se descartan aquellos valores negativos (que indican retrasos en la señal, consecuente expansión) o anormalmente altos.

Esta parte es la que realmente está abierta a futuras mejoras. Ya que la cantidad de métodos, funciones y algoritmos a aplicar dependen del autor. Se pueden encontrar algunas líneas comentadas en `matscript` (`%% Comentario`) que no se han borrado ya que podrían ser de utilidad o servir como idea.

C.2 FUNCIONAMIENTO DEL PROGRAMA

En primer lugar se definen las características de las ventanas o kernels con los parámetros `overlapping` y `windowSize`. Para `overlapping` se pueden utilizar tamaños de entre 20 y 80%, como mayor sea este valor, más tiempo durará el postprocesado (la ventana avanzará a menor velocidad y se realizará un mayor número de iteraciones). Para `windowSize` se recomienda un tamaño similar al de la compresión aplicada.

El algoritmo de búsqueda de desplazamientos es la parte más abierta del proyecto, ya que se puede modificar para introducir mejoras en cuanto a rendimiento y a la discriminación de señales.

La idea es disponer de una ventana fija formada por una porción de la señal original o *kernel* precompresión e ir desplazando sobre la porción de señal deformada o *kernel* postcompresión. En cada iteración se calcula mediante la FCC el valor del máximo y su ubicación respecto al centro, que nos indicará el desplazamiento entre las porciones de ambas señales (*kernels* pre- y postcompresión)

Una vez que se han detectado los valores de los desplazamientos se puede proceder a una mejora de los resultados, utilizando un enventanado (Hanning, Hamming, triangular, cuadrado) que favorezca los resultados centrales del resultado de la FCC y descarte resultados anormalmente altos o bajos.

Finalmente se representa la imagen elastográfica completa en una escala de colores para poder comparar con la imagen original de `mapa_elasticidades.bmp`.

Nota: Antes de utilizar el conjunto de programas eLab, por favor, realice estos pasos previos:

1. Establezca la ubicación de trabajo para la ejecución del programa desde MATLAB (*File / Set Path / [Add with subfolders]*).
2. Descargue la versión correcta de FIELD II desde la <http://field-ii.dk/>. La versión probada de FIELD II es *Field_II_ver_3_24_windows_gcc.tar*
3. Instale las librerías de estadística de MATLAB para tener acceso a las funciones de `xcorr` y similares.
4. Instale los archivos de lectura de imágenes como `bmpread` o `imread` y adapte el programa a las nuevas funciones.

BIBLIOGRAFÍA

- [1] C. I. P. H. Y. Y. a. L. X. J. Ophir, "Elastography: Imaging the Elastic Properties of Soft Tissues with Ultrasounds.," *J Med Ultrasonics*, pp. 155-171, 2002.
- [2] T. Shiina, "JSUM Ultrasound Elastography Practice Guidelines," *JSUM*, 2013.
- [3] "Accuvix A30," 18 February 2016. [Online]. Available: <http://www.samsungmedison.com/ultrasound/general-imaging/accuvix-a30/>.
- [4] "Aplio 500," 18 February 2016. [Online]. Available: <http://www.toshiba-medical.eu/eu/product-solutions/diagnostic-ultrasound/aplio-500-overview/aplio-500-advanced-applications/>.
- [5] J. C. I. P. H. Y. Y. a. L. X. Ophir, "Elastography: a method for imaging the elasticity of biological tissues.," *Ultrasonic Imaging, Vol 13 (2)*, pp. 111-134, 1991.
- [6] T. D. M. F. M. T. J.-L. Gennisson, "Ultrasound elastography: Principles and techniques," *Journal de Radiologie Diagnostique et Interventionnelle, Volume 94, Issue 5*, pp. 504-513, May 2013.
- [7] L. Gao, "Imaging of the elastic properties of tissues, a review.," *Ultrasound Med. Biol. 22,* pp. 959-977, 1996.
- [8] N. M. Tole, "Chapter 3: Interaction of ultrasound with matter," in *Basic Physics of Ultrasonographic Imaging*, Nairobi, World Health Organization, 2005, pp. 21-32.
- [9] H. B. C. M. L. S. Kim GW, "X-ray elastography: a feasibility study," *Conf Proceedings IEEE Eng Med Biol Soc*, 2009.
- [10] D. S. Freese, "Obtención de imágenes elastográficas en una dimensión," Escuela Politécnica de Gandía. Departamento de Física Aplicada, 2010.
- [11] J. Raymod A. Serway y John W. Jewett, *Física para Ciencias e Ingeniería, Volumen 1*, 7ª edición, Cengage Learning .
- [12] A. Ferreira, *MATLAB Codes for Finite Element Analysis. Solids and structures. Chapter 2*, Springer, 2009.

- [13] E. O. J. Konofagou, "A new elastographic method for estimation and imaging of lateral displacements, lateral strains, corrected axial strains and poisson's ratios in tissues," *Ultrasound in Medicine & Biology*, pp. Volume 24, Issue 8, October 1998, Pages 1183–1199, 1998.
- [14] J. Jensen, "A new approach to calculating spatial impulse responses," 1997.
- [15] J. Jensen, Users guide for the FIELD II program, 2011.
- [16] "RSNA Radiographics," 14 03 2016. [Online]. Available: <http://pubs.rsna.org/doi/full/10.1148/rg.294085199>. [Accessed 14 03 2016].
- [17] K. F. S. R. O. J. Srinivasan S, "Analysis of an adaptive strain estimation technique in elastography.," *Ultrason Imaging*, pp. 109-118., 2002 April.
- [18] K. T. A., "Tissue property measurement - modulus, stiffness and scale.," *Ultrasonics Technical University of Delft, The Netherlands*, 2001.
- [19] C. D. G. y. J. A. H. A V PATIL, " 3D prostate elastography: algorithm, simulations and experiments," *Biomedical Engineering, University of Virginia, Charlottesville*, no. VA 22903, USA, May 2007.
- [20] R. S. A. RONALD O. BUDE, "An easily, made low-cost, tissue-like ultrasound," *J Clin Ultrasound* 23:271-273, , May 1995.
- [21] L. PEREIRA NEVES, "Otimização do algoritmo de block matching aplicado a estudos elastográficos.," *Universidade de Sao Paulo. FFCLRP Departamento de física e matemática. Riberão Preto*, September 2007.
- [22] V. V. MAHAJAN., "Ultrasonic elastography measurements of the mechanical properties," *Georgia Institute of Techonology.*, no. August 2005.
- [23] O. JONATHAN, C. IGNACIO and P. HARRI., "Method and apparatus for elastographic measurement and imaging.," *United states patent. Patent number 5.923.870.* , March 1994.
- [24] J. H. TIMOTHY, B. MEHMET, M. F. INSANA and A. K. THOMAS., "Phantom materials for elastography. IEEE Transactions on ultrasonics, ferroelectrics, and frequency control.,"

IEEE Transactions on ultrasonics, ferroelectrics, and frequency control., Vols. 44, no 6,
November 1995.

- [25] I. M. MICHAEL., "Elastography imaging modalities for characterizing properties of tissue.,"
United States Patent. Patent number 2.257.244 B2. , no. August 2007 .