

3  
6  
10

MEDIDA IN SITU DE LA PÉRDIDA ACÚSTICA POR REFLEXIÓN DE BARRERAS ANTIRRUÍDO POR EL MÉTODO "ADRIENNE"

4 de Junio de 1999

E.T.S. DE INGENIERÍA (I.C.A.I.)

INGENIERO INDUSTRIAL  
ESPECIALIDAD MECÁNICA



MEDIDA IN SITU DE LA PÉRDIDA ACÚSTICA POR REFLEXIÓN DE BARRERAS ANTIRRUÍDO POR EL MÉTODO "ADRIENNE"

Autor: Adolfo Corral Quintana

4 de Junio de 1999

ESTE PROYECTO CONTIENE LOS SIGUIENTES DOCUMENTOS:

DOCUMENTO n°1, MEMORIA.

1.1. Memoria descriptiva.	pág. 2 a 82	81 páginas.
1.2. Cálculos.	pág. 83 a 117	35 páginas.
1.3. Anejos.	pág. 118 a 210	93 páginas.

DOCUMENTO n°2, PLANOS.

2. Planos.	pág. 1	
------------	--------	--

DOCUMENTO n°3, PLIEGO DE CONDICIONES.

3.1. Generales y económicas.	pág. 2 a 3	2 páginas.
3.2. Técnicas y particulares.	pág. 4 a 23	20 páginas.

DOCUMENTO n°4, PRESUPUESTO.

4.1. Introducción.	pág. 2	
4.2. Costes directos.	pág. 3	
4.3. Costes indirectos	pág. 4	
4.4. Presupuesto general.	pág. 5	

# MEDIDA IN SITU DE LA PÉRDIDA ACÚSTICA POR REFLEXIÓN DE BARRERAS ANTIRRUIDO POR EL MÉTODO "ADRIENNE".

**Autor: Corral Quintana, Adolfo.**

Director: Cobo Parra, Pedro.

Entidad colaboradora: Instituto de Acústica, perteneciente al C.S.I.C..

## RESUMEN DEL PROYECTO.

Las barreras acústicas se están convirtiendo en uno de los dispositivos de control del ruido más populares. Su misión es la de obstruir la transmisión directa entre la fuente de ruido (tráfico rodado, trenes,...) y el receptor (la población).

Fruto de la expansión de este tipo de dispositivos, surge la necesidad de poder **caracterizarlos acústicamente**, con objeto de poder comparar unos con otros. De este modo, las empresas que fabrican materiales para barreras antirruído podrán dar las características de sus distintos productos. También se podrá comprobar si las propiedades acústicas de una determinada barrera se han degradado con el tiempo.

Existe un Comité de Normalización Europeo, **CEN/TC226**, que se ocupa de este tema. Dicho Comité se encarga, esencialmente, de redactar la norma **EN 1793**, cuyo título es:

*“Dispositivos reductores de ruido de tráfico en carreteras –  
Método de ensayo para determinar su comportamiento acústico”*

A la hora de redactar estas normas surgieron problemas con los métodos existentes, debido a la inexistencia de métodos apropiados de medida **in situ** (en el mismo lugar de la instalación: carreteras, vías de tren..). Ante la necesidad de resolver estos y otros problemas para redactar la norma **EN 1793** surgió en Enero de 1995 el **Proyecto Europeo “ADRIENNE”**, del programa de “Normas, Medidas y Ensayos” (SMT).

El proyecto Adrienne dio como resultado un **método** válido para medir in situ características intrínsecas de barreras antirruído. El proyecto Adrienne finalizó en febrero de

1998. Se ha redactado ya un borrador de la norma **EN-1793**, no una versión definitiva, ya que todavía existen problemas a la hora de caracterizar barreras no planas.

Este proyecto de fin de carrera implementa el método “Adrienne”, para obtener la pérdida acústica por reflexión, en un **programa informático** que realiza rápidamente los complejos cálculos matemáticos asociados al método y que es de fácil e intuitivo manejo para el usuario. De este modo, el usuario podrá obtener la pérdida acústica por reflexión de una determinada barrera antirruído alimentando al programa con las medidas realizadas in situ (en el mismo lugar de instalación de la barrera) y con los parámetros geométricos asociados a dichas medidas.

La **pérdida acústica por reflexión (RL)** se define como *“la parte de la energía sonora que se refleja en el mismo medio cuando una onda sonora incide en una frontera”*. Es función de la frecuencia de la señal acústica y del ángulo con el que inciden las ondas acústicas en la barrera. La **RL** caracteriza las propiedades absorbentes del dispositivo antirruído.

Básicamente, el **método Adrienne** para la obtención de la pérdida acústica por reflexión, consiste en emitir una señal de presión acústica **M.L.S.** (gracias a la cual se obtiene la respuesta impulsiva de la barrera) a través de un altavoz. Un micrófono unido solidariamente al altavoz recoge tanto el pulso de presión directo como el reflejado en la barrera. Por un posterior procesado de las señales recogidas por el micrófono se obtienen **espectros de energía**, en bandas de tercio de octava, asociados al eco directo y al reflejado. Dividiendo el espectro de energía reflejado entre el directo se obtiene la pérdida por reflexión para un determinado ángulo de incidencia. Este proceso se repite para 9 ángulos de incidencia distintos que simulan el típico ruido de tráfico. Finalmente, se dan promedios, para distintos ángulos de incidencia, de la pérdida por reflexión, **RL**, en 18 bandas de tercio de octava, entre 100 y 5000 Hz. A partir de estos promedios, se podrá expresar la absorción acústica de la barrera con un único número, **DL $\alpha$** , siguiendo los pasos que se indiquen en las futuras normas **EN 1793-1** y **EN 1793-3**.

Para realizar dicho sistema informático se ha escogido el programa **MATLAB** por ser muy potente en comandos matemáticos y por estar muy implantado en el entorno científico

europeo. La comunicación con el sistema es muy intuitiva al realizarse a través de interfaces gráficas.

En este proyecto de fin de carrera se ha realizado un **avance**, con respecto al método Adrienne, en cuanto a la **extracción del pulso de presión reflejado** a partir de una señal que contiene el eco directo, el reflejado y otros ecos procedentes de reflexiones y difracciones no deseadas. Esta extracción de señal se realiza para obtener la energía que se refleja en la barrera. Tal y como está definido, el método Adrienne presenta problemas en este aspecto. Un exhaustivo estudio de P.Cobo (1999), que se basa en considerar los parámetros geométricos asociados a la medida, ha sido tenido en cuenta en el programa, de modo que se optimiza el proceso de extracción de la señal reflejada.

El **coste total** de realización del proyecto es de 3.227.700 pesetas. En este presupuesto se han considerado las horas de trabajo del proyectista y la amortización de los distintos sistemas usados para su realización.

A handwritten signature in black ink, appearing to read 'Alonso', with a long horizontal stroke extending to the right.

**E.T.S. DE INGENIERÍA (I.C.A.I.)**

**INGENIERO INDUSTRIAL  
ESPECIALIDAD MECÁNICA**



**MEDIDA IN SITU DE LA PÉRDIDA ACÚSTICA POR  
REFLEXIÓN DE BARRERAS ANTIRRUIDO POR EL  
MÉTODO "ADRIENNE"**

**Autor: Adolfo Corral Quintana**

*4 de Junio de 1999*

M E M O R I A

## **1.0. ÍNDICE.**



1.0. ÍNDICE.....1

1.1. MEMORIA DESCRIPTIVA.....2

1.2. CÁLCULOS.....83

1.3. ANEJOS.....118

## **1.1. MEMORIA DESCRIPTIVA.**

## **1.1.0. ÍNDICE.**

<b>1.1.0. ÍNDICE</b> .....	2
<b>1.1.1. OBJETO DEL PROYECTO</b> .....	4
1.1.1.1. INTRODUCCIÓN.....	4
1.1.1.2. EL PROYECTO ADRIENNE.....	7
1.1.1.3. OBJETIVO DE ESTE PROYECTO DE FIN DE CARRERA.....	12
<b>1.1.2. PÉRDIDA ACÚSTICA POR REFLEXIÓN</b> .....	13
1.1.2.1. INTRODUCCIÓN.....	13
1.1.2.2. DEFINICIÓN DE LA PÉRDIDA ACÚSTICA POR REFLEXIÓN.....	14
1.1.2.3. DESARROLLO MATEMÁTICO PARA LA OBTENCIÓN DE LA PÉRDIDA ACÚSTICA POR REFLEXIÓN.....	16
<b>1.1.3. MÉTODO ADRIENNE</b> .....	20
1.1.3.1. INTRODUCCIÓN.....	20
1.1.3.2. EL MÉTODO ADRIENNE.....	21
1.1.3.2.1. TÉCNICA DE SUBSTRACCIÓN DE SEÑAL.....	27
1.1.3.2.2. VENTANA ADRIENNE.....	31
1.1.3.2.3. CORRECCIÓN POR DEBILITAMIENTO DE ONDA.....	36
1.1.3.2.4. MÍNIMA FRECUENCIA FIABLE.....	38
1.1.3.2.5. EL MÉTODO M.L.S. Y EL SISTEMA M.L.S.S.A.....	43
1.1.3.3. CARACTERÍSTICAS DEL MÉTODO ADRIENNE.....	47
<b>1.1.4. ESTRUCTURA DEL PROGRAMA DESARROLLADO</b> .....	53
1.1.4.1. INTRODUCCIÓN.....	53
1.1.4.2. FUNCIONES DE INTRODUCCIÓN DE DATOS.....	55
1.1.4.3. FUNCIONES DE CÁLCULO.....	67
1.1.4.4. FUNCIONES DE SALIDA DE RESULTADOS.....	75
<b>1.1.5. COSTE TOTAL</b> .....	80

1.1.6. REFERENCIAS.....81

## **1.1.1. OBJETO DEL PROYECTO.**

### 1.1.1.1. INTRODUCCIÓN.

Las barreras acústicas se están convirtiendo en uno de los dispositivos de control del ruido más populares. Su misión es la de obstruir la transmisión directa entre la fuente de ruido (tráfico rodado, trenes,...) y el receptor (la población).

Fruto de la expansión de este tipo de dispositivos, surge la necesidad de poder **caracterizarlos acústicamente**, con objeto de poder comparar unos con otros. De este modo, las empresas que fabrican materiales para barreras antirruído podrán dar las características de sus distintos productos. También se podrá comprobar si las propiedades acústicas de una determinada barrera se han degradado con el tiempo.

Existe un Comité de Normalización Europeo, **CEN/TC226**, que se ocupa de este tema. La tarea fundamental del **CEN/TC226** es redactar normas europeas acerca de las características acústicas y no acústicas de los dispositivos de reducción del ruido del tráfico rodado (barreras antirruído, falsos túneles, revestimientos absorbentes,..). El objetivo de este grupo de trabajo es redactar normas que sean capaces de caracterizar dispositivos antirruído, de modo que se pueda determinar si estos productos pueden circular libremente por el Mercado Europeo.

Dicho Comité se encarga, esencialmente, de redactar la norma **EN 1793**, cuyo título es:

*“Dispositivos reductores de ruido de tráfico en carreteras –  
Método de ensayo para determinar su comportamiento acústico”*

Esta norma incluye 5 partes:

**1793-1:** Características intrínsecas – Absorción acústica.

**1793-2:** Características intrínsecas- Aislamiento a ruido aéreo

**1793-3:** Espectro normalizado de ruido de tráfico

**1793-4:** Características extrínsecas - Eficiencia in situ

**1793-5:** Características intrínsecas- Valores in situ de absorción acústica y aislamiento a ruido aéreo.

A la hora de redactar estas normas surgieron problemas con los métodos existentes, especialmente en las partes 1,4 y 5, debido a la inexistencia de métodos apropiados de medida **in situ**. Estos problemas fueron:

- Ningún método de medida del funcionamiento acústico de barreras antirruído había sido diseñado especialmente para su uso específico. Las normas **ISO** disponibles (ISO 354 para absorción sonora e ISO 140 para aislamiento a ruido aéreo) fueron diseñadas para materiales de construcción, por lo que requieren ser caracterizados en laboratorio en condiciones de campo acústico difuso (cámara reverberante). La mayor parte de las barreras antirruído están sometidas a unos ángulos de incidencia específicos, y no a un campo acústico difuso. Las medidas en campo difuso subestiman (sobrestiman) la absorción de productos planos (no planos), mientras que subestiman el aislamiento a ruido aéreo, en los dos casos (*Clairbois, 1998*). La necesidad de reemplazar las citadas normas **ISO** se hacía evidente.
  
- Existe la necesidad de medir las características acústicas de los dispositivos antirruído exactamente en las condiciones en las que han de funcionar. Es decir, era necesario disponer de un método que permitiera medir tanto en laboratorio como **in situ** (carreteras, vías de tren...). Esta necesidad se recoge específicamente en la parte 5 de la norma **EN 1793**. Hasta la fecha, sólo existía un método normalizado en Francia para las medidas in situ (**AFNOR-NF-S31-089**) que usa, como fuente de ruido, una señal explosiva (un disparo). Este método fue desestimado debido a su falta de repetitividad y a que no proporciona resultados fiables por



debajo de 400 Hz.

Ante la necesidad de resolver estos y otros problemas con las partes 1, 4 y 5 de la norma EN 1793 surgió en Enero de 1995 el Proyecto Europeo "ADRIENNE", del programa de "Normas, Medidas y Ensayos" (SMT)

**E.C. MEASUREMENT AND TESTING PROGRAMME.**

**Project MAT1-CT94049 "ADRIENNE".**

**Test Methods for the Acoustic Performance of Road Traffic Noise Reducing  
Devices**

En lo sucesivo, el Proyecto Adrienne.

## **1.1.1.2. EL PROYECTO ADRIENNE.**

### **1.1.2.1. EQUIPO INVESTIGADOR DEL PROYECTO ADRIENNE.**

El equipo investigador del Proyecto Adrienne está formado por 9 socios:

**A-Tech:** Acoustical Technologies, Bruselas, Bélgica (coordinador del proyecto).

**I.B.P.:** Institute für Bauphysik, Stuttgart, Alemania.

**E.N.T.P.E.:** Ecole National des Travaux Publics de l'Etat, Lyon, Francia.

**C.S.T.B.:** Scientifique et Technique du Bâtiment, Grenoble, Francia.

**L.C.P.C.:** Central des Ponts et Chaussées, Nantes, Francia.

**D.I.E.N.C.A.:** Dipartimento di Ingegneria Energetica, Nucleare e del Controllo Ambientale, Bologna, Italia.

**I.A.:** Instituto de Acústica, CSIC, Madrid, España.

**F.I.G.E.:** Forschungsinstitut Geräusche und Erschütterungen, Herzogenrath, Alemania.

**C.E.D.I.A.:** Centre d'Etude et de Developpement en Ingenierie Acoustique, Lieja, Bélgica.

Este proyecto de fin de carrera ha sido realizado en el **Instituto de Acústica**, perteneciente al C.S.I.C., bajo la dirección de **Pedro Cobo**, que forma parte del equipo investigador del proyecto Adrienne.

### **1.1.1.2.2. OBJETIVO DEL PROYECTO ADRIENNE.**

El objetivo fundamental del Proyecto fue poner a punto un método que:

- Fuese relevante para caracterizar barreras antirruído en carreteras. Esto implica que sea capaz de discriminar ángulos específicos de incidencia, en lugar de usar campo difuso (cámara reverberante). Hay que tener en cuenta que la presión acústica del ruido procedente del tráfico incide con unos **ángulos determinados**.
- Fuese capaz de **caracterizar** todos los **dispositivos** antirruído que circulan en el mercado europeo, incluyendo materiales planos y no planos.
- Usara esencialmente los mismos principios y equipamiento para la medida de la absorción y para la del aislamiento a ruido aéreo.
- Se pudiera aplicar tanto en laboratorio como **in situ** (en las carreteras).
- Proporcionara las medidas de absorción acústica y el aislamiento a ruido aéreo en bandas de tercios de octava desde 100 Hz a 5 kHz.
- Expresando los resultados con un único número, como se describe en la norma EN 1793, partes 1 y 2, se tuviera una precisión de  $\pm 2$  dB.

### **1.1.1.2.3. ETAPAS.**

El Proyecto, desarrollado entre Enero de 1995 y Diciembre de 1997, constó de las siguientes etapas:

#### **1.1.1.2.3.1. SELECCIÓN DEL MÉTODO.**

Se consideraron los métodos N.I.T., T.D.S., M.L.S., G.I.T., intensimetría, B.E.M., belga, e ISO CD 10847. Después de un análisis exhaustivo basado en la comparabilidad, representatividad, margen de frecuencias, resolución en frecuencias, inmunidad a ruido de fondo, efecto del tamaño de la muestra, precisión, repetitividad, calibración, facilidad de aplicación, aplicabilidad in situ, y sensibilidad a efectos externos, se llegó a la conclusión de la superioridad del método de las Secuencias Seudoaleatorias de Máxima Longitud (MLS, en lo sucesivo) (*Garai, 1993*) con la técnica de substracción de señal (*Mommertz, 1995*).

#### **1.1.1.2.3.2. DESARROLLO DEL MÉTODO.**

Las señales **MLS** consisten en secuencias binarias periódicas de longitud  $L$  que tienen la singularidad de proporcionar autocorrelación unidad sobre una longitud de señal  $L$ . La ventaja esencial es que sólo se requiere un micrófono para medir la respuesta impulsiva del sistema. Como otros métodos impulsivos, permite separar en la traza la señal reflejada o transmitida en la muestra de otros eventos indeseables, tales como difracción en los bordes o reflexiones en objetos cercanos. Mediante ventaneo se extrae la señal deseada. El tamaño de la ventana limita la resolución en baja frecuencia del método. Dado que uno de los objetivos iniciales era bajar hasta los 100 Hz, la optimización de la ventana temporal resulta una de las cuestiones claves del Proyecto.

#### **1.1.1.2.3.3. APLICACIÓN Y VALIDACIÓN EXPERIMENTAL DEL MÉTODO.**

Se realizaron medidas en muestras planas (vidrio, chapa perforada, lana de roca) y no planas (mezcla de madera y cemento, zig-zag), tanto en laboratorio como

in situ. Las medidas de laboratorio se realizaron en el IBP, DIENCA y ENTPE, con muestras similares y equipos distintos. Para las medidas *in situ* se construyó una barrera de 20 metros de larga y 4 metros de alta, en el CSTB de Grenoble (la barrera Adrienne). A lo largo de Septiembre de 1997, ocho grupos de los participantes del Proyecto llevaron a cabo un "blind test" sobre la barrera Adrienne, con cuatro equipos distintos. La máxima diferencia de resultados entre estos ocho grupos fue de  $\pm 2$  dB para el  $DL\alpha$ , parámetro por el que se determinan las propiedades absorbentes. Los resultados de estas experiencias han sido publicados en los Congresos Nacionales de las Sociedades de Acústica Italiana, Española y Francesa, y en los Congresos Internacionales Euronoise, Internoise e ICA.

#### **1.1.1.2.3.4. ANÁLISIS E INFORME FINAL.**

En Febrero de 1998 se envió el Informe Final del Proyecto Adrienne a la Comisión Europea.

#### **1.1.1.2.3.5. SITUACIÓN EN LA ACTUALIDAD.**

El proyecto Adrienne dio como resultado un método válido para medir características intrínsecas de dispositivos antirruído. El proyecto Adrienne finalizó en febrero de 1998.

Este método es totalmente satisfactorio para caracterizar dispositivos planos, sin embargo, en dispositivos no planos presenta problemas.

Se ha redactado ya un borrador de la norma EN-1793 para los dispositivos planos. El método Adrienne necesita algunas mejoras para caracterizar dispositivos no planos.

Se está intentando extrapolar el uso del método Adrienne para caracterizar acústicamente **pavimentos porosos** que disminuyen el ruido de tráfico. El método Adrienne, en estos casos, es ideal ya que se puede determinar **in situ** (en la misma carretera) si el pavimento ha perdido sus propiedades acústicas. Con el paso del tiempo, el pavimento va perdiendo su porosidad y disminuyendo su absorción acústica.

El método Adrienne está en plena expansión.

### 1.1.1.3. OBJETIVO DE ESTE PROYECTO DE FIN DE CARRERA.

El proyecto tiene como objetivo implementar el método “Adrienne” en un **programa informático** que realice rápidamente los complejos cálculos matemáticos asociados al método y que sea de fácil e intuitivo manejo para el usuario.

De este modo, el usuario podrá obtener la pérdida acústica por reflexión (RL) de una determinada barrera antirruído alimentando al programa con las medidas realizadas en su mismo lugar de instalación

Teniendo en cuenta que este método pasará a ser norma europea para la caracterización acústica de barreras antirruído, éste será el primer programa en España en el que a partir de las señales medidas en la barrera se puedan obtener las características acústicas de la barrera, según la normativa. Es probable que este programa sea usado en el **Instituto de Acústica**, perteneciente al C.S.I.C. (Centro Superior de Investigaciones Científicas).

Puede que el método Adrienne todavía sufra unas pequeñas variaciones para extender su uso a dispositivos no planos. En este caso, bastaría con introducir algunas modificaciones en el programa para adaptarlo. Previendo tal circunstancia, el listado del programa se halla suficientemente comentado para que alguien que no conozca el programa pueda rápidamente conocerlo e introducir los cambios pertinentes.

## **1.1.2. PÉRDIDA ACÚSTICA POR REFLEXIÓN.**



### 1.1.2.1. INTRODUCCIÓN.

El objetivo final del proyecto “Adrienne” es caracterizar las barreras antirruido con dos únicos números:

- **DL $\alpha$** , que es representativo de las características absorbentes de la barrera.
- **DL $r$** , que es representativo de las características aislantes de la barrera.

Estos dos números deberán ser proporcionados por los fabricantes de materiales para barreras antirruido cuando entre en vigor la norma EN 1793. De este modo, se podrán comparar los distintos materiales, proporcionados por los distintos fabricantes, entre sí. Estos productos podrán circular libremente por el Mercado Europeo.

Este proyecto de fin de carrera se centrará en obtener sólo las características absorbentes en barreras antirruido, a partir de unas medidas realizadas en una determinada barrera antirruido. Para ello, se implementa el método Adrienne en un programa informático, de modo que se puede obtener **la pérdida por reflexión** (de la cual se obtienen las propiedades absorbentes) de una forma rápida e intuitiva.

### 1.1.2.2. DEFINICIÓN DE LA PÉRDIDA ACÚSTICA POR REFLEXIÓN.

La **pérdida acústica por reflexión** está estrechamente ligada con las **propiedades absorbentes** de la barrera acústica. A continuación, se presentan dichas relaciones.

La **absorción acústica** es una de las propiedades más importantes de materiales y dispositivos acústicos. Como absorción acústica se entiende *“la parte de la energía acústica incidente que queda absorbida por una superficie o un medio en una frecuencia determinada. Su valor también es función del ángulo con el que incide el sonido” (Piroux, 1966).*

Se expresa habitualmente dando los valores del coeficiente de absorción acústica como función de la frecuencia y del ángulo de incidencia del sonido,  $\alpha(f, \theta)$ . **Garai (1995)** lo define como *“el cociente de la energía acústica absorbida por una determinada superficie entre la energía acústica incidente”*.

Esta cantidad puede medirse directamente o bien puede calcularse a partir de los valores del factor de reflexión de presión acústica  $R_p(f, \theta)$ . Este último se define como el cociente entre la presión acústica reflejada y la presión acústica incidente, multiplicado por un **factor corrector de debilitamiento**<sup>1</sup> de onda por divergencia geométrica. La energía acústica es proporcional a la presión acústica al cuadrado. La relación entre el coeficiente de absorción acústica y el factor de reflexión de presión acústica es:

---

<sup>1</sup> Para los cálculos se asume el modelo de onda esférica para las ondas de presión acústica. En este modelo, la amplitud de la onda decae de forma inversamente proporcional con la distancia avanzada por la onda.

$$\alpha(f, \theta) = 1 - |R_p(f, \theta)|^2 \quad (1.1.2.1)$$

- $R_p$  es una magnitud compleja, pero la información de fase no afecta a la absorción, por lo cual sólo tendremos en cuenta su módulo.

$|R_p(f, \theta)|^2$  es lo que se denomina **pérdida acústica por reflexión, RL**. Se define como *“la parte de la energía sonora que se refleja en el mismo medio cuando una onda sonora incide en una frontera” (Stephens, 1974)*. La **RL** caracteriza las propiedades absorbentes del dispositivo antirruído.

Por tanto, existe una relación estrecha entre la **RL** y la absorción acústica de un dispositivo antirruído.

El **objetivo final** es obtener promedios de la pérdida por reflexión, **RL**, en 18 bandas de tercio de octava, entre 100 y 5000 Hz, para distintos ángulos de incidencia. A partir de estos promedios, se podrá expresar la absorción acústica de la barrera con un único número, **DL $\alpha$** , siguiendo los pasos que se indican en las normas **EN 1793-1** y en la norma **EN 1793-3**.

Estas dos normas, cuya redacción todavía no es definitiva, se presentan en el apartado **"1.3. ANEJOS"**, en la parte final de la memoria.

### 1.1.2.3. DESARROLLO MATEMÁTICO PARA LA OBTENCIÓN DE LA PÉRDIDA ACÚSTICA POR REFLEXIÓN.

Se presenta en este apartado una breve descripción de la **teoría matemática** en la que se fundamenta la obtención de la pérdida acústica por reflexión (RL) a partir de las señales medidas de presión acústica.

La pérdida acústica por reflexión, en el dominio del tiempo,  $r_p(t)$  se puede obtener a partir de la presión incidente  $p_i(t)$  y de la presión reflejada  $p_r(t)$  por la superficie de la muestra ensayada. Las dos presiones son recogidas por un micrófono. La presión reflejada puede escribirse como:

$$p_r(t) = K_r \cdot r_p(t) * p_i(t - \tau_r) \quad (1.1.2.2)$$

donde:

- \* denota convolución
- $K_r$  es un factor corrector del debilitamiento de la onda por divergencia geométrica.

$$K_r = \frac{H - h}{H + h} \quad (1.1.2.3)$$

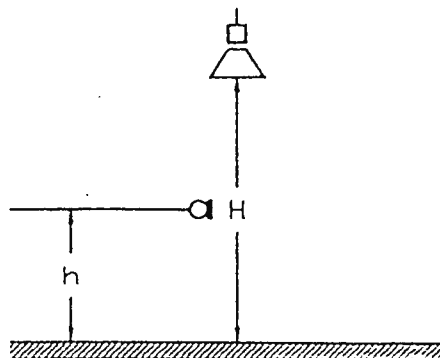


Figura 1.1.2.3.1. Croquis con las distancias relativas entre altavoz y micrófono.

- $\tau_r$  es el retraso temporal de la onda reflejada con respecto a la onda directa.

$$\tau_r = \frac{2 \cdot h}{c}, \quad \text{donde } c \text{ es la velocidad del sonido.} \quad (1.1.2.4.)$$

La ecuación (1.1.2.2) muestra que, sin tener en cuenta el factor  $K_r$ ,  $r_p(t)$  es la respuesta a un impulso del sistema lineal e invariante con el tiempo constituido por la superficie del material bajo ensayo. Haciendo la **transformada de Fourier** de dicha ecuación se obtiene:

$$P_r(f) = K_r \cdot R_p(f) \cdot P_i(f) \cdot e^{-2\pi f \tau_r} \quad (1.1.2.5.)$$

Combinando las ecuaciones (1.1.2.2.) y (1.1.2.5.) se obtiene el coeficiente de absorción acústica como el cociente de dos espectros en el dominio de la frecuencia:

$$\alpha(f) = 1 - \frac{1}{K_r} \cdot \frac{|P_r(f)|^2}{|P_i(f)|^2} \quad (1.1.2.6)$$

Los impulsos de presión incidente y reflejada deben ser extraídos de la señal de presión total recibida por el micrófono,  $p_m(t)$ .

$$p_m(t) = p_i(t) + K_r \cdot r_p(t) * p_i(t - \tau_r) + \sum_u (K_u \cdot r_u * p_i(t - \tau_u)) + p_n(t) \quad (1.1.2.7.)$$

donde:

- Los términos con subíndice **u** representan reflexiones no deseadas debidas a ecos en el suelo, difracciones en el borde de la muestra, difracciones en el altavoz, reverberaciones,..
- $p_n(t)$  es el ruido de fondo recogido por el micrófono. En medidas in situ, sería el ruido producido por el tráfico o por otros agentes externos.
- **K** y  $\tau$  son factores geométricos y retrasos temporales.

En el dominio de la frecuencia la ecuación (1.1.2.7) se transforma en:

$$P_m(f) = P_i(f) + K_r \cdot R_p(f) \cdot P_i(f) \cdot e^{-2\pi f \tau_r} + \sum_u (K_u \cdot R_u(f) \cdot P_i(f) \cdot e^{-2\pi f \tau_r}) + P_n(f) \quad (1.1.2.8)$$

Que también puede ser escrita como:

$$P_m(f) = P_i(f) \left( 1 + K_r \cdot R_p(f) \cdot e^{-2\pi f \tau_r} + \sum_u (K_u \cdot R_u(f) \cdot e^{-2\pi f \tau_r}) \right) + P_n(f) = P_i(f) \cdot H(f) + P_n(f) \quad (1.1.2.9)$$

donde

- **H(f)** es la respuesta en frecuencia de un supuesto sistema lineal e invariante con el tiempo teniendo en cuenta las reflexiones de la superficie bajo ensayo y de todas las superficies existentes en el entorno del ensayo;
- $P_n(f)$  es el espectro en frecuencia del ruido de fondo.

La transformada de Fourier inversa de **H(f)** nos da la respuesta impulsiva **h(t)**.

En un principio puede parecer que para obtener  $\alpha(f)$  de la ecuación (1.1.2.6.) es necesario extraer el término relevante de la respuesta en frecuencia **H(f)**, contaminada por el ruido de fondo  $P_n(f)$ . En la práctica este problema se soluciona

más fácilmente en el dominio del tiempo, donde las distintas reflexiones pueden ser identificadas por simple inspección visual trabajando con  $h(t)$ .

El posterior paso al dominio de la frecuencia se obtiene aplicando un algoritmo FFT.

De este modo, el problema de la medida se reduce a escoger la señal de presión incidente más adecuada,  $p_i(t)$ , teniendo en cuenta la presión producida por el ruido de fondo,  $p_n(t)$ .

Como resultado, **podremos obtener el valor de la pérdida acústica por reflexión en un amplio rango de frecuencias a partir de una única medida.**

### **1.1.3. MÉTODO ADRIENNE.**



### 1.1.3.1. INTRODUCCIÓN.

Se trata de obtener la pérdida acústica por reflexión (**RL**) de una determinada barrera antirruído. Para ello hemos de seguir los pasos indicados en el **método Adrienne**. En este apartado se definen dichos pasos. Primero, hemos de obtener las señales de presión acústica (incidentes y las reflejadas). Posteriormente se debe hacer un procesado de dichas señales para la obtención de la RL.

El **programa** diseñado en este proyecto de fin de carrera realiza dicho **procesado**.

La fuente de sonido (altavoz) emite una onda transitoria (tipo **MLS**) que primero pasa por el micrófono, incide en la barrera bajo ensayo y posteriormente es reflejada. El método MLS proporciona la **respuesta impulsiva** del sistema.

El micrófono, situado entre el altavoz y la barrera antirruído, recibe tanto la onda acústica de presión **directa** como la onda de presión **reflejada**. Hay que tener en cuenta que la onda reflejada llega debilitada debido a divergencia geométrica.

Los espectros de potencia de las ondas directa y reflejada corregidos por debilitamiento de la onda, teniendo en cuenta la diferencia de caminos recorridos por cada uno de ellos, nos da la base para calcular la pérdida por reflexión (**RL**).

Es aconsejable que la medida tenga lugar en un entorno tal que no tengamos más **reflexiones** que las procedentes de la barrera bajo ensayo. Como esto no se consigue siempre, se recomienda que los picos de la respuesta en frecuencia sean tan pronunciados como sea posible. De este modo, las reflexiones procedentes de otras superficies ajenas a la barrera pueden ser rechazadas por simple inspección visual de la señal recibida por el micrófono, ya que se caracterizan por tener distinto retraso temporal.

### 1.1.3.2. EL MÉTODO ADRIENNE.

El proyecto Adrienne dio como resultado una expresión matemática para evaluar la pérdida por reflexión (RL) a partir de las señales de presión acústica medidas. Esta expresión está basada en el desarrollo matemático que aparece en el apartado 1.1.2.3. El programa desarrollado en este proyecto de fin de carrera incorpora la siguiente **expresión** usada para obtener la pérdida por reflexión (RL) como función de la frecuencia (*Garai, 1997*):

$$RL(\Delta f_j) = \frac{1}{n} \cdot \sum_{k=1}^n \frac{\int_{\Delta f_j} |F(d_r(t) \cdot p_{r,k}(t) \cdot w_r(t))|^2 \cdot df}{\int_{\Delta f_j} |F(d_i(t) \cdot p_i(t) \cdot w_i(t))|^2 \cdot df}$$

donde:

$p_i(t)$  es la presión acústica de la señal de referencia, en campo libre (sin reflexiones).

$p_{r,k}(t)$  es la presión acústica de la señal reflejada, para el ángulo k-ésimo.

$d_i(t)$  es el factor de corrección geométrica por divergencia para la señal de referencia.

$d_r(t)$  es el factor de corrección geométrica por divergencia para la onda acústica reflejada.

$w_i(t)$  es la ventana temporal para la onda acústica de referencia (ventana Adrienne).

$w_r(t)$  es la ventana temporal para la onda acústica reflejada (ventana Adrienne).

$F$  simboliza la transformada de Fourier.

$\Delta f_j$  es la  $j$ -ésima banda de tercio de octava de frecuencia (hay 18, desde 100Hz a 5kHz)

$n$  es el número de ángulos en los que se promedia (5 o 9).

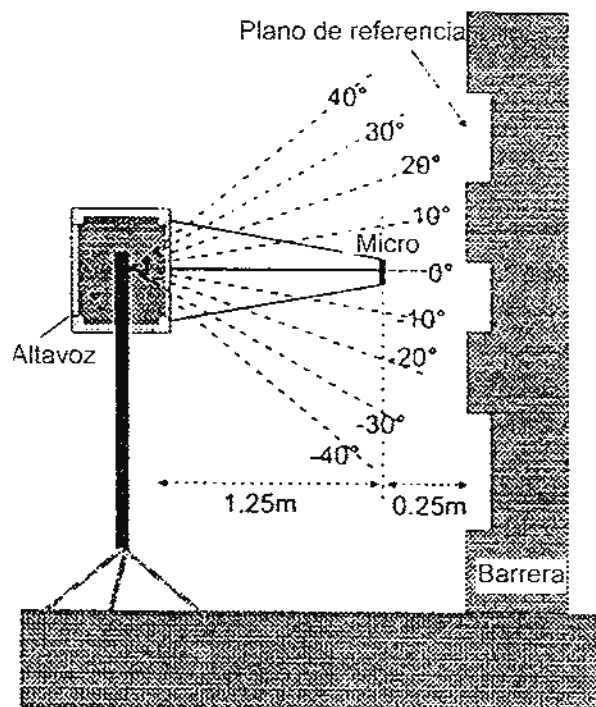


Figura 1.1.3.2.1. Disposición del equipo de medida para la obtención de la pérdida por reflexión.

El **objetivo final** es obtener promedios de la pérdida por reflexión,  $RL$ , en 18 bandas de tercio de octava, entre 100 y 5000 Hz, para distintos ángulos de incidencia.

Se debe tener en cuenta que la pérdida por reflexión es función de la **frecuencia** y del **ángulo de incidencia**.

Se hacen medidas para **nueve ángulos de incidencia** distintos, como se puede observar en la figura 1.1.3.2.1.

Para obtener la pérdida por reflexión de la barrera en cada uno de estos ángulos se deben seguir los siguientes **pasos**:

1. Medir únicamente la señal **directa** orientando el equipo de medida a lugares donde no se tengan reflexiones indeseables, como puede ser el cielo.
2. Medir la señal **directa + reflejada** dirigiendo el sistema altavoz-micrófono hacia la barrera.
3. Aplicar la "**Técnica de substracción**" para obtener únicamente el eco reflejado. La técnica de la substracción permite separar óptimamente las señales directa y reflejada en la muestra.
4. Aplicar tanto al eco directo como al reflejado una **corrección por debilitamiento** de potencia, teniendo en cuenta el camino recorrido por cada una de ellas.
5. Aplicar tanto al eco directo como al reflejado una **ventana temporal Adrienne** para extraer únicamente la parte de las señales que tienen interés.
6. Las señales directa y reflejada, ventaneadas y corregidas por la caída de amplitud, son convertidas a espectros de frecuencia mediante la **FFT**.
7. Se eleva el módulo de dichos espectros al cuadrado para obtener señales proporcionales a las **energías** reflejada e incidente, respectivamente.

8. Estos espectros FFT de energía son transformados a espectros en **bandas de tercio de octava** mediante un algoritmo desarrollado por *Garai (1996)*.
9. Dividiendo el espectro de energía reflejada entre el espectro de energía incidente se obtiene el valor de la pérdida por reflexión para el ángulo que estemos ensayando.

Este procedimiento ha de repetirse para nueve ángulos de incidencia: **50°, 60°, 70°, 80°, 90°, 100°, 110°, 120° y 130°**. Estos son los ángulos que forma el equipo de medida con la normal al suelo.

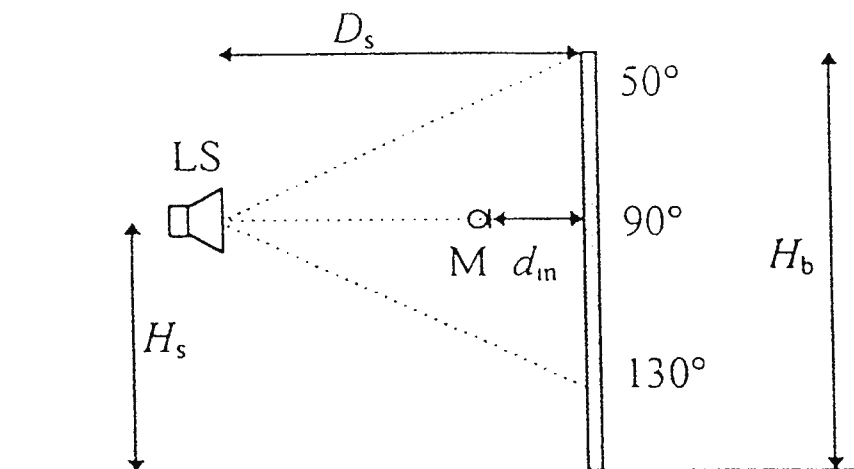


Figura 1.1.3.2.2. Ángulos de orientación del sistema altavoz-micrófono.

Finalmente, se debe presentar como resultado la RL, en bandas de tercio de octava, promediada para los siguientes ángulos:

- RL promediada entre 50° y 90°.
- RL promediada entre 90° y 130°.
- RL promediada entre 50° y 130°.

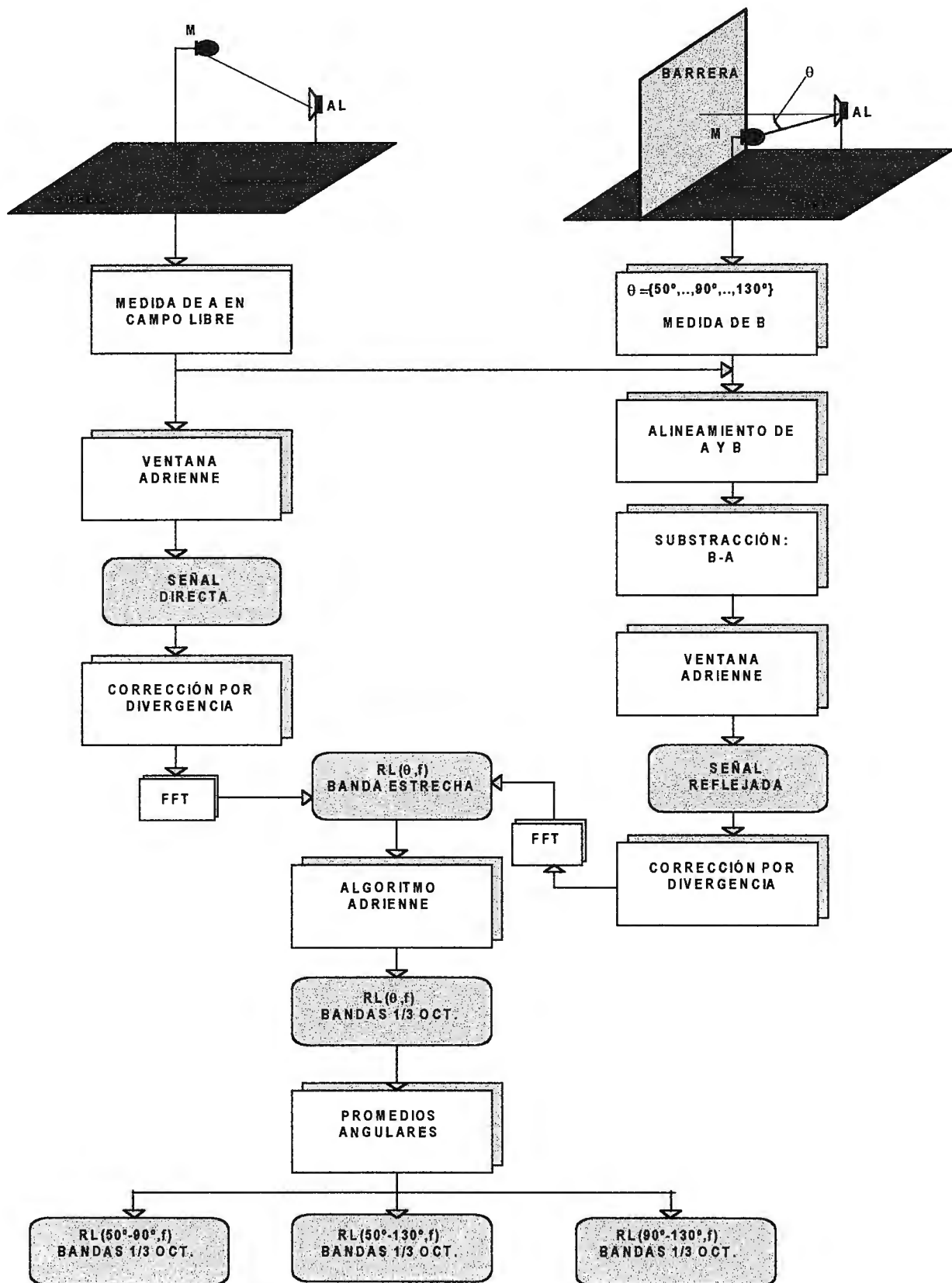
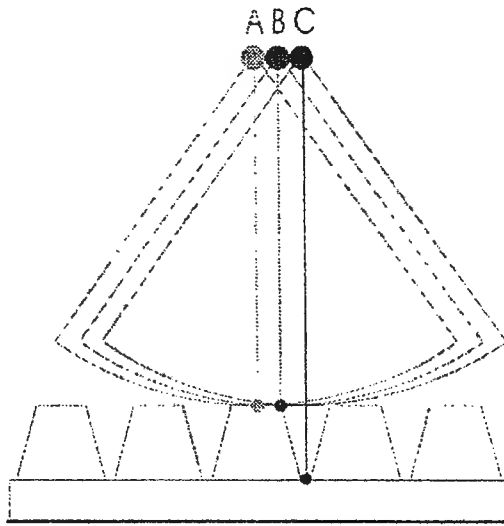


Figura 1.1.3.2.3. Método Adrienne para hallar la pérdida por reflexión.

Cuando la superficie de la barrera ensayada no sea plana, sino que disponga de un determinado perfil no bastará con hacer el barrido de ángulos para un solo punto de la barrera. Teniendo en cuenta el perfil se tomarán algunos **puntos característicos** del mismo y, en cada uno de ellos se hará el barrido angular (desde  $50^\circ$  hasta  $90^\circ$ ).



Para la obtención de la RL se hace, en este caso, el promedio para los distintos puntos característicos, dando el mismo peso a cada uno de ellos.

Figura 1.1.3.2.3. Distintos puntos característicos en una barrera no plana



Figura 1.1.3.2.4. Ejemplo de barrera no plana (en zig-zag)

### **1.1.3.2.1. TÉCNICA DE SUBSTRACCIÓN DE SEÑAL.**

Se aplica para obtener los términos  $p_i(t)$  y  $p_r(t)$  que aparecen en la expresión del apartado "1.1.3.2. MÉTODO ADRIENNE" para evaluar la pérdida por reflexión.

Hay que tener en cuenta que en el método Adrienne las señales directa y reflejada son recogidas por un mismo micrófono. La señal de presión acústica recogida contendrá, por tanto, tanto al eco directo como al reflejado. Como la pérdida por reflexión se calcula como el cociente de la energía reflejada entre la incidente, para hallar estas energías se deben tener las señales directas y reflejada separadas. **La técnica de substracción permite separar óptimamente las señales directa y reflejada en la barrera.**

Para la correcta aplicación de esta técnica se deben seguir los siguientes **pasos:**

- 1.- Medir la señal directa en campo libre, de modo que no tengamos ninguna reflexión. Esto implica que el conjunto altavoz-micrófono no debe estar orientado hacia la barrera antirruído. La llamaremos señal **A**.
- 2.- Medir la señal directa y reflejada dirigiendo el sistema altavoz-micrófono hacia la barrera. La llamaremos señal **B**.
- 3.- Alinear las señales **A** y **B** en el registro informático. Esto significa que los máximos de las dos señales deben estar en la misma posición en los dos vectores que contienen las señales **A** y **B**.
- 4.- Substraer la señal **A** de la señal **B**, es decir hacer  $B-A$ . De este modo, tendremos dos señales: la señal **A** que contiene únicamente la señal directa y



la señal **B-A** que contiene la señal reflejada y otras perturbaciones no deseadas.

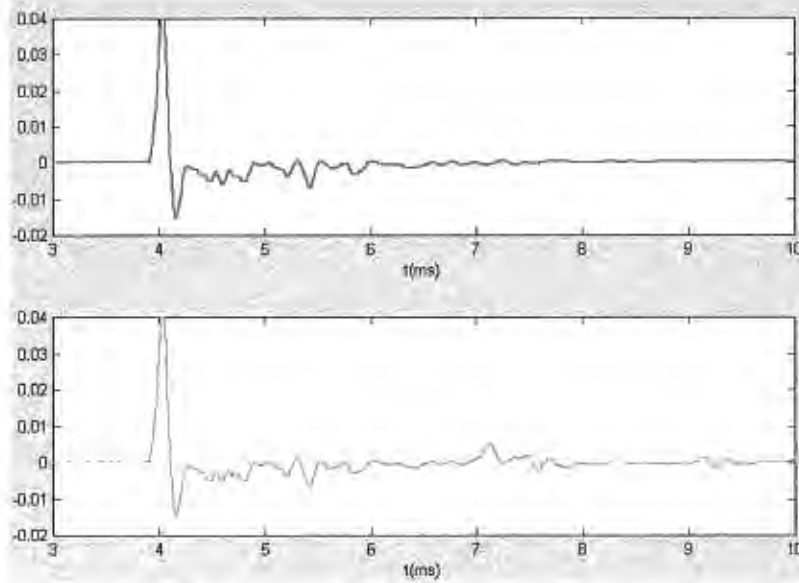


Figura 1.1.3.2.1.1. (Arriba) Señal de presión acústica para la señal directa (A).  
(Abajo) Señal de presión acústica para la señal directa + reflejada (B).

Este procedimiento se explica en la siguiente figura. Para que el límite inferior de frecuencia del método sea lo más bajo posible, la separación entre la señal reflejada en la muestra (R) y difractada o reflejada en objetos próximos (E1, E2,...) ha de ser máxima. Esto requiere que el micrófono esté lo más próximo posible a la muestra y que el altavoz esté alejado de ambos. La consecuencia es que los eventos directo (D) y reflejado (R) están muy próximos en la traza. Si mantenemos fija la separación altavoz-micrófono y giramos el conjunto hacia el espacio abierto, se obtendrá la traza de la **Figura (b)**, donde el evento directo (D) aparece exactamente en la misma posición que en la traza de la **Figura (a)**. De esta traza de referencia se extrae entonces la **señal directa**. De la diferencia entre ambas trazas, **Figura (c)**, se extrae la **señal reflejada**.

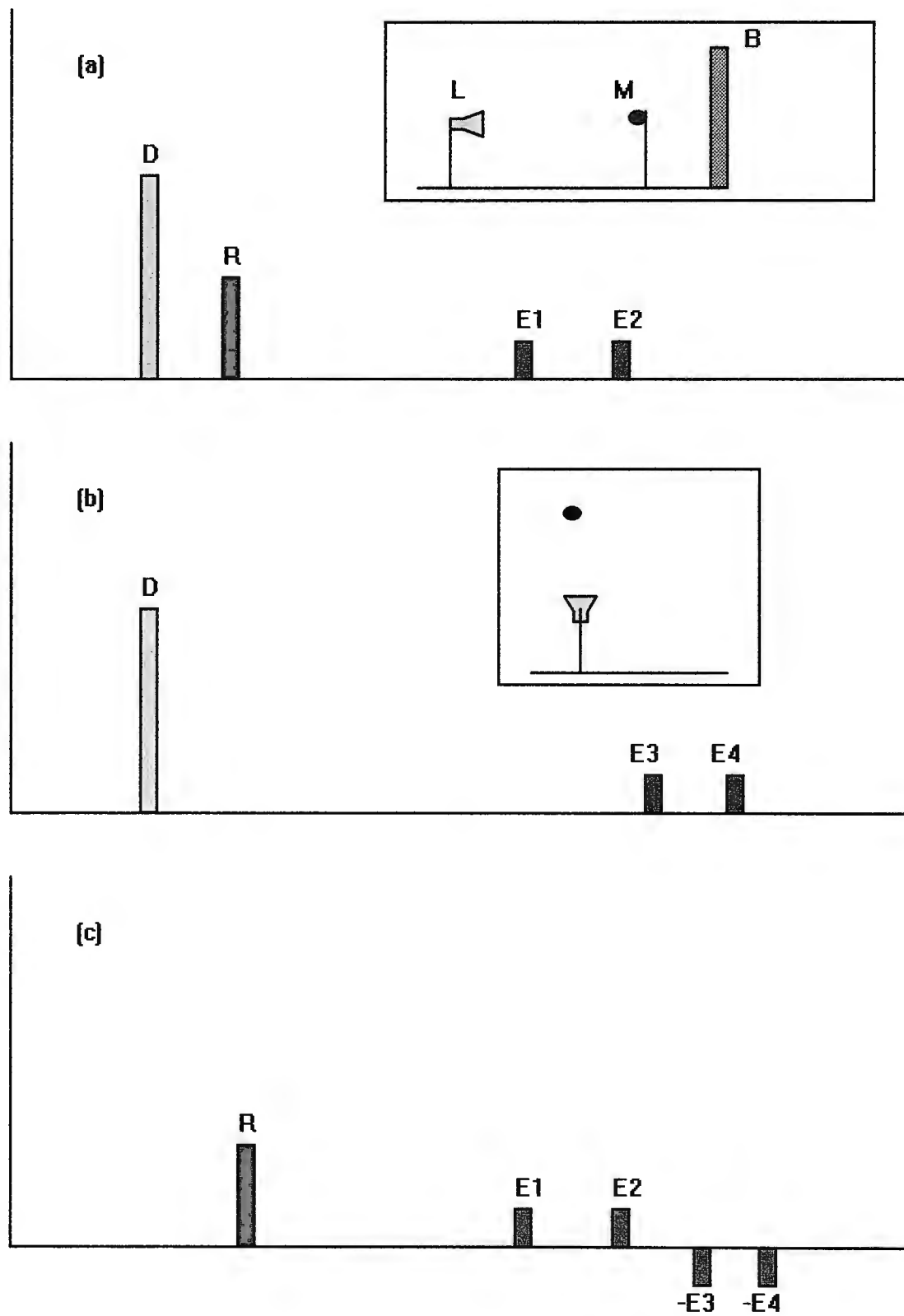


Figura 1.1.3.2.1.2. El método de la substracción de señal. (a) Traza de reflexión. (b) Traza de referencia. (c) Traza (a)-(b)

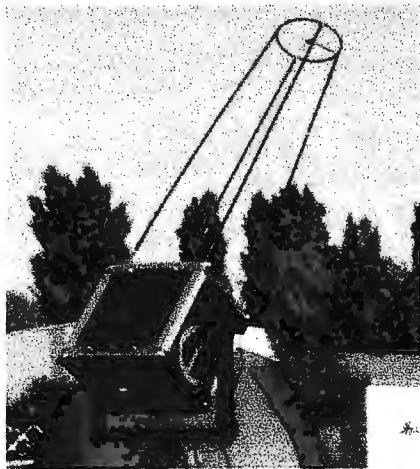


Figura 1.1.3.2.1.3. El sistema electroacústico.

Para la aplicación de la técnica de la substracción resulta esencial que la **distancia altavoz-micrófono** se mantenga **constante** a lo largo de las medidas. Para ello, se usa un sistema electroacústico (Figura 1.1.3.2.1.3.) que consta de un altavoz JBL 2123H de 10 pulgadas, y un micrófono de electret, unidos rígidamente a través de cuatro varillas telescópicas.

El sistema de medida incluye además un amplificador y un ecualizador para el altavoz, y un Instrumento Virtual (PC + tarjeta DSP + software de control) que implementa el método **MLS**.

A continuación, se aplica a las señales **A** y **B-A** una ventana temporal para extraer únicamente la información que queremos de cada señal. Es la llamada **ventana Adrienne**, que se describe en el siguiente apartado. Este proceso de ventaneo también forma parte de la "Técnica de substracción".

### 1.1.3.2.2.VENTANA ADRIENNE.

#### 1.1.3.2.2.1. DEFINICIÓN DE LA VENTANA ADRIENNE.

La ventana temporal, como se ha descrito en el apartado anterior, se aplica tanto a la señal reflejada como a la directa para separarlas. Forma parte de la "**Técnica de substracción**".

La ventana temporal es una función del tiempo por la que se va a multiplicar la señal de presión acústica obtenida en el micrófono. Valdrá cero en aquellos tramos de la señal medida que queramos ignorar, que no queramos transformar al dominio de la frecuencia.

La ventana Adrienne presenta los siguientes **tramos**:

- Una "**pre-ventana**" en la parte izquierda, con la forma de la mitad izquierda de una ventana **Blackman-Harris**, de **0.5 ms** de longitud.
- Un **cuerpo principal rectangular** (aquí la ventana toma valor 1) de longitud **0.7** veces una magnitud llamada "**tiempo de retraso**".
- El **tramo derecho** tiene la forma de la mitad derecha de una ventana **Blackman-Harris** y tiene una longitud de **0.3** veces el "**tiempo de retraso**".

La forma de la ventana Adrienne se presenta en la siguiente figura:

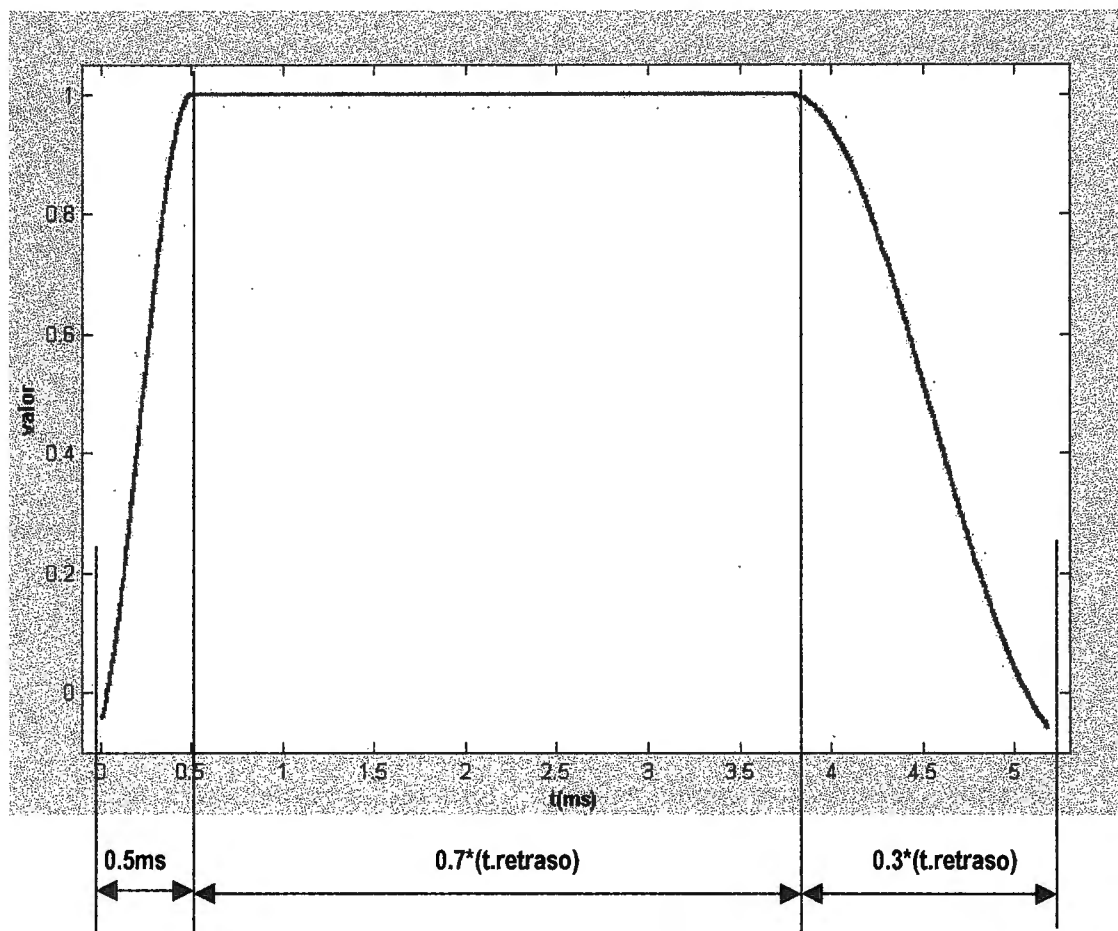


Figura 1.1.3.2.2.1.1. Ventana Adrienne.

Por tanto, la longitud temporal de la ventana Adrienne es de:

$$\Delta t = .5 + (\text{tiempo de retraso}) \text{ (en ms)}$$

El valor del tiempo de retraso aparece en el apartado "1.2.1.2. Conclusiones." perteneciente al apartado "1.2.1. Cálculos concernientes a los parámetros geométricos asociados a la medida".

### 1.1.3.2.2.2. UBICACIÓN Y TAMAÑO DE LA VENTANA ADRIENNE.

La ventana para la señal directa y para la reflejada es la misma, pero su **ubicación** en el tiempo es **distinta** debido a la diferencia de tiempos en la señal para las ondas directa y reflejada. Esto se puede observar en la siguiente figura.

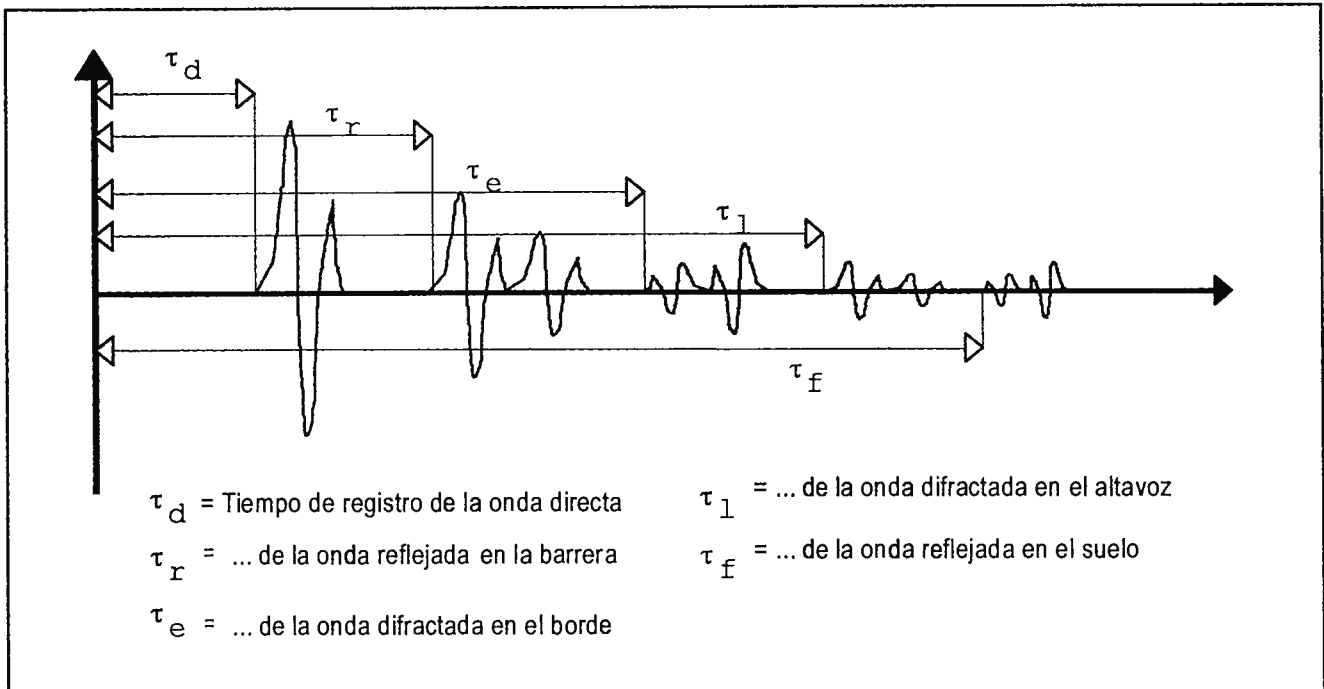


Figura 1.1.3.2.2.2.1. Tiempos de llegada al micrófono de los distintos pulsos de presión.

La **longitud temporal** de la ventana Adrienne es **variable**, depende de los parámetros geométricos del ensayo. Se trata de conseguir que la ventana tenga la máxima longitud temporal posible (*así la frecuencia fiable será lo menor posible*) pero que, a su vez, no coja reflexiones indeseadas.

Esto último representa un **avance** con respecto a lo concluido por el equipo investigador del proyecto Adrienne. Se concluyó que la ventana Adrienne debe ser la misma independientemente de las condiciones geométricas del ensayo. Un estudio realizado por P.Cobo (1999) demuestra que se debe hacer un estudio más riguroso

del tema. Existe, también una configuración óptima para la cual mínima frecuencia más baja fiable.

**Este estudio ha sido tenido en cuenta en este proyecto de fin de carrera. En el programa se informa al usuario de la configuración geométrica óptima, así como de la mínima frecuencia fiable para la configuración usada en el ensayo.**

Es necesario hacer un pequeño **estudio de los tiempos** que tardan en llegar los distintos ecos (directo, reflejado en la barrera, reflejado en el suelo, difractado en el borde de la barrera...) al micrófono con el fin de determinar:

- Ubicación de la ventana para el eco directo y el reflejado.
- Longitud temporal de la ventana Adrienne (tiempo de retraso).
- Parámetros geométricos óptimos del ensayo para una óptima mínima frecuencia fiable.

Las conclusiones obtenidas en dicho estudio han sido incluidas en el procesado de la señal (software), de modo que:

- El programa procesa la señal con la **ventana** de tamaño más **adecuado**. En una de las pantallas de salida ("Comprobación del ventaneo") se puede comprobar cómo se ha realizado el proceso de ventaneo.
- También es una salida del programa la **mínima frecuencia** por debajo de la cual los resultados obtenidos para la pérdida por reflexión no son fiables.

Este estudio aparece en el apartado 1.2. "CÁLCULOS", perteneciente a la memoria. En él se presentan las ecuaciones que relacionan los parámetros

geométricos con los tiempos de llegada de los distintos pulsos de presión al micrófono.



### **1.1.3.2.3. CORRECCIÓN POR DEBILITAMIENTO DE ONDA.**

También se denomina "**corrección por divergencia**".

Si se asume modelo de onda **esférica**, la presión acústica decae de forma inversamente proporcional a la distancia avanzada. Este modelo de onda es el asumido para las ondas acústicas.

Para compensar esta caída de amplitud se multiplica a la presión acústica por una función ganancia:

$$d(t) = \text{distancia avanzada} = c \cdot t$$

donde:

- **c** es la velocidad del sonido (0.34 m / ms).
- **t** es el tiempo transcurrido.

En la **figura 1.1.3.2.3.1.** se puede observar la función ganancia y una señal de presión corregida y sin corregir.

Se puede observar que la mayor diferencia entre las dos señales está en la parte final, ya que ahí el valor de la función ganancia es mayor.

Es importante hacer la corrección por divergencia ya que cuando se halle la pérdida por reflexión según la expresión del apartado "**1.1.3.2. EL MÉTODO ADRIENNE**", la pérdida de energía debida al debilitamiento de la onda no afectará al resultado.

Si no se hiciese esta **corrección** el valor obtenido para la **RL** sería **menor** al real.

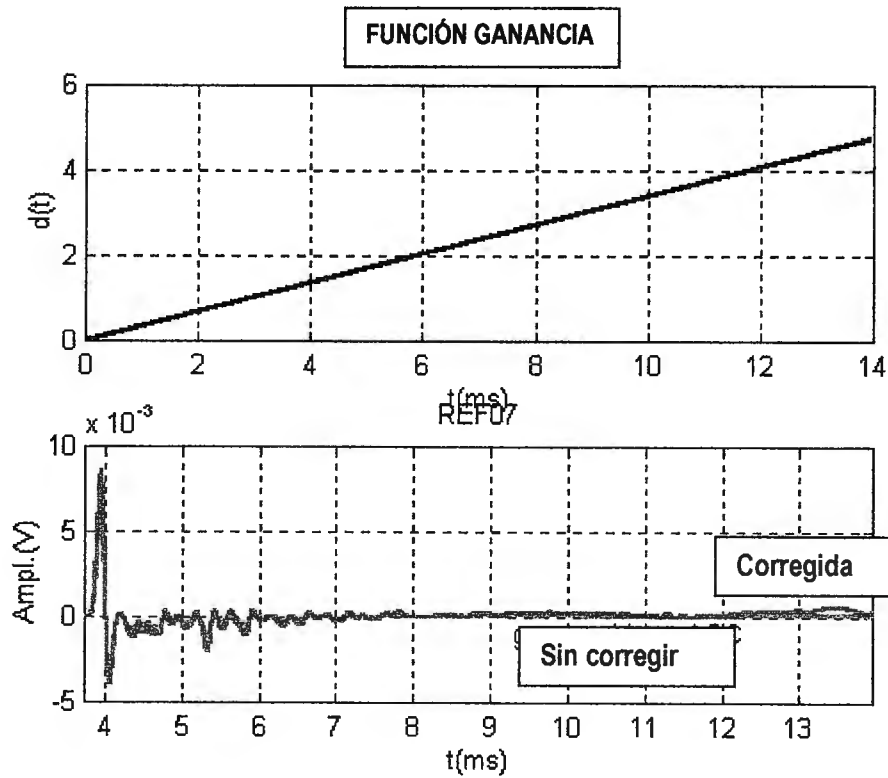


Figura 1.1.3.2.3.1. Corrección de la onda de presión acústica por divergencia.

### 1.1.3.2.4. MÍNIMA FRECUENCIA FIABLE.

El efecto de **truncamiento** en frecuencia debido al uso de ventanas temporales para realizar el paso a espectro de frecuencia es algo inevitable. Cuando una señal es ventaneada parte de su energía en **baja frecuencia** no es tomada en cuenta. Por tanto el ventaneo siempre implica un límite inferior de resolución en frecuencia.

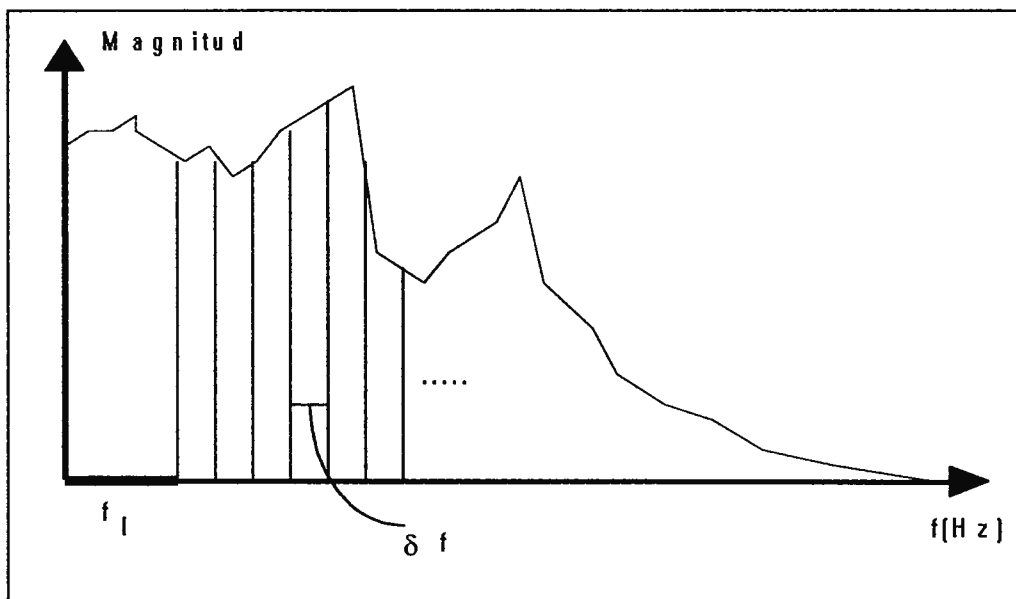


Figura 1.1.3.2.4.1. Mínima frecuencia fiable e incremento de frecuencia.

$f_l$ : Mínima frecuencia fiable; depende de la longitud temporal de la ventana.

$\delta f = (1/N\delta t)$ : Incremento de frecuencia. Decece al aumentar N, número de puntos donde se aplica la transformada de Fourier

Para una ventana rectangular se cumple:

$$f_l = 1/T_w \quad (16)$$

donde  $T_w$  es el tamaño (duración de la ventana).

Pero, ¿qué ocurre con otras ventanas?. Las ventanas se caracterizan por varios parámetros (*Nuttal, 1981; Gade y Herlufsen, 1987*). Algunos de estos parámetros son:

- La **anchura del lóbulo principal** (primer mínimo del espectro (módulo) de la ventana).
- El **ratio de amplitud** lóbulo principal / primer lóbulo.
- La **tasa de decrecimiento** de los lóbulos secundarios.

El espectro de módulo de la ventana rectangular presenta el lóbulo principal más estrecho, pero los lóbulos secundarios son los más altos. Esto puede dar problemas para detectar bajas frecuencias comunes en aplicaciones acústicas. Ventanas de Hanning, Hamming, **Blackman-Harris**, Kaiser-Bessel, o Chebyshev pueden ser usadas en estos casos. Todas estas ventanas tienen sus propios parámetros característicos.

El criterio tomado para determinar la mínima frecuencia fiable asociada a una determinada ventana es la **anchura del lóbulo principal** (*Cobo, 1999*). Se explica a continuación.

- La **figura 1.1.3.2.4.2.(a)** muestra una señal periódica de 100Hz.
- La **figura 1.1.3.2.4.2.(b)** muestra la señal ventaneada con una ventana rectangular de 10 ms.
- La **figura 1.1.3.2.4.2.(c)** muestra la señal ventaneada con una ventana Blackman-Harris de 10 ms.

Mientras que la ventana rectangular recoge un periodo completo, la ventana Blackman-Harris no lo consigue

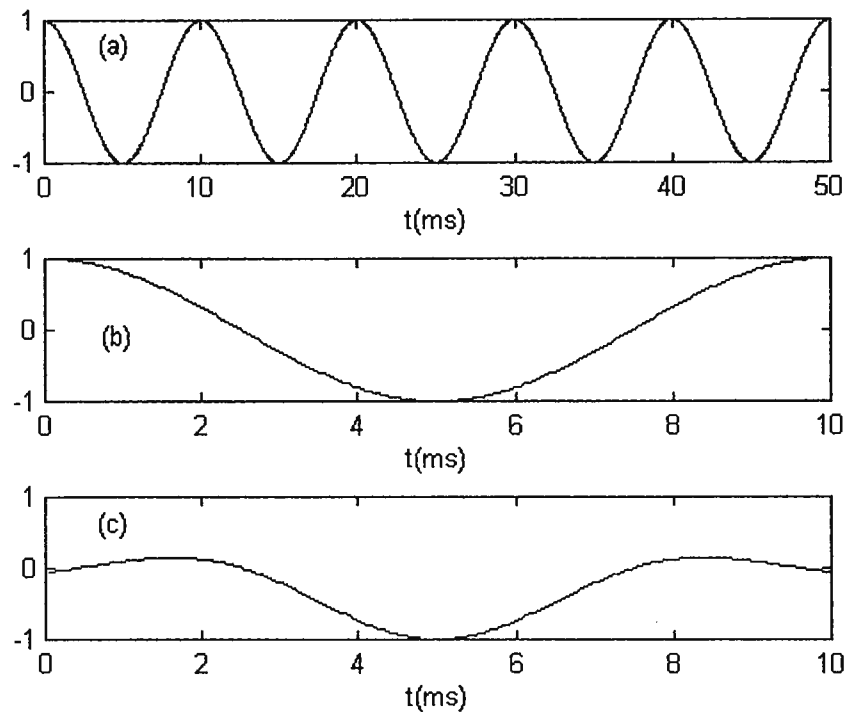


Figura 1.1.3.2.4.2. : (a) señal periódica de 100Hz. (b) señal ventaneada con una ventana rectangular de 10 ms. (c) señal ventaneada con una ventana Blackman-Harris de 10 ms.

Este efecto de truncamiento se aprecia mejor en el dominio de la frecuencia (Figura 1.1.3.2.4.3.). En la figura 1.1.3.2.4.2. se puede apreciar el teorema de la convolución. Como es conocido, el espectro de la señal ventaneada es la convolución del espectro de la señal original con el espectro de la ventana (la función *sinc* en el caso de la ventana rectangular).

Nótese que:

- La ventana **rectangular** tiene el lóbulo principal más estrecho, pero los lóbulos laterales más altos.
- La ventana de **Blackman-Harris** tiene un lóbulo principal más ancho pero lóbulos laterales bajos.
- Mientras la ventana rectangular puede recoger la totalidad de la señal a 100 Hz, Blackman-Harris no.

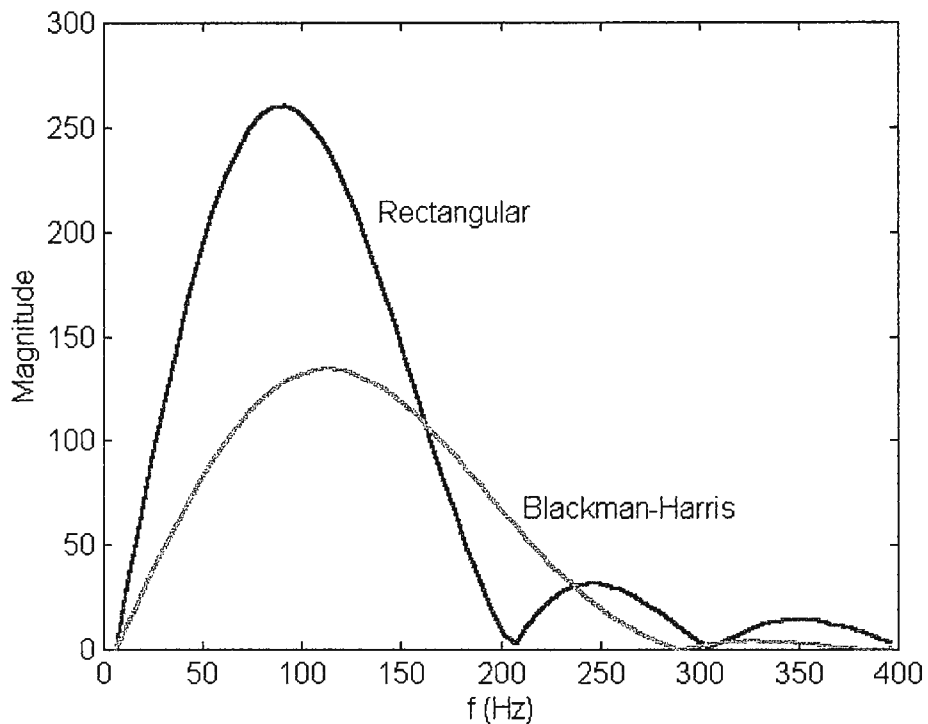


Figura 1.1.3.2.4.2. Espectro de módulo de las ventanas de 10ms rectangular y Blackman-Harris.

**Por tanto, la mínima frecuencia fiable viene limitada por la anchura del lóbulo principal del espectro de módulo de la ventana.**

La Figura 1.1.3.2.3.3. muestra la ventana para la señal reflejada acordada en el proyecto Adrienne. El tamaño de la ventana Adrienne se calcula como:

$$T_w = 0.5 + (\tau_{ne} - \tau_r) \quad (\text{ms}) \quad (17)$$

donde  $\tau_{ne}$  es el tiempo de registro del pulso de presión inmediatamente posterior al pulso de la onda reflejada en la barrera (o el difractado en el altavoz o el reflejado en el suelo o el difractado en el borde, dependiendo de la disposición geométrica).

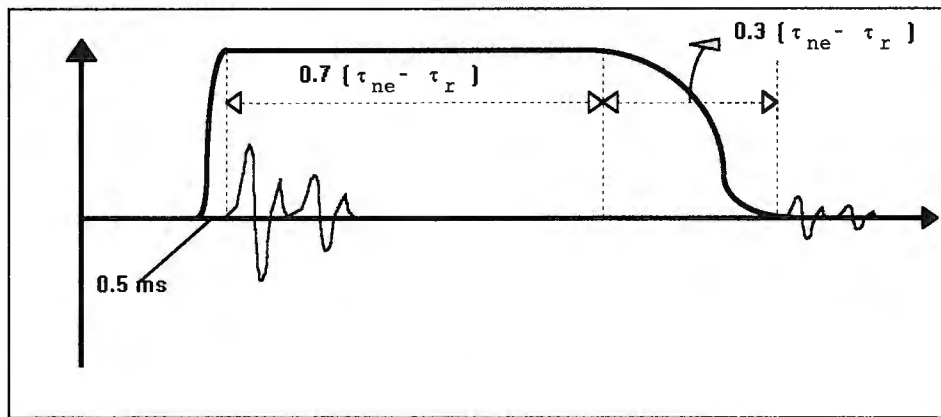


Figura 1.1.3.2.4.3. La ventana Adrienne.

La Figura 1.1.3.2.3.4. muestra la ventana Adrienne y su espectro, obtenidos con el programa desarrollado en este proyecto de fin de carrera, para los siguientes parámetros geométricos:

- Distancia micrófono-barrera: 0.25 m.
- Distancia altavoz-micrófono: 1.25 m.
- Altura de la barrera: 4m.

La mínima frecuencia fiable, en este caso es de 240 Hz.

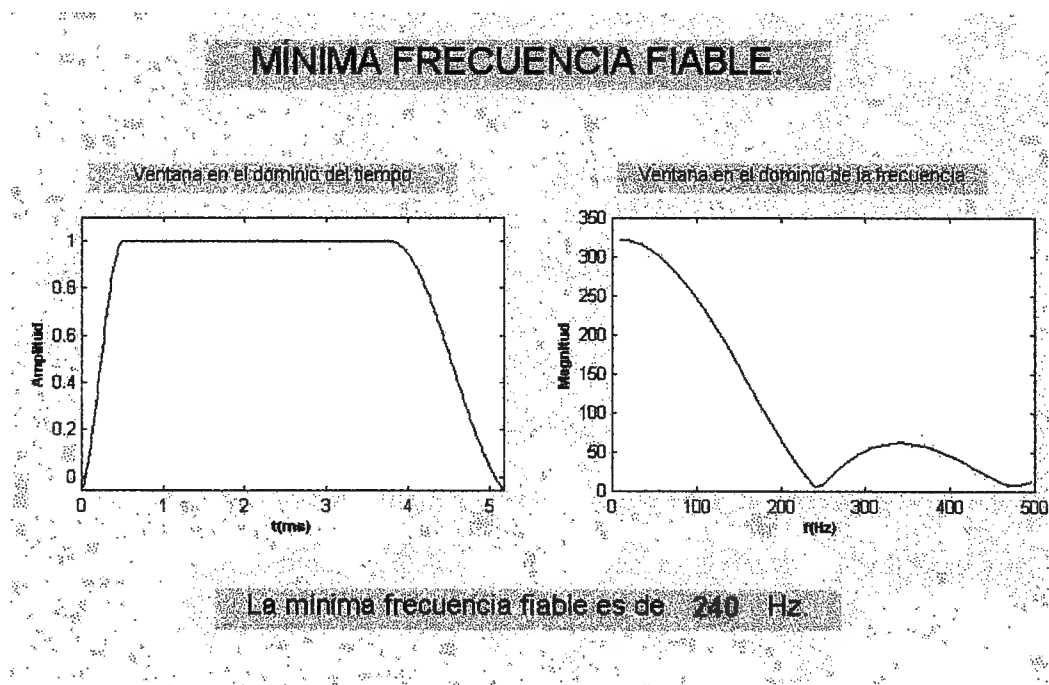


Figura 1.1.3.2.4.4. Ventana Adrienne con su espectro.

### 1.1.3.2.5. EL MÉTODO MLS Y EL SISTEMA MLSSA

El método MLS requiere sólo una señal de entrada. Como señal de salida usa una secuencia pseudoaleatoria de máxima longitud. Esta secuencia consiste en una serie binaria de  $\pm 1$ , de longitud  $L=2^N-1$ , donde  $N$  es un número entero. Esta señal MLS, aún siendo periódica de periodo  $T=L \Delta t$ , donde  $\Delta t$  es el periodo de muestreo, tiene autocorrelación circular 1 en el mismo periodo  $T$ .

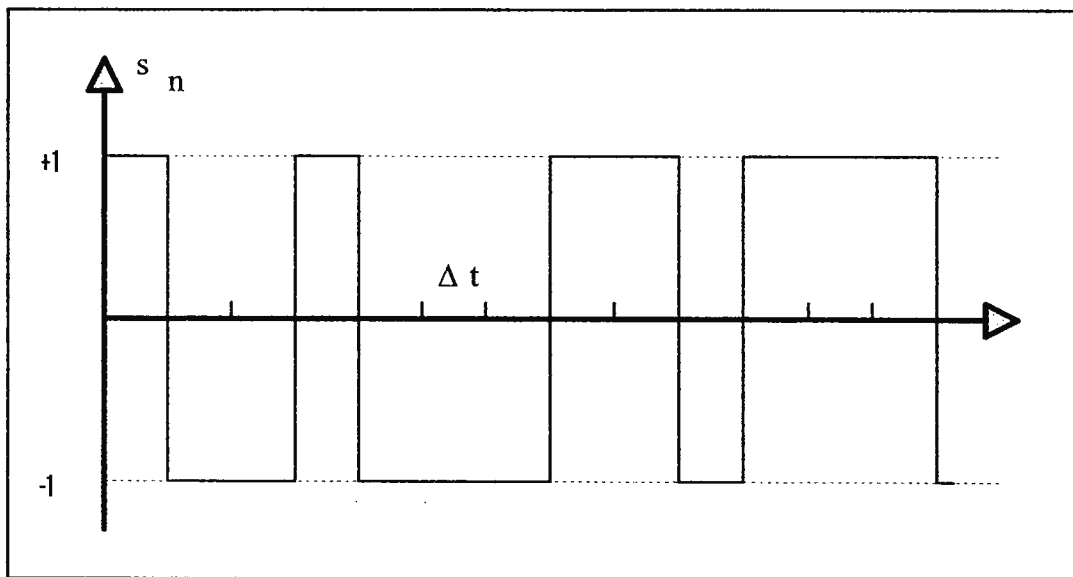


Figura 1.1.3.2.5.1. Una secuencia pseudoaleatoria de máxima longitud (MLS)

El esquema de la medida de la respuesta impulsiva de un sistema lineal usando señales MLS se muestra en la Figura 8. Sean  $s(t)$  la señal MLS,  $u(t)$  la respuesta impulsiva del altavoz,  $p_i(t)$  la señal radiada por el altavoz, y  $p_m(t)$  la señal medida por el micrófono. La señal que radia el altavoz es:



$$p_i(t) = s_n * u(t) = \sum_{n=0}^{L-1} s_n u(t - n\Delta t) \quad (1)$$

donde \* denota convolución.

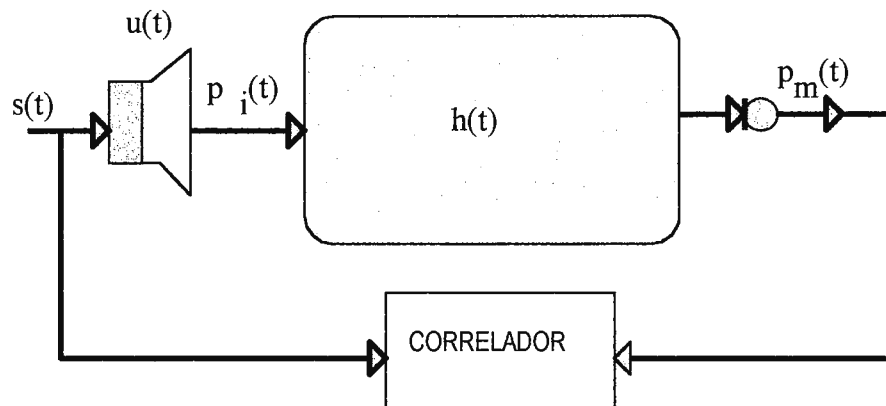


Figura 1.1.3.2.5.2. Esquema de la medida de la respuesta impulsiva por el método MLS

La señal que mide el micrófono es

$$p_m(t) = s_n * u(t) * h(t) = \sum_{n=0}^{L-1} s_n u(t - n\Delta t) h(t - n\Delta t) \quad (2)$$

Y si muestreamos a  $f_s = 1/\Delta t$  (y despreciamos  $u(t)$ ; aunque no es necesario.

Es posible *deconvolucionar* o *ecualizar* la respuesta del altavoz)

$$(p_m(t))_k = s_n * h_k = \sum_{n=0}^{L-1} s_n h_{k-n} = \sum_{n=0}^{L-1} s_{k-n} h_n \quad (3)$$

La correlación cruzada con la secuencia MLS proporciona

$$\Phi_{sm}(n\Delta t) = \frac{1}{L+1} \sum_{k=0}^{L-1} s_k (s_{k+n} * h_{k+n}) = \frac{1}{L+1} \sum_{k=0}^{L-1} (s_k s_{k+n}) * h_{k+n} = \Phi_{ss}(n) * h_n = \left( \delta_n - \frac{1}{L+1} \right) * h_n \quad (4)$$

Poniendo  $\tau = n\Delta t$  y partiendo  $\frac{1}{L+1} = \frac{1}{L} - \frac{1}{L(L+1)}$  queda

$$\Phi_{sm}(\tau) = \Phi_{ss} * h_n = h(\tau) - \frac{1}{L} \sum_{n=0}^{L-1} h(\tau) + \frac{1}{L(L+1)} \sum_{n=0}^{L-1} h(\tau) \quad (5)$$

En resumen, la correlación cruzada de la secuencia MLS con la señal medida en el micrófono, sobre un periodo  $L$ , tiende a la respuesta impulsiva cuando  $L$  tiende a un número grande.

El método MLS tiene las siguientes **ventajas**:

- Permite la **separación** temporal de los diferentes términos que intervienen en la respuesta impulsiva, como en las técnicas de impulsos.
- Tienen un **espectro en frecuencias plano**, como un impulso ideal.
- **Repetibilidad** absoluta, como los *tono-burst*.
- Sin alinealidades.
- **Inmunidad a ruido de fondo**, como en los métodos de correlación.
- Se pueden usar **in situ**.

A diferencia del ruido blanco, una secuencia de máxima longitud es **determinista** y periódica, aunque retiene muchas de las propiedades del ruido blanco. Su naturaleza determinista significa que puede ser computada, y no requiere ser medida simultáneamente con la respuesta del sistema.

Una característica numérica importante del método MLS es la eficiencia en el cálculo de la correlación cruzada de secuencias largas ( $L=65535$ , por ejemplo). Una transformada de Hadamard permite reducir el cálculo de esta correlación a sumas y restas.

El método MLS se ha empleado para medir la Pérdida por Reflexión de barreras antirruído (Garai, 1993; Cobo et al, 1997), para medir la Perdida por Inserción de encapsulamientos acústicos (Cobo et al, 1998), y para la medida de las características acústicas de transductores aéreos (Sigüero y Cobo, 1998) e hidroacústicos (Ranz y Cobo, 1998).

**MLSSA** (Maximum Length Sequences System Analysis) es un sistema comercial, de DRA (Rife and Vanderkooy, 1989; Rife, 1996), basado en el método MLS, que esencialmente mide la respuesta impulsiva de un sistema lineal. Una ventaja importante de MLSSA es su producto  $BT=20000$ , lo que permite medir en un ancho de banda de 20 kHz con una resolución de 1 Hz. Además, MLSSA ofrece una serie de funciones de post-procesado.

Sus **características** de hardware son:

- Convertidor A/D de *12 bits*
- Frecuencia de muestreo de *120 kHz* (bus ISA) o *160 kHz* (bus EISA)
- Filtro antialiasing programable tipo Chebyshev, Bessel o Butterworth, con frecuencia de corte programable entre 1 y 40 kHz, y ganancia programable entre -14 y +54 dB
- Longitud de estímulo, *L*, de 4095, 16383, 32767, y 65535 puntos
- Señales de entrada entre 10 mV y 20 V rms. Impedancia de entrada de 13 kohm
- Señal de salida de hasta 5.25 V y 75 ohm de impedancia
- Ruido de fondo de -72 dB
- Distorsión armónica residual de 0.015 %, o -76 dB

Mediante software se pueden programar las condiciones de adquisición, y el post-procesado, tanto en el dominio de tiempos como de frecuencias.

### 1.1.3.3.CARACTERÍSTICAS DEL MÉTODO ADRIENNE.

#### 1. Comparabilidad.

Los resultados obtenidos por este método, recientemente desarrollado, son comparables con los resultados de otros métodos ya establecidos, en los que ya existe normativa específica. En la práctica, esto significa comparabilidad entre el método **Adrienne** y los métodos de laboratorio existentes, ya que no existen todavía otros métodos in situ. Los valores obtenidos en incidencia normal por el método del tubo **KUNDT** son perfectamente comparables con los valores para el nuevo método bajo la misma incidencia.

#### 2. Representatividad.

Los resultados de las medidas deben ser representativos de las propiedades acústicas de la muestra bajo prueba, ya que éstos son relevantes para la ingeniería acústica. Por esta razón, las condiciones de la prueba son representativas del uso del material o barrera ensayada. Los resultados no sólo deben describir la calidad del material, sino también la **calidad del montaje**.

Los resultados obtenidos por el método Adrienne son representativos de las características acústicas de la barrera antirruído; se hacen sobre la misma barrera una vez instalada (por tanto, influye la calidad del montaje) y los ángulos de incidencia del ensayo son aquellos que se suelen presentar en fuentes de ruido de tráfico.

### **3. Margen de frecuencias**

Debe incluir todas las frecuencias pertinentes a la generación y transmisión de la clase de sonido con que la barrera tendrá que trabajar, teniendo en cuenta la escala A de ponderación acústica. Muchas normas de la EN adoptan un margen de frecuencias desde 100 Hz a 5 kHz ahora. Por eso, aunque esto pueda resultar demasiado exigente para los métodos actuales de medida absorción acústica in situ, el margen de frecuencias adoptado para el método Adrienne es:

$$100\text{Hz} \leq f \leq 5\text{kHz}$$

Todavía se tienen problemas para conseguir que la frecuencia más baja fiable sea de 100Hz. Ésta está relacionada con la distancia que ha de recorrer la onda acústica, con el procesado de la señal y con el tamaño de la barrera antirruido.

### **4. Resolución de frecuencia.**

La determinación de la resolución en frecuencia es importante ya que un cambio en ella puede dar lugar a un cambio de frecuencia en la cresta de  $\alpha(f)$ , así como una variación en su ancho de banda puede dar valiosa información sobre el grado de contaminación y degradación del material bajo prueba.

Según las pautas del CEN (Comité de Normalización Europeo), la resolución de frecuencia requerida debe ser de 1/3 octava:

$$\frac{\Delta f}{f} \leq 0.23$$

## **5. Inmunidad al ruido de fondo.**

Ya que el método se puede usar en laboratorio y en carretera, debe ser lo más **insensible** posible **al ruido del fondo** (típicamente el ruido de tráfico en carretera) y a las reflexiones perturbadoras del ambiente, es decir, el método debe lograr una alta relación SEÑAL/RUIDO cuando el ruido de fondo tenga un nivel acústico comparable o aún mayor que el de la señal del ensayo.

Esto se consigue en el método Adrienne gracias a la señal emitida por el altavoz, una señal **MLS**, que por sus propiedades matemáticas (consigue autocorrelación circular unitaria) es altamente inmune al ruido de fondo.

## **6. Efectos del tamaño de la muestra**

El método es aplicable para los diferentes ángulos de incidencia en la muestra bajo prueba. Cambiar la orientación del equipo de medida de un ángulo a otro requiere poco tiempo.

El método debe tomar valores representativos de  $\alpha(f)$  incluso cuando el campo acústico es complicado debido a la estructura tridimensional del dispositivo (por ejemplo, una barrera no plana). Que la barrera sea no plana puede afectar al margen de frecuencias válido y a la dependencia angular de  $\alpha(f)$ .

El método **Adrienne** todavía, a fecha de entrega de este proyecto de fin de carrera, tiene problemas para determinar la pérdida por reflexión en dispositivos no planos (típicamente en barreras en zig-zag o en dispositivos volumétricos. Se debe tener en cuenta que la forma de la superficie y los ángulos de incidencia están relacionados entre sí y con el tamaño mínimo de la muestra requerida para la prueba.

## **7. Precisión.**

El método es lo suficientemente preciso para detectar, de forma fiable, diferencias en los valores de  $\alpha(f)$  significativos de la actuación acústica de la barrera antirruído bajo prueba. Con respecto a los campos principales de aplicación, el rango de la medida y la precisión es:

$$0 \leq \alpha(f) \leq 1 \quad (\pm 0.05)$$

Este criterio está basado en la fiabilidad de las prácticas de laboratorio actuales. En cualquier caso, la precisión será una función de la frecuencia, de la instrumentación usada y de los procedimientos de cálculo.

## **8. Repetibilidad.**

La repetibilidad deben ser comparable con la de ensayos acústicos rutinarios, incluso cuando se realicen in situ. Ya que no existen todavía ensayos que cumplan esta condición, ni a nivel europeo (EN) ni a nivel mundial (ISO), este método sentará las bases de repetibilidad para futuros ensayos.

La idea es que dos barreras acústicas iguales que sean ensayadas en dos países distintos de la Comunidad Económica Europea den resultados semejantes, independientemente de las condiciones en que han sido ensayados. De este modo, los materiales para barreras antirruído pueden circular libremente por el mercado europeo.

### **9. Calibración.**

El procedimiento de medida descrito más arriba está basado en obtener cocientes de espectros de potencia de señales obtenidas con el mismo equipo de medida y bajo las mismas condiciones de ensayo. Es evidente que, por tanto, no es necesaria una calibración absoluta del nivel de presión acústica para el equipo de medida.

### **10. Facilidad de aplicación.**

La instrumentación se instala y se desinstala rápidamente en el sitio de ensayo. Además, el equipo de medida es fácilmente transportable, es resistente y no requiere la especialización particular del operario.

### **11. Resultados inmediatos.**

No requiere mucho tiempo procesar los datos, una vez adquiridos. El software necesario para el procesado de las señales es relativamente sencillo y tarda poco tiempo (del orden de pocos minutos) en obtener resultados. Esto es algo a tener en cuenta, ya que el método es usado *in situ*. De este modo, se consiguen resultados inmediatos para verificar la exactitud de medidas y permitir tomar decisiones en el mismo lugar de ensayo.

### **12. Disposición del equipo de medida.**

Ya que el método se usa en carreteras con tráfico, la medida no debe causar ninguna interferencia con las actividades normales de este entorno. El equipo de medida ocupa un espacio mínimo y las actividades normales no deben detenerse



durante la medida o, al menos, no durante mucho tiempo (**véase también facilidad de aplicación**).

### **13. Sensibilidad a agentes externos.**

El método es **estable** frente a cambios en la velocidad y dirección del viento, cambios en la temperatura, condiciones climáticas,... . Si el método fuera demasiado sensible a agentes externos, obligaría a los operadores a reducir las distancias altavoz-micrófono-muestra, y entonces se tendría problemas con la frecuencia y los efectos de la forma de la superficie.

### **14. Generación de la señal.**

Debido a su facilidad práctica de uso, se han escogido los **altavoces** como las fuentes adecuadas de sonido, en vez de pistolas, dispositivos neumáticos o mecánicos.

## **1.1.4. ESTRUCTURA DEL PROGRAMA DESARROLLADO.**

### 1.1.4.1. INTRODUCCIÓN.

El proyecto tiene como objetivo implementar el método “Adrienne” para la obtención de la pérdida por reflexión en un programa informático que realice rápidamente los complejos cálculos matemáticos y que sea de fácil e intuitivo manejo para el usuario.

Se ha considerado adecuado realizar el programa en **Matlab** por tres razones:

1. Se trata de un lenguaje de programación muy potente en el campo de los cálculos matemáticos. Muchas de las complejas funciones matemáticas vienen ya implementadas en el propio lenguaje.
2. Usando los llamados **G.U.I.** (Graphic User’s Interface), interfaces gráficas entre el sistema informático y el usuario, se consigue que la introducción de datos sea un proceso intuitivo. Asimismo, los resultados se presentan de una forma clara y vistosa.
3. El lenguaje de programación en Matlab se está convirtiendo en el “**estándar**” en el mundo científico. La mayoría de los programas escritos en el entorno científico europeo se hallan realizados en Matlab. Por ejemplo, todo el material informático manejado por el equipo investigador europeo del proyecto Adrienne se halla escrito en Matlab.

El programa está estructurado por **funciones**. Se distinguen dos clases de funciones:

1. Funciones llamadas “**pantallas gráficas**”, que a su vez pueden ser de dos tipos:

- Pantallas gráficas de introducción de datos, donde el usuario alimenta al programa con los datos necesarios para realizar los cálculos. Estos datos se introducen de forma intuitiva mediante el uso de ratón y teclado.
- Pantallas gráficas de salida de datos, donde se presentan los resultados obtenidos mediante representaciones gráficas.

2. Funciones de **cálculo** matemático, donde se realizan las operaciones indicadas en el método Adrienne. El usuario no “tiene contacto” con estas funciones. Los datos necesarios para que se realicen los cálculos se introducen vía pantallas gráficas. Los resultados de los cálculos también son expuestos a través de pantallas gráficas.

Hay que tener en cuenta que el código de halla perfectamente comentado, como se puede observar en el **anejo “1.3.1. LISTADO DEL PROGRAMA”**, previendo el caso de que alguien quiera modificarlo a posteriori. Esta circunstancia es bastante probable ya que todavía no existe una redacción definitiva de la norma **EN-1793**, debido a los problemas que hay con las barreras antirruído volumétricas (no planas) y con la mínima frecuencia por debajo de la cual los resultados de la pérdida por reflexión (**RL**) no son fiables (todavía es superior al objetivo que se tenía de 100Hz, como se indica en el apartado "**1.1.3.2. MÉTODO ADRIENNE**" )

Los **programas** creados en Matlab tiene extensión **\*.m**. Las **variables** que contienen datos tienen extensión **\*.mat**. A continuación se pasa a describir cada una de estas funciones y las variables de datos necesarias.

### **1.1.4.2. FUNCIONES DE INTRODUCCIÓN DE DATOS.**

Estas funciones tienen como objetivo adquirir los datos necesarios para calcular la pérdida por reflexión (**RL**) de la barrera antirruído.

Todo el proceso de adquisición de datos se hace a través de **G.U.I.**, interfaces gráficas, que resultan cómodas e intuitivas.

Dentro de las siguientes funciones hay algunas destinadas a que el usuario compruebe que los datos han sido correctamente introducidos. De este modo, se puede asegurar que los cálculos serán representativos de las propiedades absorbentes de la barrera antirruído.

### **1.1.4.2.1 adrienne.m**

No es una función, sino un **programa \*.m**. Lo único que hace es cargar el sonido que se oirá al apretar los botones y llama a la primera función. Luego, las funciones se irán llamando entre sí.

El usuario teclea “**adrienne**” en la “**Command Window**” de Matlab y se inicia el programa. Carga el sonido que se oirá al apretar los pushbuttons y llama a la pantalla de presentación.

### **1.1.4.2.2. portada.m (pantalla gráfica)**

*[] = portada ()*

Pantalla gráfica de presentación del programa, que aparece con música de presentación. En ella aparece el título del proyecto, el autor, la institución donde se ha realizado. Cuando se pulsa el botón “SEGUIR”, se cierra esta pantalla gráfica y se abre la siguiente, r11.

- No tiene parámetros de entrada.
- No tiene parámetros de salida.

### **1.1.4.2.3. r11.m (pantalla gráfica)**

$$[] = r11 ()$$

Pantalla gráfica donde se introducen los siguientes datos necesarios para el cálculo de la pérdida por reflexión:

- directa, señal directa medida.
- n, número de puntos que vamos a estudiar en la barrera.
- dt, periodo de muestreo, en ms.
- 2\*dSE, altura de la barrera, en m.
- dMS, distancia entre el micrófono y la barrera, en m.

- No tiene parámetros de entrada, al ser la primera pantalla gráfica de introducción de datos.
- No tiene parámetros de salida, pero en la memoria general del Matlab quedan las siguientes variables:

*directa*

*n*

*nom\_dir*, variable necesaria para la siguiente pantalla gráfica, r11a. Es el nombre de la variable que contiene la señal directa.

*dt*

*dSE*

*dMS*



#### **1.1.4.2.4. fila.m**

*[fila] = fila (vector)*

Es una subfunción de las funciones **rl1**, **rl1a** y **rl2**.

Devuelve un vector en formato fila, independientemente de si se ha introducido como fila o como columna. Esta función se aplica siempre que el usuario introduce una señal, directa o reflejada. De este modo conseguimos que el programa sea insensible a la introducción de las señales como fila o como columna. Esto también representa comodidad para el usuario ya que no tiene que estar preocupado por meter las señales con un determinado formato.

□ Parámetro de entrada:

*vector*, vector que puede ser introducido en formato fila o columna.

□ Parámetro de salida:

*fila*, vector que contiene la misma información que 'vector', pero en formato fila.

### **1.1.4.2.5. r11a.m (pantalla gráfica)**

$[] = r11a(nom\_dir, directa, n, dt, dSE, dMS)$

Es la misma pantalla gráfica que la anterior, r11, pero con la diferencia de que en esta también se introduce la distancia entre el altavoz y el micrófono. Se informa en esta pantalla de la distancia entre altavoz y micrófono ideal para una óptima mínima frecuencia fiable. Para el cálculo de esta distancia se usa la función **alt\_mic.m**.

□ Parámetros de entrada:

*nom\_dir*, nombre de la variable que contiene la señal directa.

*directa*, señal directa medida.

*n*, número de puntos que vamos a estudiar en la barrera.

*dt*, periodo de muestreo, en ms.

$2*dSE$ , altura de la barrera, en m.

*dMS*, distancia entre el micrófono y la barrera, en m.

□ No tiene parámetros de salida, pero en la memoria general del Matlab quedan las siguientes variables:

*directa*

*n*

*dt*

*dSE*

*dMS*

*dLM*, distancia entre altavoz y micrófono.

*i*, variable necesaria para la siguiente pantalla gráfica, r12.

**NOTA:** A pesar de haber introducido las variables *directa*, *n*, *dt*, *dSE* y *dMS* en la pantalla gráfica anterior, se prevé que esta pantalla el usuario pueda querer

cambiarlas. Por eso, esta función tiene tanto de argumentos de entrada como de salida estas variables.

### 1.1.4.2.6. alt mic.m

$$\text{alt\_mic} = \text{alt\_mic} (dSE, dMS)$$

Es una subfunción de la función **rlla**.

Cálculo de la distancia altavoz-micrófono óptima para el método Adrienne. Es la distancia que nos va a dar la frecuencia mínima fiable más baja fiable. Se calcula según la fórmula dada por **Pedro Cobo (1999)** (Instituto de Acústica, C.S.I.C.) en el documento correspondiente al proyecto “Adrienne”:

“Some calculations about the lowest reliable frequency.”

□ Parámetros de entrada:

*dSE*, semialtura de la barrera, en metros.

*dMS*, distancia micrófono-barrera, en m.

□ Parámetro de salida:

*dLM*, distancia altavoz-micrófono óptima, en m.

### **1.1.4.2.7. rl2.m (pantalla gráfica)**

$$[] = rl2(n, i);$$

Pantalla gráfica donde se introducen las distintas señales reflejadas medidas, en función del punto de estudio y del ángulo de incidencia. En cada punto característico se estudia la señal reflejada para ángulos de incidencia de 50°, 60°, 70°, 80°, 90°, 100°, 110°, 120° y 130°.

□ Parámetros de entrada:

*n*, número de puntos que vamos a estudiar en la barrera.

*i*, variable necesaria para la siguiente pantalla gráfica, rl2. Es el punto "actual" en el que se está introduciendo las señales.

□ No tiene parámetros de salida, pero en la memoria general del Matlab queda la siguiente variable:

*matr\_ref*, matriz que contiene, en filas, cada una de las señales reflejadas medidas.

### **1.1.4.2.8. ecograma.m (pantalla gráfica)**

$[] = \text{ecograma}(\text{matr\_ref}, i, dt, n);$

Esta función tiene como objetivo que el usuario pueda comprobar que las variables han sido correctamente introducidas. Si el usuario sospecha que puede haber cualquier error tiene la opción de volver a introducir los datos.

Pantalla gráfica donde aparece el ecograma de las señales introducidas, para un punto, en  $r12$ . De este modo se puede comprobar si nos hemos confundido al introducirlas. Aparecen, para un mismo punto, las señales medidas para los ángulos de incidencia:  $50^\circ$ ,  $60^\circ$ ,  $70^\circ$ ,  $80^\circ$ ,  $90^\circ$ ,  $100^\circ$ ,  $110^\circ$ ,  $120^\circ$  y  $130^\circ$ .

Tiene como subfunción a **ecog**.

□ Parámetros de entrada:

$\text{matr\_ref}$ , matriz que contiene las señales medidas.

$i$ , punto donde se han realizado las medidas.

$dt$ , periodo de muestreo, en ms.

$n$ , número de puntos característicos de la barrera acústica.

□ No tiene parámetros de salida.

### **1.1.4.2.9. ecog.m**

$$[] = \text{ecog}(A, dt, i1, i2, solp)$$

Es una subfunción de **ecograma**.

Dada la matriz de datos  $A(m,n)$  dibuja el ecograma(para las columnas) en el intervalo  $(i1,i2)$  ,con un solapamiento dado por  $solp$  (en tanto por ciento);  $dt$  es el periodo de muestreo (ms).

□ Parámetros de entrada:

$A$ , matriz de datos.

$dt$ , periodo de muestreo, en ms.

$i1$ , fila inicial de la columna que queremos dibujar.

$i2$ , fila final de la columna que queremos dibujar.

$solp$ , solapamiento, en tanto por ciento.

□ No tiene parámetros de salida.

### **1.1.4.2.10. espere.m**

*espere()*

Pantalla gráfica en la que aparece el mensaje "Calculando. Espere, por favor" Aparece mientras se están ejecutando los cálculos para la obtención de la pérdida acústica por reflexión (RL).

- No tiene parámetros de entrada.
  
- No tiene parámetros de salida.



### 1.1.4.3. FUNCIONES DE CÁLCULO.

Estas funciones son las que realmente implementan el método Adrienne descrito en el apartado "1.1.3.2. MÉTODO ADRIENNE".

**nucleo** es la función principal de cálculo. El resto de las funciones que aparecen en este apartado son subfunciones de **nucleo**.

### **1.1.4.3.1. nucleo.m**

$[ref\_loss, RL5090, RL50130, RL90130, MinFrec] = nucleo (matr\_ref, directa, dSE, dMS, dLM, dt);$

Calcula la pérdida por reflexión de nuestro dispositivo antirruído aplicando el método Adrienne.

□ Parámetros de entrada:

*matr\_ref*; matriz que contiene, en filas, cada una de las señales reflejadas medidas para un punto determinado, en ángulo de incidencia determinado.

*directa*; vector fila que contiene la señal directa.

*dSE*; semialtura de la barrera, en m.

*dMS*; distancia entre el micrófono y la barrera, en m.

*dLM*; distancia entre el altavoz y el micrófono, en m.

*dt*; periodo de muestreo en ms.

□ Parámetros de salida:

*ref\_loss*; pérdida por reflexión para cada una de las frecuencias dadas por las siguientes 18 bandas separadas un tercio de octava: [100 125 160 200 250 315 400 500 630 800 1000 1250 1600 2000 2500 3150 4000 5000] Hz.

*RL5090*; promedio de la pérdida por reflexión (RL), para las distintas bandas de tercio de octava, para los ángulos comprendidos entre 50 y 90 grados.

*RL50130*; promedio de la pérdida por reflexión (RL), para las distintas bandas de tercio de octava, para los ángulos comprendidos entre 50 y 130 grados.

*RL90130*; promedio de la pérdida por reflexión (RL), para las distintas bandas de tercio de octava, para los ángulos comprendidos entre 90 y 130 grados.

*MinFrec*; mínima frecuencia para la cual los valores obtenidos de la pérdida por reflexión son válidos.

### 1.1.4.3.2. alinear.m

$$Ad = \text{alinear}(A,B)$$

Es una subfunción de **nucleo**.

Dados dos vectores A y B, desplaza el vector A, de modo que los máximos de los dos vectores se dan para la misma posición dentro del vector. Los vectores A y B han de tener el mismo número de datos.

□ Parámetros de entrada:

*A.*

*B.*

□ Parámetros de salida:

*Ad*, vector A desplazado.

### 1.1.4.3.3. tdel.m

$$[tdel, taud, taur] = tdel (dSE, dLM, dMS);$$

Es una subfunción de **nucleo**.

Halla la longitud temporal de la ventana Adrienne, así como el tiempo, en ms, que tarda en llegar la onda reflejada en incidencia normal. Estos parámetros se calculan a partir de los parámetros geométricos del dispositivo antirruído. Para calcular estos parámetros se usan las fórmulas que aparecen en el apartado **tal**.

□ Parámetros de entrada:

*dSE*, semialtura de la barrera acústica, en m.

*dLM*, distancia altavoz-micrófono, en m.

*dMS*, distancia micrófono-barrera, en m.

□ Parámetros de salida:

*tdel*, longitud temporal de la ventana Adrienne, en ms.

*taud*, tiempo que tarda en llegar la onda directa, en ms.

*taur*, tiempo que tarda en llegar la onda reflejada en incidencia normal, en ms.

#### 1.1.4.3.4. wadri.m

$w = \text{wadri}(t_{del}, dt)$

Es una subfunción de **nucleo**.

Cálculo de la ventana Adrienne, *wadri*, a partir de los datos del periodo de muestreo, *dt* (en ms), y retardo temporal, *t<sub>del</sub>* (en ms)

□ Parámetros de entrada:

*t<sub>del</sub>*, longitud temporal de la ventana Adrienne, en ms.

*dt*, periodo de muestreo, en ms.

□ Parámetros de salida:

*w*, vector fila que contiene la ventana Adrienne.

### 1.1.4.3.5. nutwin.m

$x = \text{nutwin}(nw, iw)$

Es una subfunción de **wadri**.

Calcula una ventana de Nuttall de  $nw$  puntos definida por:

$$x(t) = a_0 + a_1 \cos(\pi 2t/T) + a_2 \cos(2\pi 2t/T) + a_3 \cos(3\pi 2t/T),$$

donde  $|t| > T/2$ , siendo  $nw \cdot dt = T$ , la longitud de la ventana.

Los coeficientes  $\{a_0, a_1, a_2, a_3\}$  dependen del tipo de ventana

$iw$	$a_0$	$a_1$	$a_2$	$a_3$
1 (RECT)	1	0	0	0
2 (HANN)	0.5	0.5	0	0
3 (HAMM)	0.53836	0.461614	0	0
4 (BLACK)	0.42	0.5	0.08	0
5 (NUT98)	0.3635819	0.48911775	0.1365995	0.00106411

Es una función desarrollada por Pedro Cobo (Instituto de Acústica)

□ Parámetros de entrada:

$nw$ , número de puntos de la ventana.

$iw$ , tipo de ventana.

□ Parámetros de salida:

$x$ , ventana temporal.

### **1.1.4.3.6. ventana.m**

*ventana* = *ventana(vent,tau,num\_datos,dt)*;

Es una subfunción de **nucleo**.

Sitúa la ventana obtenida por el método Adrienne 0.5 ms antes del tiempo dado por tau, en un vector con tantos elementos como num\_datos. El resto de los elementos serán ceros. De este modo, se recogerá el evento que comienza en tau y el vector tendrá la misma longitud que las señales medidas, de modo que dichos vectores se podrán multiplicar.

□ Parámetros de entrada:

*vent*, ventana obtenida por el método Adrienne.

*tau*, tiempo de origen del evento que se quiere que recoja la ventana Adrienne.

*num\_datos*, longitud del vector que contiene la señal.

*dt*, periodo de muestreo, en ms.

□ Parámetros de salida:

*ventana*, vector que contiene la ventana Adrienne en el lugar adecuado para recoger el evento que tiene su origen en tau.

### 1.1.4.3.7. bandas.m

*tercio\_oct = bandas (banda\_estr,delta)*

Es una subfunción de **nucleo**.

Pasa una función de la frecuencia en banda estrecha a su equivalente en bandas de tercio de octava, desde 100Hz hasta 5000Hz. Implementa el algoritmo desarrollado por Garai (1997).

□ Parámetro de entrada:

*banda\_estr*, función de la frecuencia en banda estrecha.

*delta*, incremento de frecuencia entre dos puntos de la función, en banda estrecha.

□ Parámetro de salida:

*tercio\_oct*, función equivalente en 18 bandas de tercio de octava de valores nominales [100, 125, 160, 200, 250, 315, 400, 500, 630, 800, 1000, 1250, 1600, 2000, 2500, 3150, 4000, 5000] Hz; los valores reales de las bandas de tercio de octava son [100 125.8925 158.4893 199.5262 251.1886 316.2278 398.1072 501.1872 630.9573 794.3282 1000 1258.9254 1584.8932 1995.2623 2511.8864 3162.2777 3981.0717 5011.8723] Hz;



#### 1.1.4.4. FUNCIONES DE SALIDA DE RESULTADOS.

El programa da como resultado 4 pantallas gráficas, donde el usuario puede extraer conclusiones. Estas 4 pantallas gráficas son:

- **ventaneo**, donde se puede comprobar si el ventaneo ha sido correctamente realizado.
- **graf\_con**, donde se comparan las pérdidas por reflexión para los tres promedios angulares.
- **graf\_bar**, donde se obtienen los valores de la pérdida por reflexión, que serán usados para obtener las propiedades absorbentes de la barrera, según EN-1793.
- **min\_frec**, donde se obtiene la mínima frecuencia por debajo de la cual los resultados no son fiables.

### **1.1.4.4.1. ventaneo.m (pantalla gráfica)**

*ventaneo (i,directa,vent\_dir,reflejada,vent\_ref,v\_tiempo,tiempos);*

Pantalla gráfica donde se representan las señales con sus correspondientes ventanas superpuestas. El objetivo es que se compruebe que se ha hecho el ventaneo correctamente.

□ Parámetros de entrada:

*i*, número que nos da la iteración de la función nucleo.m.

*directa*, señal directa medida.

*vent\_dir*, ventana de la señal directa.

*reflejada*, señal reflejada medida(distinta en cada iteración).

*vent\_ref*, ventana de la señal reflejada.

*v\_tiempo*, vector que contiene el tiempo transcurrido correspondiente a cada punto de las señales medidas.

*tiempos*, vector que contiene variables temporales necesarias.

□ No tiene parámetros de salida.

### **1.1.4.4.2. graf con.m**

*graf\_con(RL5090,RL90130,RL50130)*

Realiza un gráfico en representación continua, comparando los tres promedios de la pérdida por reflexión(RL).

□ Parámetros de entrada:

*RL5090*; promedio de la pérdida por reflexión (RL), para las distintas bandas de tercio de octava, para los ángulos comprendidos entre 50 y 90 grados.

*RL50130*; promedio de la pérdida por reflexión (RL), para las distintas bandas de tercio de octava, para los ángulos comprendidos entre 50 y 130 grados.

*RL90130*; promedio de la pérdida por reflexión (RL), para las distintas bandas de tercio de octava, para los ángulos comprendidos entre 90 y 130 grados.

□ No tiene parámetros de salida.

### **1.1.4.4.3. graf\_bar.m (pantalla gráfica)**

*[] = graf\_bar (RL5090,RL90130,RL50130);*

Pantalla gráfica en la que aparecen representados los promedios de las pérdidas por reflexión, para los distintos rangos angulares de incidencia. Se representan en forma de gráficos de barras. Encima de cada barra aparece el valor de la pérdida por reflexión para cada tercio de octava de cada promedio angular.

□ Parámetros de entrada:

*RL5090*; promedio de la pérdida por reflexión (RL), para las distintas bandas de tercio de octava, para los ángulos comprendidos entre 50 y 90 grados.

*RL50130*; promedio de la pérdida por reflexión (RL), para las distintas bandas de tercio de octava, para los ángulos comprendidos entre 50 y 130 grados.

*RL90130*; promedio de la pérdida por reflexión (RL), para las distintas bandas de tercio de octava, para los ángulos comprendidos entre 90 y 130 grados.

□ No tiene parámetros de salida.

#### **1.1.4.4.4. min frec.m**

$min\_frec = min\_frec(vent, dt);$

Pantalla gráfica donde se calcula la mínima frecuencia por debajo de la cual los resultados obtenidos de la pérdida por reflexión (RL) no son fiables. Se calcula obteniendo el primer mínimo en la transformada de Fourier de la ventana Adrienne.

□ Parámetros de entrada:

$vent$ , ventana Adrienne.

$dt$ , periodo de muestreo, en ms.

□ Parámetro de salida:

$min\_frec$ , mínima frecuencia fiable, en Hz.

## **1.1.5. COSTE TOTAL.**

Como resultado de este proyecto se ha desarrollado un programa informático que calcula la pérdida acústica por reflexión de barreras antirruido por el método Adrienne.

El coste total de este proyecto es de tres millones doscientas veintisiete mil setecientas (3.227.700) pesetas.

Madrid, 4 de junio de 1999.



Adolfo Corral Quintana.

## **1.1.6. REFERENCIAS.**



- Cobo, P. and Santiago, J.S.**, 1997. "Reflection and transmission loss measurements on in situ samples using the Adrienne Method ". Doc. ADR5-1, Project MAT1-CT94049, EC Measurement and testing Programme.
- Cobo, P. and Santiago, J.S.**, 1997. "Sound absorption measurements on flat and non flat samples using the Adrienne Method (MLS with signal subtraction technique)". Doc. ADR5-1, Project MAT1-CT94049, EC Measurement and testing Programme.
- Cobo, P.**, 1999. "Some calculations concerning the Adrienne setup and the lowest reliable frequency". Project MAT1-CT94049, EC Measurement and testing Programme.
- Gade, S. and Herlufsen, H.**, 1987. "Windows to FFT analysis". B&K Technical Review, N° 3 and 4.
- Garai, M.**, 1993. "Measurement of the sound-absorption coefficient in situ: The reflection method using periodic pseudorandom sequences of maximum length". Applied Acoustics, 39, 119-139.
- Garai, M.**, 1995. "The state of the art of sound absorption measurements in situ: 1. Echo-impulse methods. 2. Sound intensity methods. ". Doc. ADR4-7, Project MAT1-CT94049, EC Measurement and testing Programme.
- Garai, M.**, 1996. "Algorithm for the reconstruction of sound absorption coefficient values in one third octave bands from narrow band data – Revised version". Doc. ADR4-39, Project MAT1-CT94049, EC Measurement and testing Programme.

**Garai, M.**, 1997. "First definition of a procedure for sound index measurement using the MLS method". Doc. ADR4-56, Project MAT1-CT94049, EC Measurement and testing Programme.

**Klinger, H. H.**, 1969. "Técnica de la acústica". Ediciones técnicas Marcombo, Barcelona.

**Linares, J.I.**, 1997. "Apuntes de medida de sonido". Ediciones ICAI.

**Nuttal, A.**, 1981. "Some windows with very good sidelobe behaviour". IEEE ASSP Trans., ASSP-29(1), 84-91.

**Piroux, H.**, 1967. "Diccionario general de Acústica y Electroacústica". Editorial Paraninfo, Madrid.

**Randall, R. B.**, 1977. "Application of the B & K Equipment to frequency analysis".

**Stephens, R.W.B.**, 1974. "Sound". Crosby Lockwood Staples, London.

**The Adrienne Research Team.** 1998. "Test methods for the acoustic performance of road traffic noise reducing devices". Final Report, Project MAT1-CT94049, EC Measurement and testing Programme.

**Math Works inc.**, 1993. "Building a Graphical User Interface". MATLAB, versión 4.

**Math Works inc.**, 1997. "Building GUIs with Matlab". MATLAB, versión 5.

**Cobo, P. and Santiago, J.S.**, 1997. "Reflection and transmission loss measurements on in situ samples using the Adrienne Method ". Doc. ADR5-1, Project MAT1-CT94049, EC Measurement and testing Programme.

**Cobo, P. and Santiago, J.S.**, 1997. "Sound absorption measurements on flat and non flat samples using the Adrienne Method (MLS with signal subtraction technique)". Doc. ADR5-1, Project MAT1-CT94049, EC Measurement and testing Programme.

**Cobo, P.**, 1999. "Some calculations concerning the Adrienne setup and the lowest reliable frequency". Project MAT1-CT94049, EC Measurement and testing Programme.

**Gade, S. and Herlufsen, H.**, 1987. "Windows to FFT analysis". B&K Technical Review, N° 3 and 4.

**Garai, M.**, 1993. "Measurement of the sound-absorption coefficient in situ: The reflection method using periodic pseudorandom sequences of maximum length". Applied Acoustics, 39, 119-139.

**Garai, M.**, 1995. "The state of the art of sound absorption measurements in situ: 1. Echo-impulse methods. 2. Sound intensity methods. ". Doc. ADR4-7, Project MAT1-CT94049, EC Measurement and testing Programme.

**Garai, M.**, 1996. "Algorithm for the reconstruction of sound absorption coefficient values in one third octave bands from narrow band data – Revised version". Doc. ADR4-39, Project MAT1-CT94049, EC Measurement and testing Programme.

**Garai, M.**, 1997. "First definition of a procedure for sound index measurement using the MLS method". Doc. ADR4-56, Project MAT1-CT94049, EC Measurement and testing Programme.

**Klinger, H. H.**, 1969. "Técnica de la acústica". Ediciones técnicas Marcombo, Barcelona.

**Linares, J.I.**, 1997. "Apuntes de medida de sonido". Ediciones ICAI.

**Nuttal, A.**, 1981. "Some windows with very good sidelobe behaviour". IEEE ASSP Trans., ASSP-29(1), 84-91.

**Piriaux, H.**, 1967. "Diccionario general de Acústica y Electroacústica". Editorial Paraninfo, Madrid.

**Randall, R. B.**, 1977. "Application of the B & K Equipment to frequency analysis".

**Stephens, R.W.B.**, 1974. "Sound". Crosby Lockwood Staples, London.

**The Adrienne Research Team.** 1998. "Test methods for the acoustic performance of road traffic noise reducing devices". Final Report, Project MAT1-CT94049, EC Measurement and testing Programme.

**Math Works inc.**, 1993. "Building a Graphical User Interface". MATLAB, versión 4.

**Math Works inc.**, 1997. "Building GUIs with Matlab". MATLAB, versión 5.

## **1.2. CÁLCULOS.**

## **1.2.0. ÍNDICE.**

**1.2.0. ÍNDICE.....83**

**1.2.1. CÁLCULOS CONCERNIENTES A LOS PARÁMETROS GEOMÉTRICOS ASOCIADOS A LA MEDIDA.....84**

1.2.1.1. TIEMPOS DE LLEGADA DE LOS DISTINTOS PULSOS DE PRESIÓN AL MICRÓFONO.....84

1.2.1.2. CONCLUSIONES.....90

**1.2.2. MEDIDAS EXPERIMENTALES.....97**

1.2.2.1. INTRODUCCIÓN.....97

1.2.2.2. ENSAYO EN LABORATORIO.....98

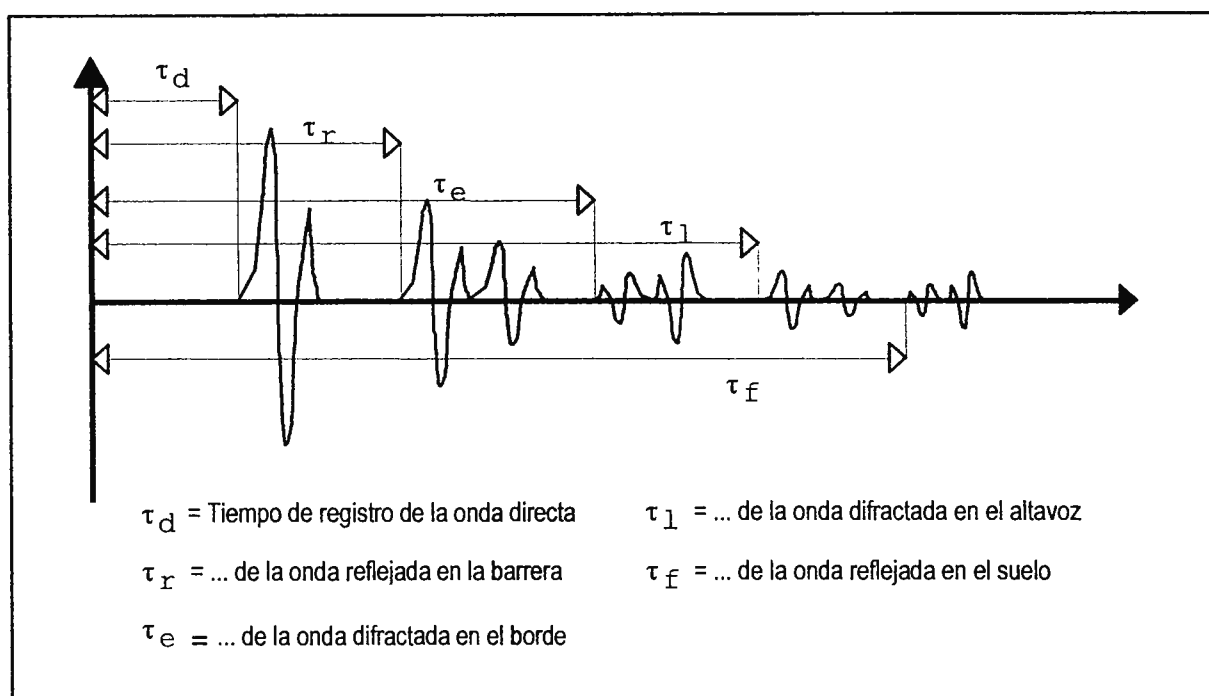
1.2.2.2. ENSAYO IN SITU.....108

**1.2.1. CÁLCULOS CONCERNIENTES A  
LOS PARÁMETROS GEOMÉTRICOS  
ASOCIADOS A LA MEDIDA.**



### 1.2.1.1. TIEMPOS DE LLEGADA DE LOS DISTINTOS PULSOS DE PRESIÓN AL MICRÓFONO.

La forma de onda típica (de presión acústica frente al tiempo) que es recogida por el altavoz tiene una forma similar a la que aparece en la **figura 1.2.1.1.1.**



**Figura 1.2.1.1.1. Señal típica recibida en el micrófono.**

Primero se recibe un pulso fuerte de presión debida al eco **directo**. Posteriormente, se reciben distintos ecos de amplitud cada vez menor. Estos siguientes ecos pueden ser debido a la **reflexión en la barrera**, a la **reflexión en el suelo**, a la **difracción en el borde** de la barrera o a la **difracción en el propio altavoz**. Se reciben también otras señales de presión, pero ya menor importancia.

Para estudiar los tiempos de llegada de cada uno de estos ecos debemos tener en cuenta los **parámetros geométricos** de la medida, sabiendo que la velocidad de propagación de las ondas acústicas es  $c$  (0.34 m / ms), la velocidad del sonido.

En la **figura 1.2.1.1.2.** aparece un croquis con los parámetros geométricos asociados a la medida: tanto de la barrera antirruído, como del equipo de medida.

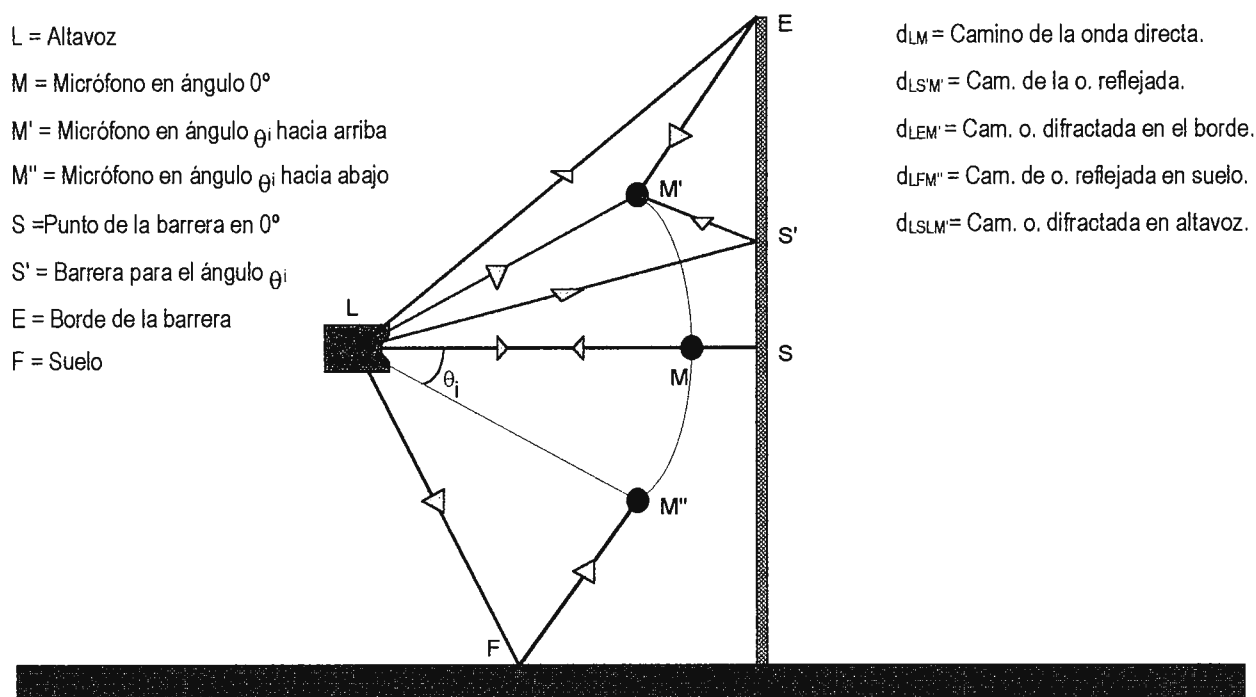


Figura 1.2.1.1.2. Disposición geométrica de medida en el método Adrienne;

Si se calcula el camino recorrido por cada uno de los pulsos de presión, en función de los parámetros geométricos asociados al ensayo, y luego se divide por la velocidad del sonido entonces se calcularán los distintos tiempos de registro de cada uno de los pulsos de presión.

Este proceso se realiza, a continuación, para cada uno de los pulsos de presión definidos en la figura 1.2.1.1.1.

### 1) Tiempo de registro de la onda directa.

$$\tau_d = d_{LM} / c$$

(1)

**2) Tiempo de registro de la onda difractada en el altavoz.**

$$\tau_i = \frac{1}{c} (2d_{LS} + d_{LM}) = \frac{1}{c} (3d_{LM} + 2d_{MS}) \quad (2)$$

**3) Tiempo de registro de la onda difractada en el borde de la barrera:**

$$\tau_e = \frac{1}{c} (d_{LE} + d_{EM'}) \quad (3a)$$

En el triángulo LSE se cumple:

$$d_{LE} = \sqrt{d_{LS}^2 + d_{ES}^2} \quad (3b)$$

Aplicando el teorema del coseno al triángulo LM'E:

$$d_{EM'} = \sqrt{d_{LE}^2 + d_{LM}^2 - 2d_{LE}d_{LM} \cos(\alpha + \theta_i)} \quad (3c)$$

$$\text{donde:} \quad \alpha = \text{atan} \left( \frac{d_{ES}}{d_{LS}} \right)$$

De las ecuaciones (3a) y (3c) se obtiene:

$$\tau_e = \frac{1}{c} \left[ \sqrt{d_{LS}^2 + d_{SE}^2} + \sqrt{d_{LS}^2 + d_{SE}^2 + d_{LM}^2 - 2d_{LM} \sqrt{d_{LS}^2 + d_{SE}^2} \cos(\alpha + \theta_i)} \right] \quad (4)$$

**4) Tiempo de registro de la onda reflejada en la barrera:**

Se supone barrera plana, donde:

$$\tau_r = \frac{1}{c} (d_{LS'} + d_{S'M'})$$

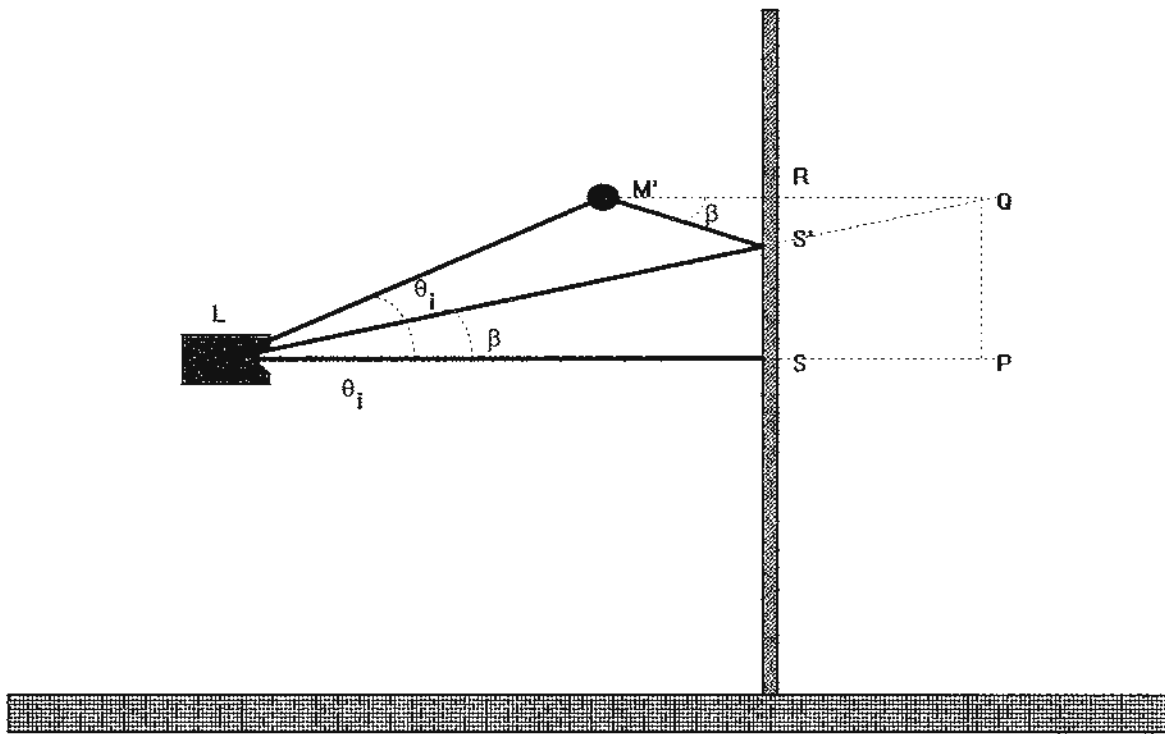


Figura 1.2.1.1.3. Disposición geométrica para la señal reflejada en la barrera en incidencia oblicua.

De la Figura 1.2.1.1.3. se deduce:

$$(d_{LS'} + d_{S'M'})^2 = (d_{LS} + d_{RM'})^2 + (d_{LM} \sin \vartheta_i)^2 \quad (5a)$$

Pero:

$$d_{RM'} = d_{LS} - d_{LM} \cos \vartheta_i \quad (5b)$$

Por tanto:

$$\begin{aligned} d_{LS'} + d_{S'M'} &= \sqrt{(2d_{LS} - d_{LM} \cos \vartheta_i)^2 + (d_{LM} \sin \vartheta_i)^2} \\ &= \sqrt{4d_{LS}^2 + d_{LM}^2 - 4d_{LS}d_{LM} \cos \vartheta_i} \end{aligned} \quad (5c)$$

Finalmente:

$$\tau_r = \frac{1}{c} \sqrt{4d_{LS}^2 + d_{LM}^2 - 4d_{LS}d_{LM} \cos \vartheta_i} \quad (6)$$

### 5) Tiempo de registro de la onda reflejada en el suelo.

Se supone suelo plano.

$$\tau_f = \frac{1}{c} (d_{LF} + d_{FM''})$$

El camino recorrido por la onda reflejada en el suelo es  $d_{LF} + d_{FM''}$ . De la Figura 1.2.1.1.4. se obtiene:

$$(d_{LF} + d_{FM''})^2 = (d_{SE} + d_{F'M''})^2 + (d_{LM} \cos \vartheta_i)^2 \quad (7a)$$

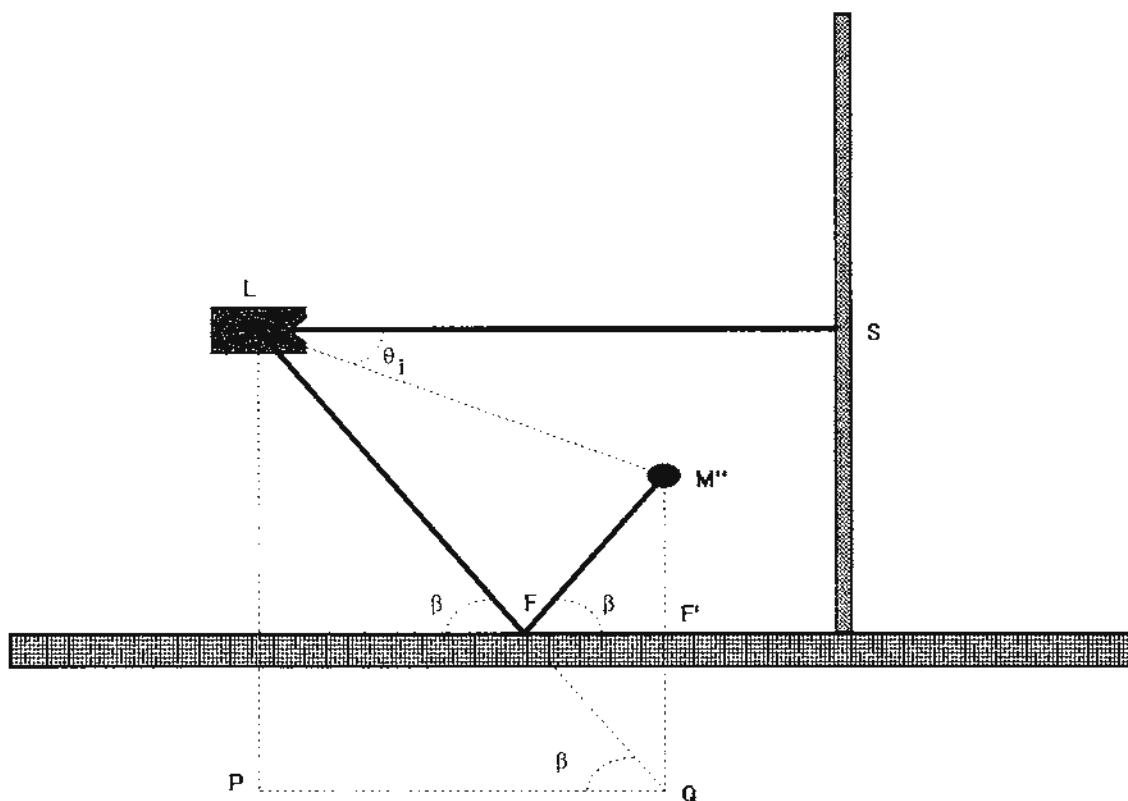


Figura 1.2.1.1.4. Disposición geométrica para la señal reflejada en el suelo en incidencia oblicua.

Pero:

$$(\mathbf{d}_{LS'} + \mathbf{d}_{S'M'})^2 = (\mathbf{d}_{LS} + \mathbf{d}_{RM'})^2 + (\mathbf{d}_{LM} \sin \vartheta_i)^2 \quad (7b)$$

Por tanto:

$$\mathbf{d}_{LF} + \mathbf{d}_{FM''} = \sqrt{4\mathbf{d}_{SE}^2 + \mathbf{d}_{LM}^2 - 4\mathbf{d}_{SE}\mathbf{d}_{LM}\sin\vartheta_i} \quad (7c)$$

Finalmente, el tiempo de registro de la onda reflejada en el suelo es:

$$\tau_f = \frac{1}{c} \sqrt{4\mathbf{d}_{SE}^2 + \mathbf{d}_{LM}^2 - 4\mathbf{d}_{SE}\mathbf{d}_{LM}\sin\vartheta_i} \quad (8)$$

### **1.2.1.2. CONCLUSIONES.**

Se exponen a continuación las **conclusiones** obtenidas a partir del estudio geométrico realizado en el apartado “**1.2.1.1. TIEMPOS DE LLEGADA DE LOS DISTINTOS PULSOS DE PRESIÓN AL MICRÓFONO.**”.

### 1.2.1.2.1. UBICACIÓN DE LA VENTANA PARA EL ECO DIRECTO Y REFLEJADO.

Como se mencionó en el apartado “1.1.3.2.2.2. UBICACIÓN Y TAMAÑO DE LA VENTANA ADRIENNE” la **ventana** para la señal directa y reflejada es la **misma**, pero su **ubicación** es **distinta** debido a la diferencia en los tiempos de registro en el altavoz entre la onda directa y las reflejadas.

Del apartado anterior se obtienen los tiempos de registro para estos dos pulsos de presión, que serán los puntos donde debe empezar la ventana Adrienne. Es decir, los tiempos que se exponen a continuación nos dan el **origen** de la ventana Adrienne (y, por tanto, su ubicación). Se debe tener en cuenta que la ventana empezará 0.5 ms antes de estos tiempos (“pre-ventana Blackman-Harris” explicada en el apartado 1.1.3.2.2.1.)

Para el eco directo:

$$\tau_d = d_{LM} / c$$

Para el eco reflejado:

$$\tau_r = \frac{1}{c} \sqrt{4d_{LS}^2 + d_{LM}^2 - 4d_{LS}d_{LM} \cos \vartheta_i}$$

Hay que tener en cuenta que este tiempo es **variable** con el **ángulo de incidencia**. Se debe tomar el valor de la expresión anterior para el ángulo de incidencia que haga **mínimo** el tiempo de registro de



la onda reflejada en la barrera. Esto se da para  $\cos(\theta) = 1$ , es decir  $\theta = 90^\circ$  (incidencia normal).

Por tanto, el tiempo de origen de la ventana Adrienne para la onda reflejada es:

$$\tau_r(\theta = 90^\circ) = \frac{1}{c} \sqrt{4d_{LS}^2 + d_{LM}^2 - 4d_{LS}d_{LM}}$$

### **1.2.1.2.2. LONGITUD TEMPORAL DE LA VENTANA ADRIENNE (TIEMPO DE RETRASO).**

La ventana Adrienne debe ser **lo más grande posible** (para obtener una menor mínima frecuencia fiable, como se explica en el apartado “**1.1.3.2.4. MÍNIMA FRECUENCIA FIABLE**”), pero no debe recoger más pulsos de presión que el provocado por la reflexión en la barrera bajo ensayo.

Por tanto su longitud viene limitada por el **pulso de presión inmediatamente posterior** al provocado por la reflexión en la barrera. La diferencia entre estos dos tiempos es lo que se denomina “**tiempo de retraso**”. Para evaluar dicha magnitud se debe usar la siguiente expresión:

$$\text{tiempo\_de\_retraso} = \min(\tau_i, \min(\tau_e(\theta)), \min(\tau_r(\theta))) - \tau_r(\theta = 90^\circ)$$

donde las expresiones para los tiempos  $\tau_i$ ,  $\tau_e$ , y  $\tau_r$  aparecen el apartado anterior, el 1.2.1.2.1..

Para hallar la longitud de la ventana Adrienne ( $T_w$ ) hay que sumar al valor del tiempo de retraso los **0.5 ms** de la “preventana” Blackman-Harris.

$$T_w = \text{tiempo de retraso} + 0.5 \quad (\text{ms})$$

La **forma** de la ventana Adrienne quedará definida, finalmente, como aparece en la **figura 1.2.1.2.2.1.**, de la siguiente página.

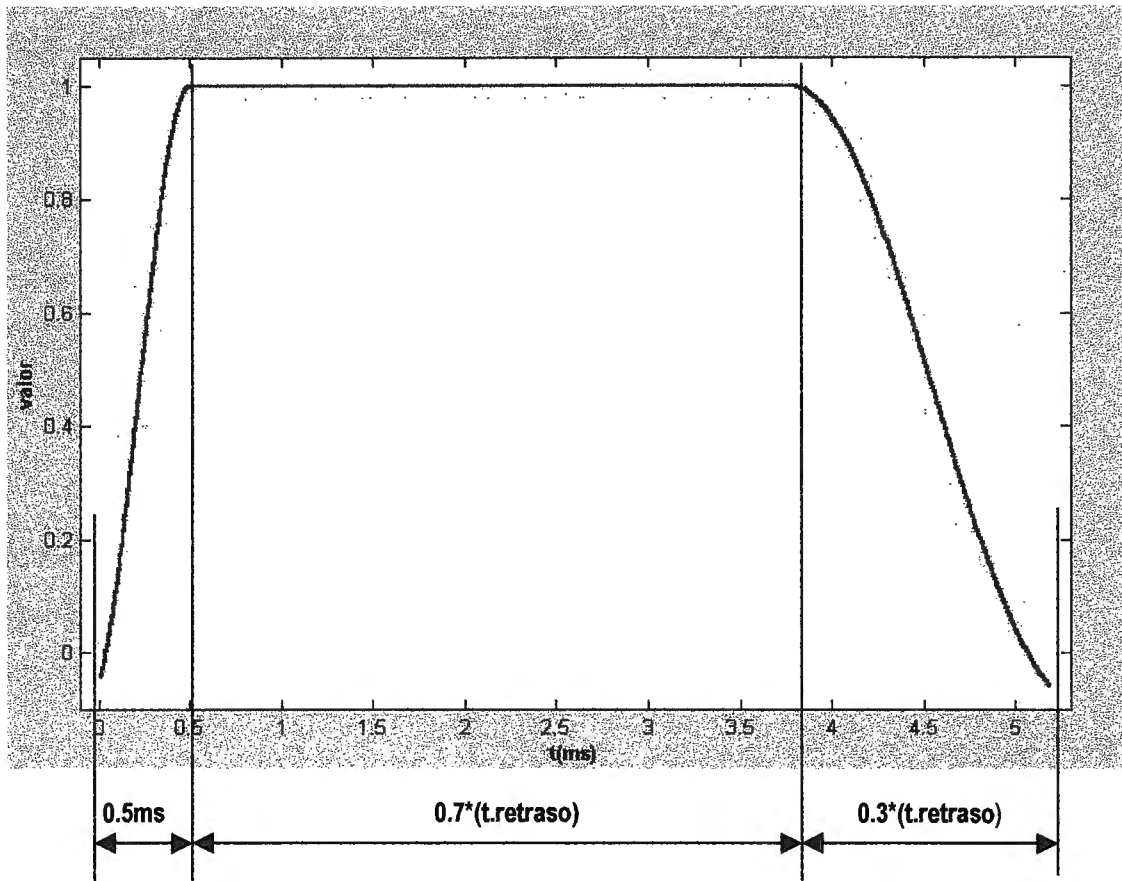


Figura 1.2.1.2.2.1. Forma de la ventana Adrienne, según sus distintos tramos.

### **1.2.1.2.3. PARÁMETROS GEOMÉTRICOS ÓPTIMOS PARA UNA ÓPTIMA MÍNIMA FRECUENCIA FIABLE.**

Existe **una única configuración** geométrica para la cual se obtiene la mayor longitud posible para la ventana Adrienne y, por tanto una **óptima (mínima) mínima** frecuencia por debajo de la cual los resultados obtenidos para la pérdida por reflexión no son fiables.

Se demuestra (*Cobo, 1999*) que para una determinada altura de la barrera ( $d_{SE}$  es la semialtura de la barrera) y para una determinada distancia entre al micrófono y la barrera ( $d_{MS}$ ), la distancia entre el altavoz y el micrófono ( $d_{LM}$ ) para la cual se obtiene la mayor longitud posible para la ventana Adrienne es:

$$d_{LM} = \frac{\sqrt{(12d_{MS} + 4d_{SE}\sin 40^\circ)^2 + 128(d_{SE}^2 - d_{MS}^2)}}{16} - \frac{12d_{MS} + 4d_{SE}\sin 40^\circ}{16}$$

Esto es interesante ya que ajustando la distancia entre el altavoz y el micrófono (lo cual es posible según se describe en el apartado “1.1.3.2.1. TÉCNICA DE SUBSTRACCIÓN DE SEÑAL.”) se puede obtener la mínima frecuencia fiable posible.

Esta distancia ha sido tenida en cuenta en el programa informático de este proyecto de fin de carrera. En él, se informa al usuario de dicha distancia óptima, como se puede apreciar en la siguiente figura.

Figure No. 1

File Edit Window Help

**Pérdida por reflexión (RL).**

Introduce la variable que contiene la señal directa:	directa
Introduce el número de puntos de medida:	3
Introduce el periodo de muestreo, en ms:	0.01325
Introduce la altura de la barrera, en m:	4
Introduce, en m, la distancia entre el micrófono y la barrera:	0.25

Introduce, en m, la distancia entre el altavoz y el micrófono. La distancia óptima calculada es de 0.984 m

1.25

SEGUIR

Figura 1.2.1.2.3. El programa desarrollado en este proyecto de fin de carrera informa al usuario de la distancia óptima, en cuanto a la mínima frecuencia fiable.

## **1.2.2. MEDIDAS EXPERIMENTALES.**

### 1.2.2.1. INTRODUCCIÓN.

Se presentan en este apartado los resultados obtenidos para dos medidas experimentales. El **procesado** de estas señales ha sido realizado por el **programa** que se presenta en este proyecto.

La primera de ellas fue tomada por el equipo investigador del proyecto Adrienne en un ensayo realizado **en laboratorio**. Se hizo para una muestra llamada Beton Bois.

La segunda de ellas fue tomada también por el equipo investigador del proyecto Adrienne, pero en un ensayo realizado **in situ**. Se construyó para la ocasión, en la ciudad de Grenoble, una barrera acústica compuesta por diversos materiales. Esta medida experimental realizada en campo se hizo para un material llamado MICE, que es chapa perforada rellena de un material fonoabsorbente, lana de roca (Rockwool).

A continuación se presentan estos dos ensayos, con sus resultados, más detalladamente.

### 1.2.2.2. ENSAYO EN LABORATORIO.

El equipo investigador del proyecto Adrienne comprobó, en primer lugar, la validez del ensayo que había creado. Para ello, se realizaron ensayos en laboratorio sobre varias muestras de materiales utilizados en barreras acústicas.

Estos ensayos fueron realizados en junio de 1997, en la Universidad de Bolonia. La **disposición geométrica** del equipo de medida puede apreciarse en la siguiente figura.

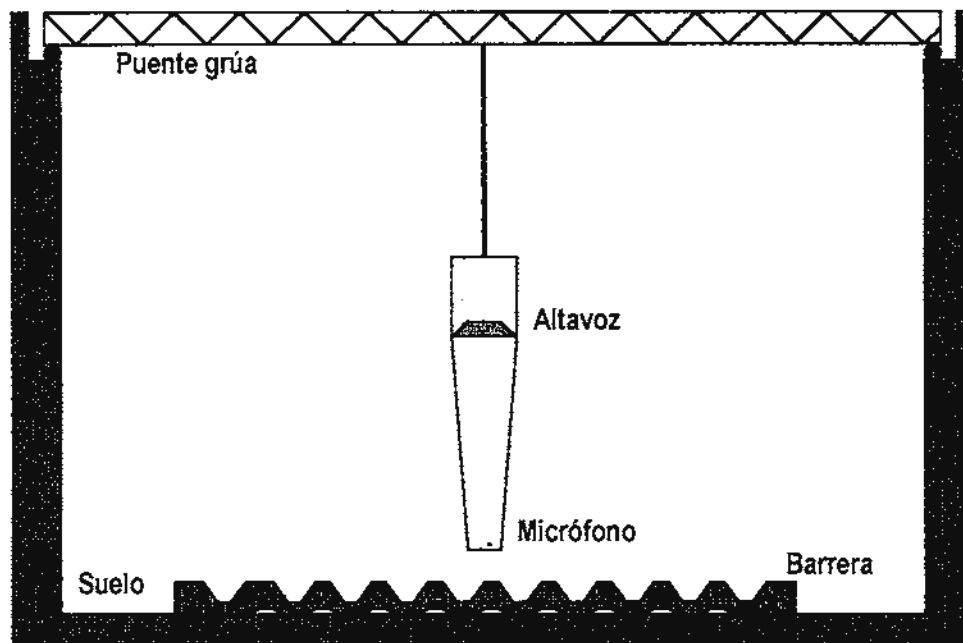


Figura 1.2.2.2.1. Disposición geométrica para el ensayo en laboratorio.

En el ensayo en laboratorio la barrera se dispuso en el **suelo**. El altavoz y el micrófono se podían orientar y mover gracias al carril del puente grúa. Se hicieron las medidas para los nueve ángulos de incidencia observados en el método. Para obtener solamente el eco directo se orientó el equipo hacia donde se encontraba la



pared más lejana, de modo que no se tuvieran reflexiones indeseadas. Esto se puede apreciar en la siguiente figura.

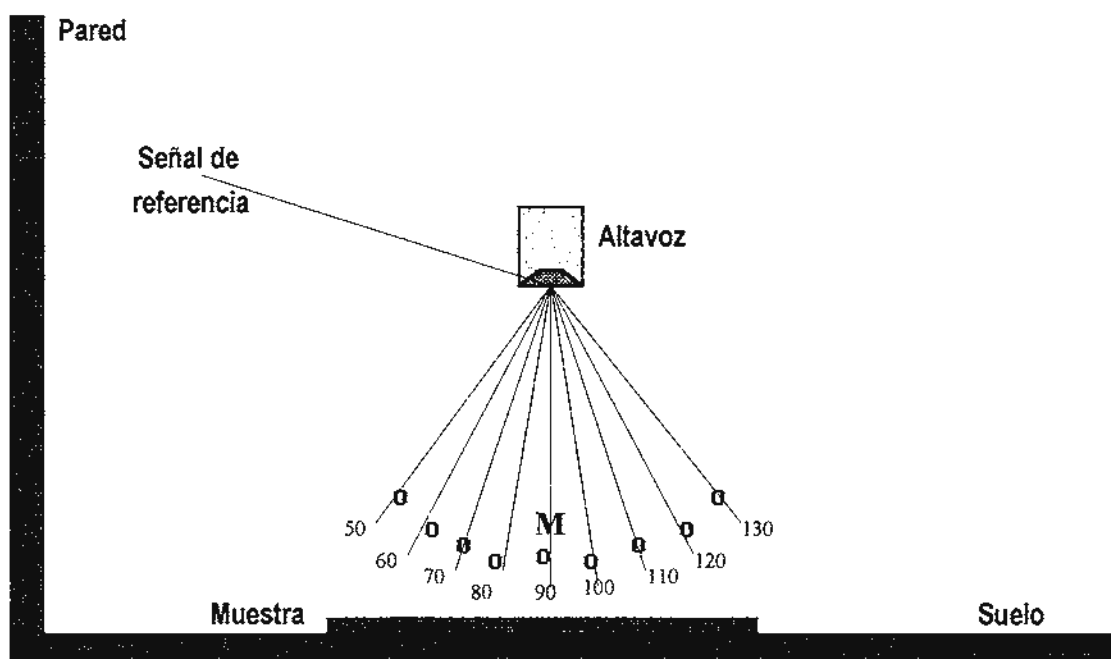


Figura 1.2.2.2. Orientación del equipo de medida para los distintos ángulos de incidencia y para la obtención de la señal de referencia.

Para dar una mejor descripción del proceso de medida aparecen a continuación dos fotografías tomadas en el laboratorio de la Universidad de Bolonia.



Figura 1.2.2.2.3. Orientación del equipo de medida para la obtención de la señal de referencia.

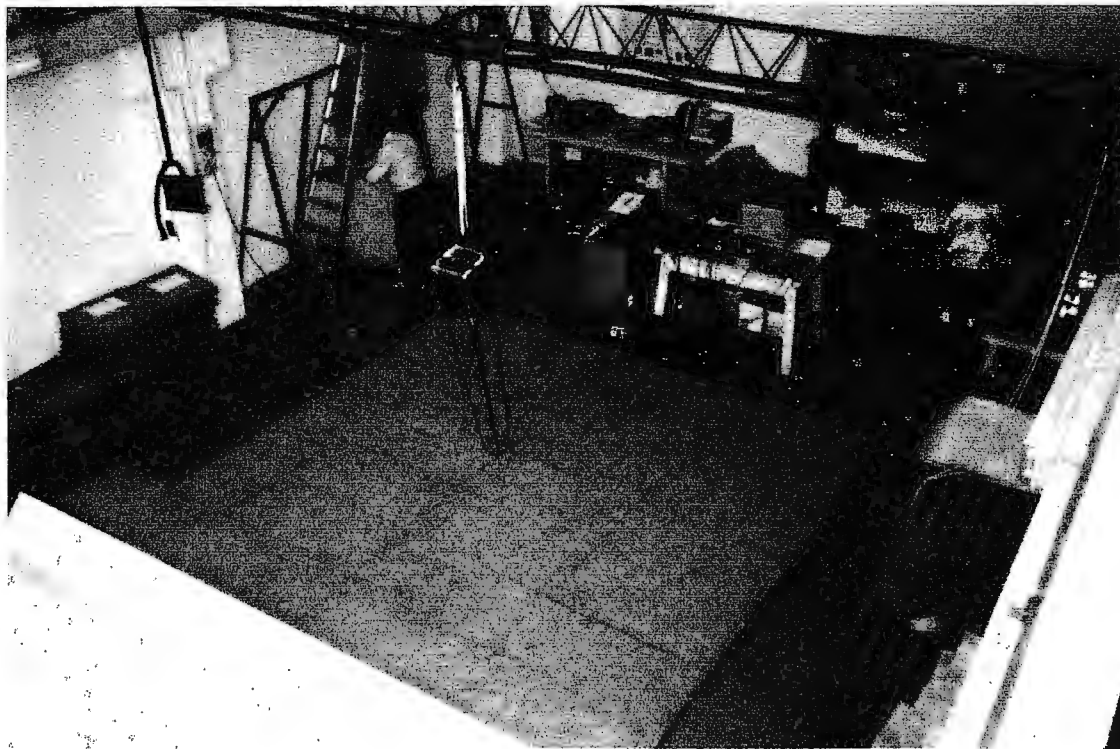


Figura 1.2.2.2.4. Realización de la medida para una muestra absorbente.

Una vez que ya se ha explicado como se realizó el ensayo, nos centramos en las medidas de las señales que, posteriormente, se han procesado con el software de este proyecto de fin de carrera.

La medida escogida para ser procesada por el software de este proyecto de fin de carrera se hizo sobre un material llamado **Beton Bois**. Es un material no plano en el que la pérdida por reflexión se tomó para tres puntos distintos; por tanto para obtener la RL se hace el promedio las RL para cada punto. En la siguiente figura aparecen estos **tres puntos** característicos.

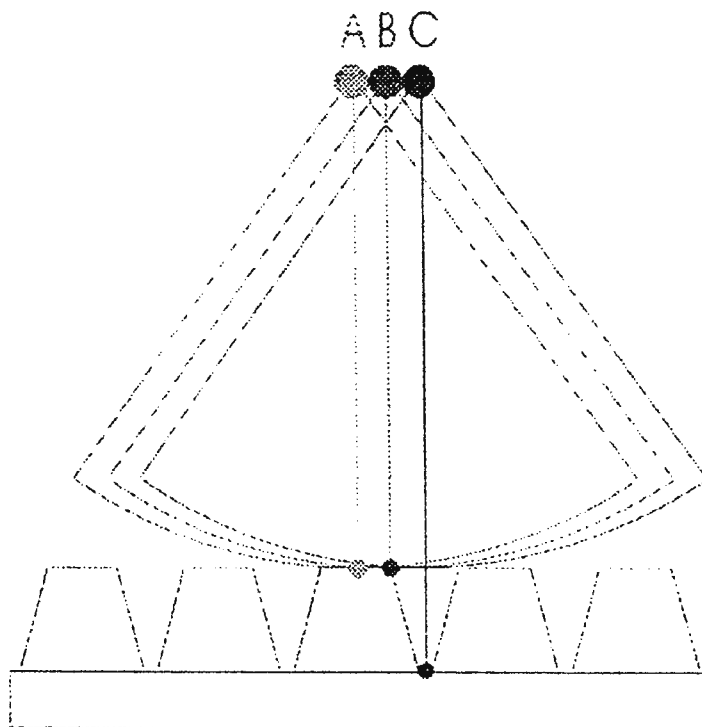

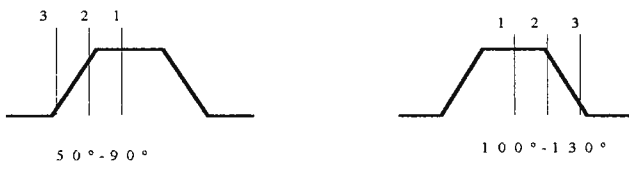


Figura 1.2.2.5. Los tres puntos característicos en la barrera de material Beton Bois.

He aquí una **tabla-resumen** con las condiciones del ensayo:

<b>MUESTRA:</b> Beton Bois	<b>TAMAÑO DE MUESTRA:</b> 4 x 4 m <sup>2</sup>
<p><b>PLANO DE REFERENCIA:</b></p> 	
<p><b>PUNTOS DE MEDIDA:</b></p> 	
<b>Longitud de la ventana:</b> 5.18 ms	<b>Mínima frecuencia fiable:</b> 240 Hz

Por tanto la medida se realizó orientando el equipo de medida según **los 9 ángulos** de incidencia específicos. Esto se hace tantas veces como puntos característicos tenga la muestra, en este caso tres. Posteriormente se halla la pérdida por reflexión de nuestra barrera antirruído como el promedio de la pérdida por reflexión (RL) para los tres puntos.

A continuación, aparece una fotografía representativa de la realización de esta medida:

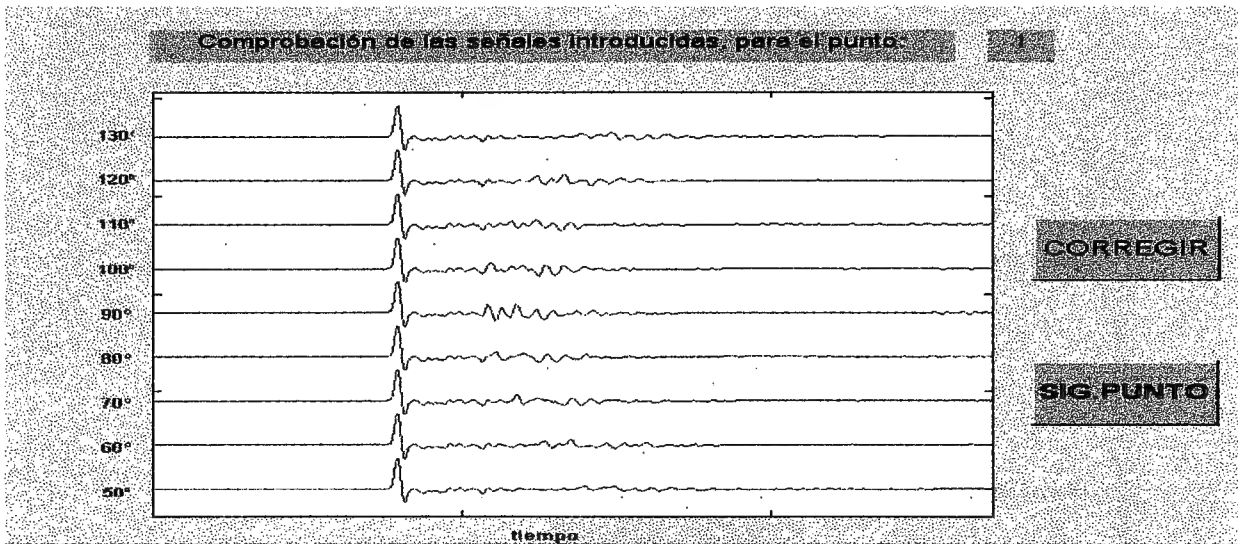


Figura 1.2.2.6. Realización de la medida de la pérdida por reflexión (RL) en laboratorio para una barrera de material Beton Bois.

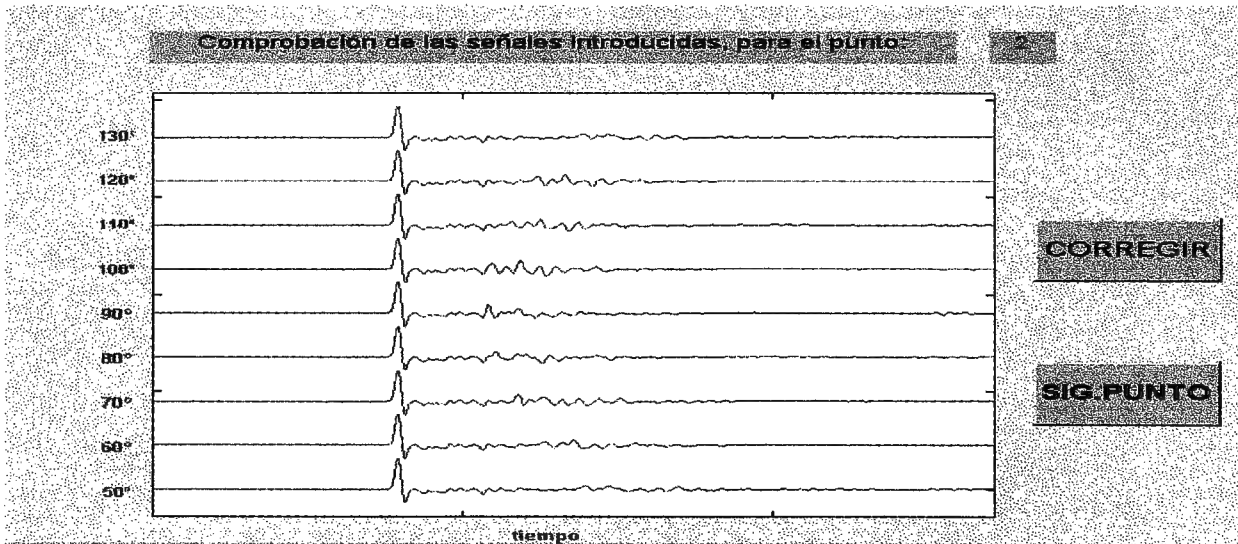
Las señales medidas según el procedimiento indicado anteriormente se introducen en el software realizado en este proyecto de fin de carrera. Se obtienen las siguientes **salidas**:

**a. Ecogramas de las señales introducidas para los tres puntos:**

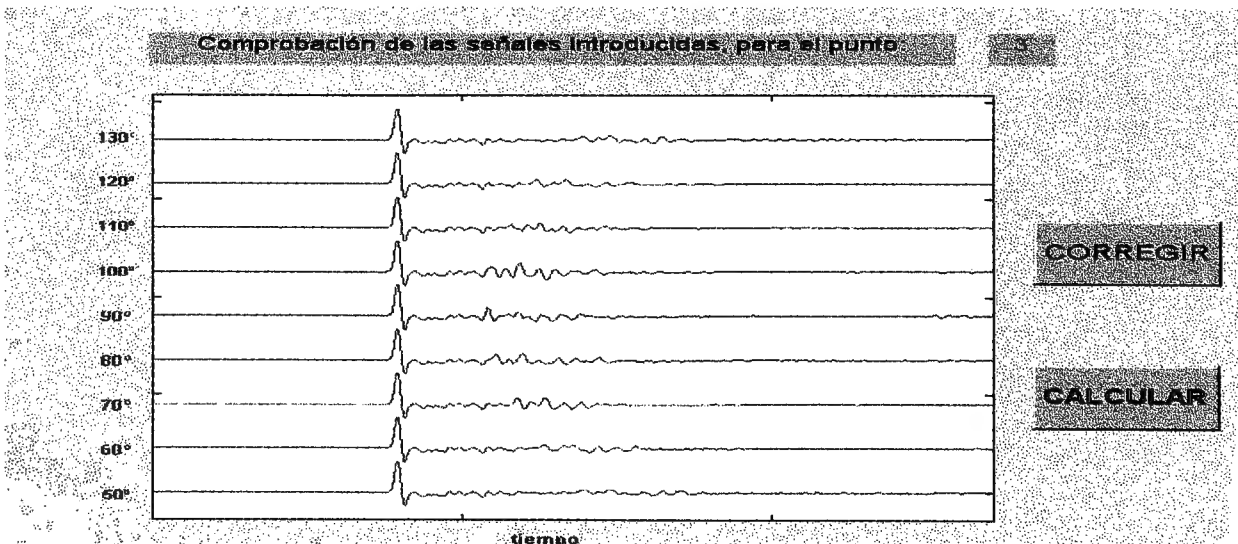
**punto 1:**



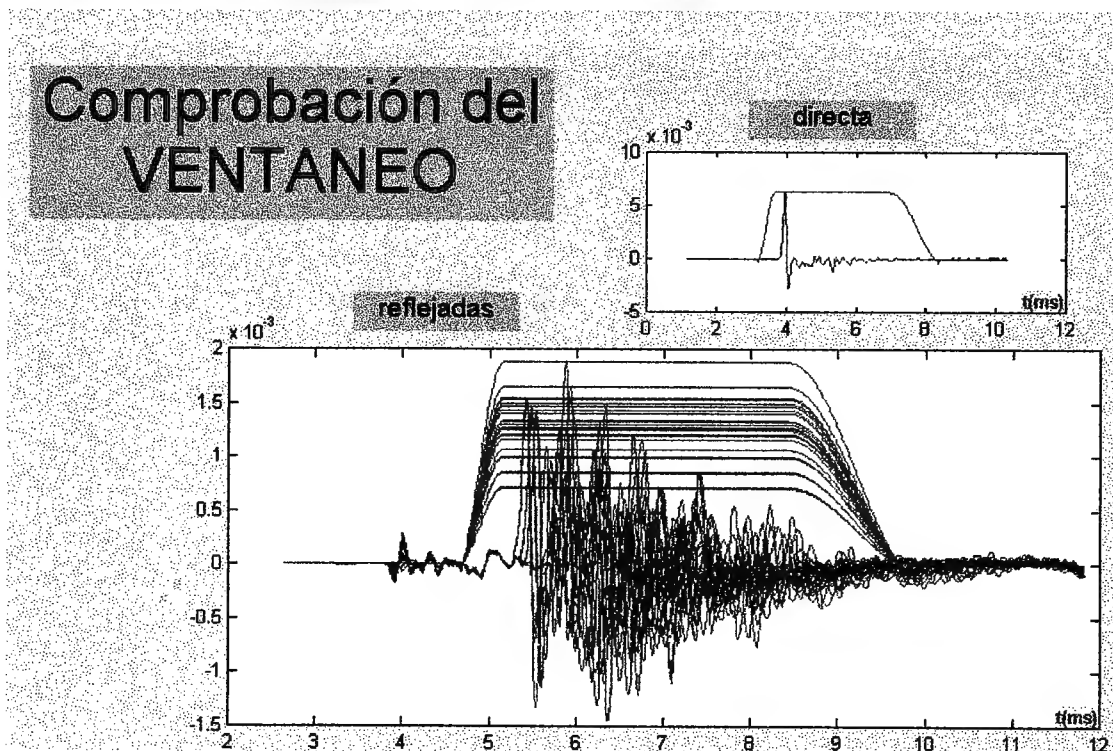
**punto 2:**



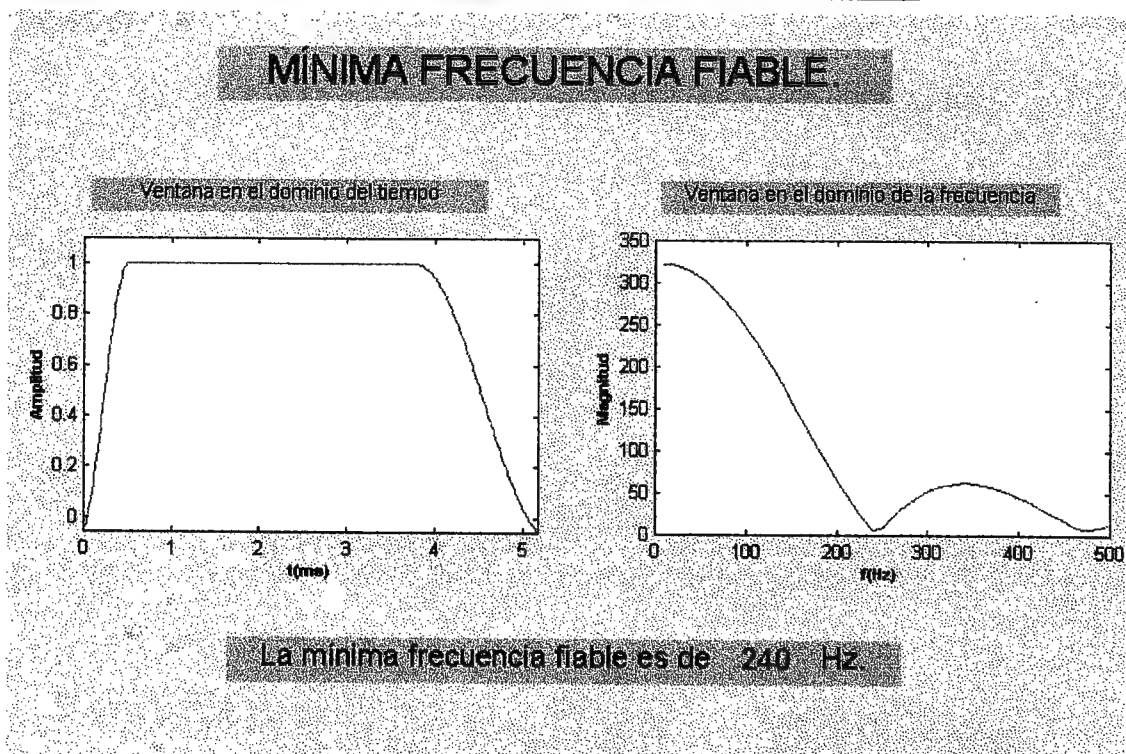
**punto 3:**



**b. Comprobación de cómo se ha efectuado el ventaneo:**

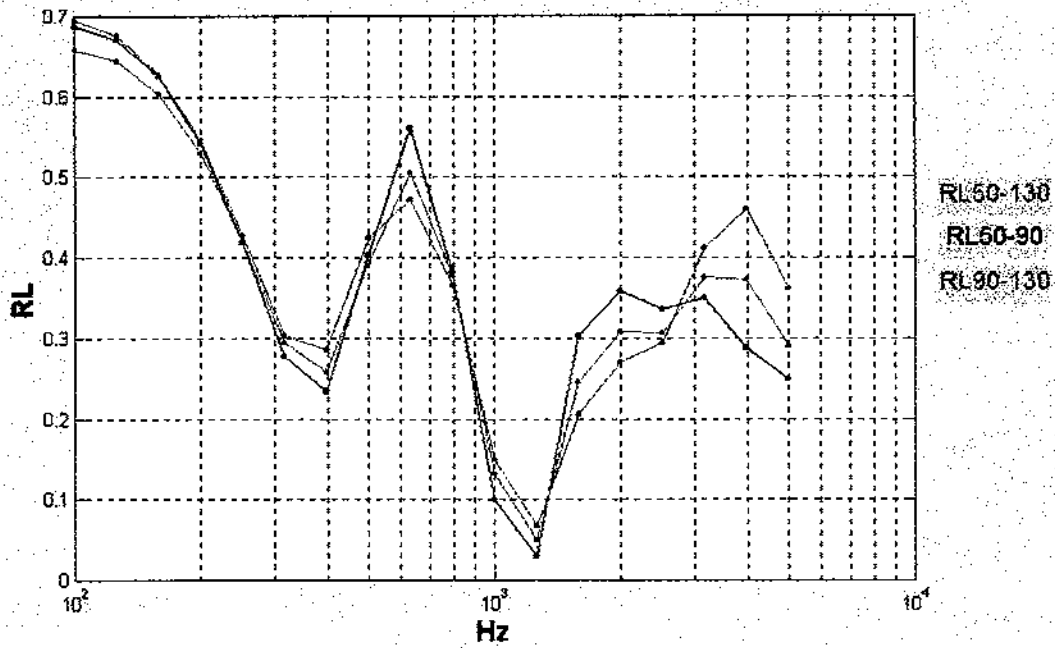


**c. Frecuencia por debajo de la cual los resultados no son fiables:**



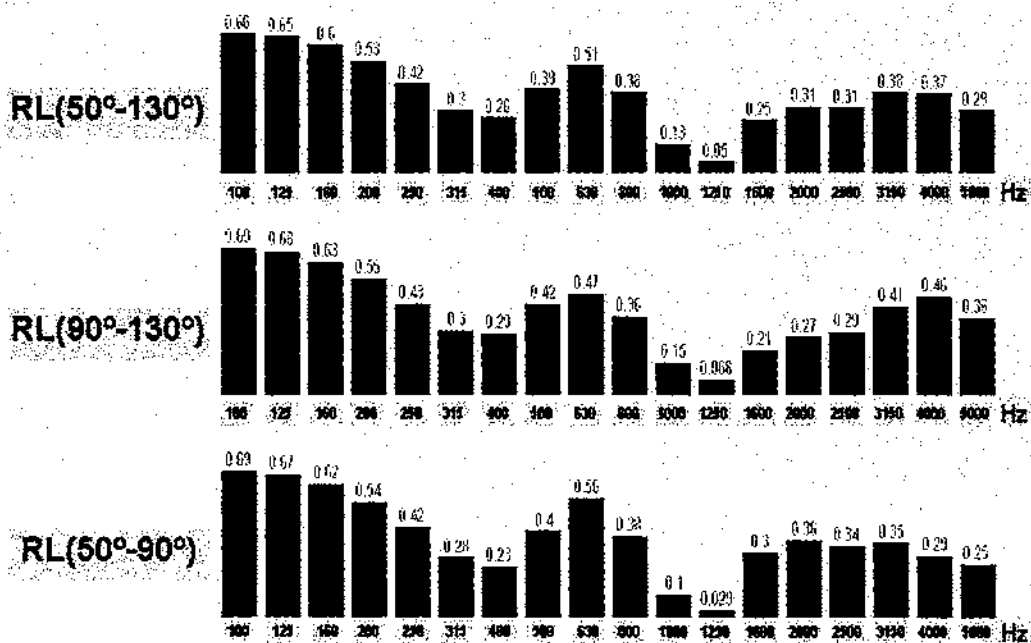
**d. Comparación de la pérdida por reflexión para los tres promedios angulares:**

**Comparación de la RL para los distintos ángulos de incidencia**



**e. Valores numéricos de la pérdida por reflexión (RL) para cada promedio angular:**

**PROMEDIOS DE LA PÉRDIDA POR REFLEXIÓN.**





### **1.2.2.2.1. RESUMEN Y CONCLUSIONES:**

La muestra ensayada se caracteriza por tener tres puntos de estudio, dada la forma de la superficie.

En el procesado de la señal se puede observar que el **ventaneo** ha sido realizado correctamente, ya que todos los ecos reflejados ( $9*3=27$ ) se hallan dentro de la ventana temporal Adrienne.

Esta barrera es especialmente absorbente en el entorno de los 1000Hz, donde se tienen pérdidas por reflexión muy bajas, del orden de 0.1. En el resto de las frecuencias presenta unas **características absorbentes medias**, si las comparamos con las de barreras reflectantes y absorbentes.

No se aprecian diferencias significativas en los valores de la pérdida acústica por reflexión (RL) para los distintos promedios angulares.

La **mínima frecuencia** por debajo de la cual no se obtienen resultados fiables de la pérdida por reflexión es de **240 Hz**.

Una descripción más detallada del contenido de cada una de estas pantallas aparece en el apartado "**3.2.1. MANUAL DE USUARIO**", perteneciente al pliego de condiciones.

### 1.2.2.3. ENSAYO IN SITU.

Una vez que la validez del método quedó demostrada en los ensayos en laboratorio el equipo investigador del proyecto Adrienne decidió hacer ensayos al aire libre, **in situ**. El verdadero objetivo del proyecto Adrienne fue poner un método que fuera capaz de caracterizar las barreras antirruído en su lugar de instalación (carretera) y bajo las mismas condiciones (ruido de tráfico) de funcionamiento habitual.

Estos ensayos fueron realizados en septiembre de 1997, en Grenoble. Se intentó que la disposición de la barrera antirruído fuese lo más parecida posible a la instalación real de esta clase de dispositivos. Se construyó para la ocasión una barrera antirruído formada por los distintos materiales que iban a ser ensayados.



Figura 1.2.2.3.1. Barrera construida para los ensayos in situ de Grenoble.

La **disposición** de la barrera y del equipo de medida, que difiere de la adoptada para los ensayos de laboratorio, puede apreciarse en la siguiente figura.

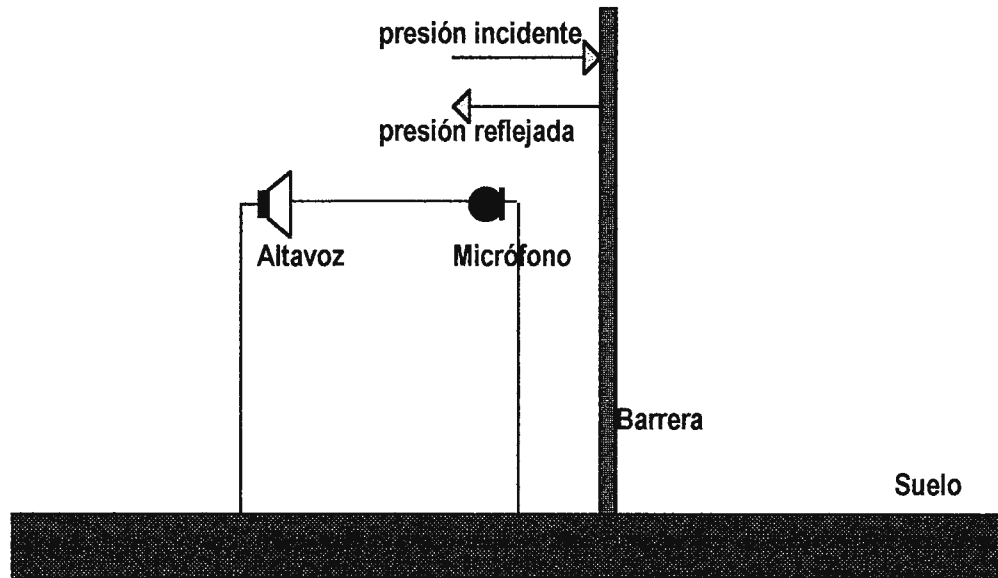


Figura 1.2.2.3.2. Disposición para el ensayo in situ.

Las medidas que han sido introducidas en el software de este proyecto de fin de carrera fueron realizadas sobre una barrera de un material denominado **MICE**. Se trata de chapa perforada con un aislante acústico en su interior, la "lana de roca" (Rockwool). Sobre esta barrera fueron tomadas varias medidas bajo los ángulos de incidencia específicos del método.

A continuación aparece una fotografía en la que se puede apreciar cómo se tomó la medida en la barrera de MICE.

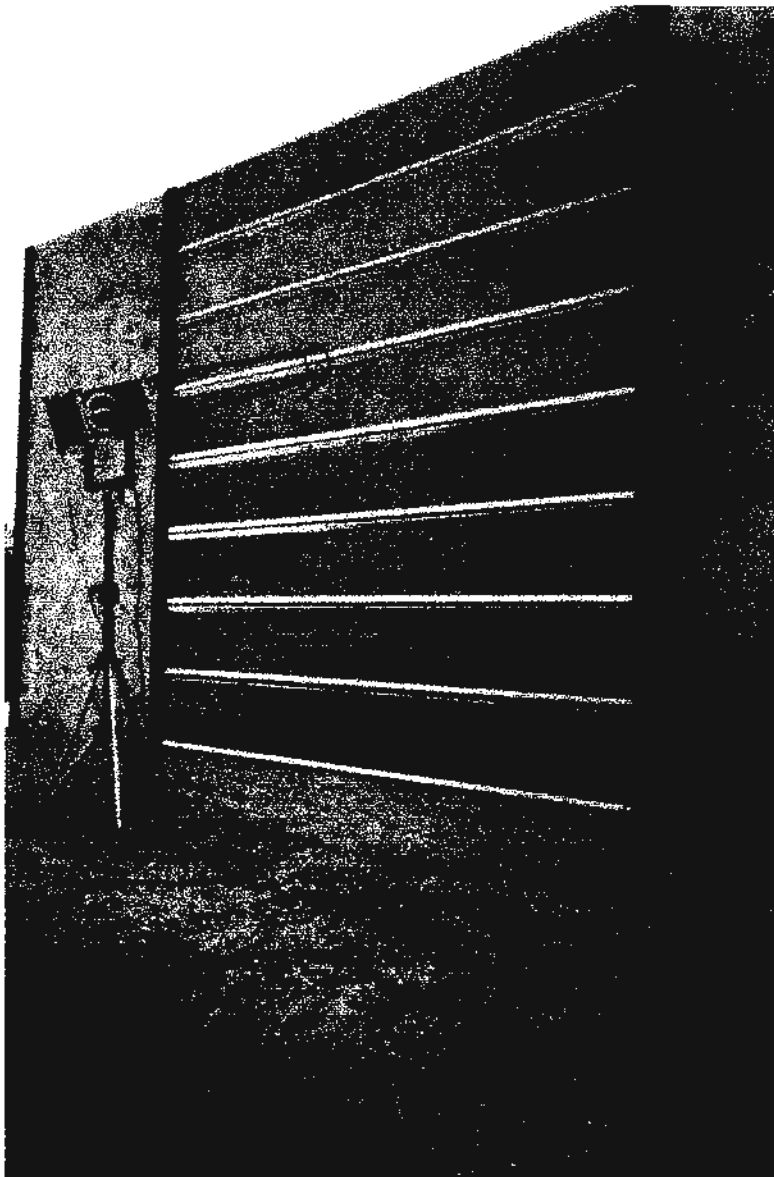


Figura 1.2.2.3.3.  
Medida de la  
pérdida por  
reflexión in situ  
en una barrera  
de MICE.

El micrófono recoge tanto la señal directa como la reflejada. Posteriormente, aplicando la "**TÉCNICA DE SUBSTRACCIÓN DE SEÑAL**" descrita en el apartado 1.1.3.2.1., se consiguen separar el eco directo y el reflejado.

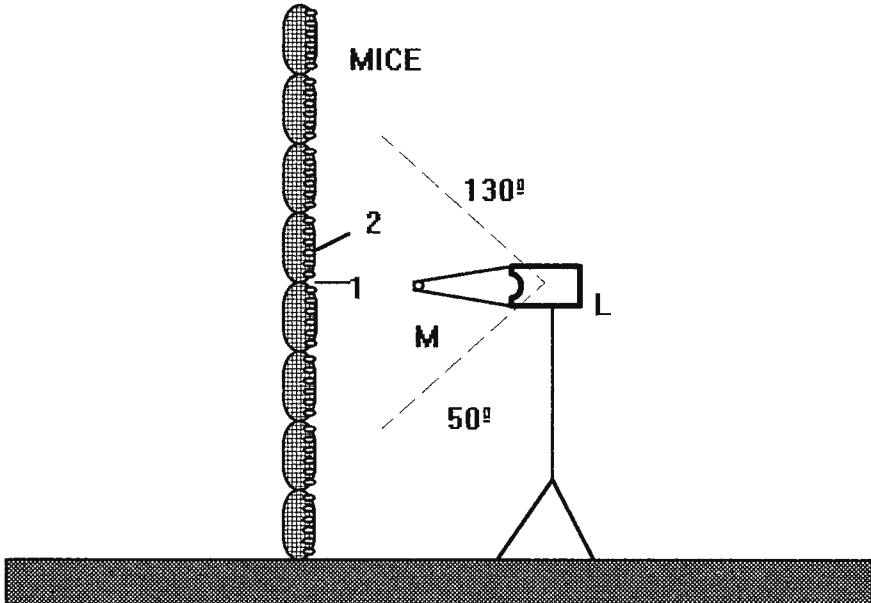
La adquisición de la **señal de referencia** es bastante simple. Basta con orientar el equipo de medida hacia algún lugar donde no haya obstáculos que puedan provocar reflexiones indeseadas. En el ensayo in situ se orienta el equipo hacia el cielo, de este modo se obtiene simplemente el eco directo.



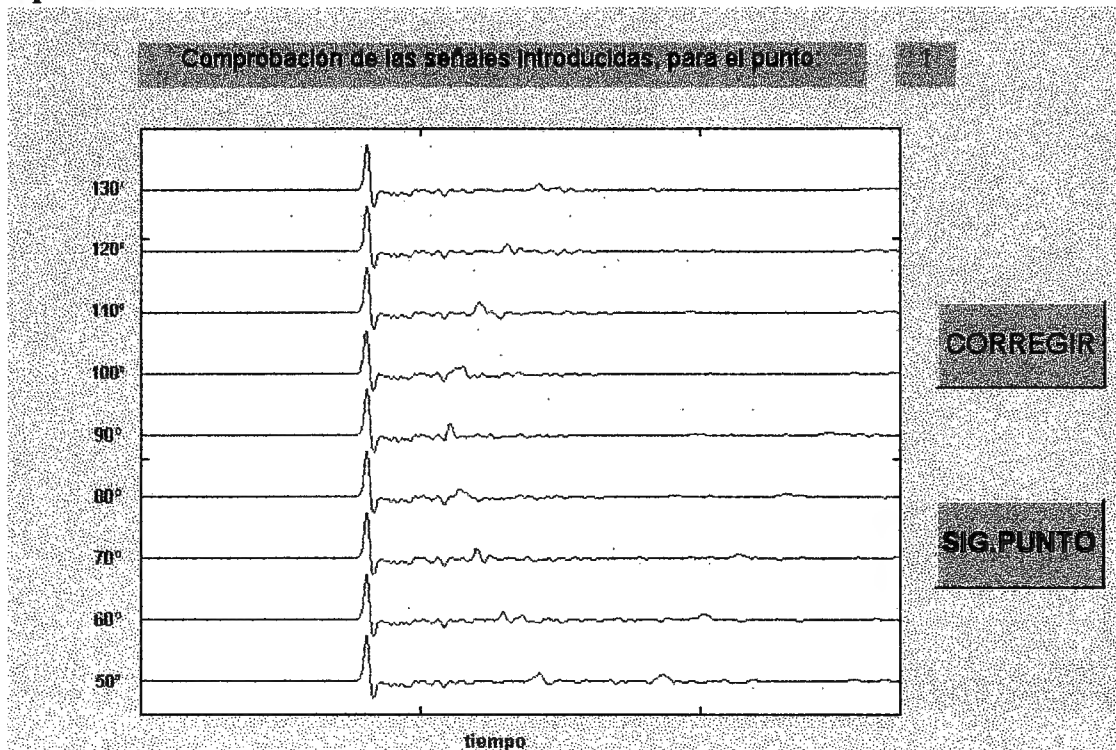
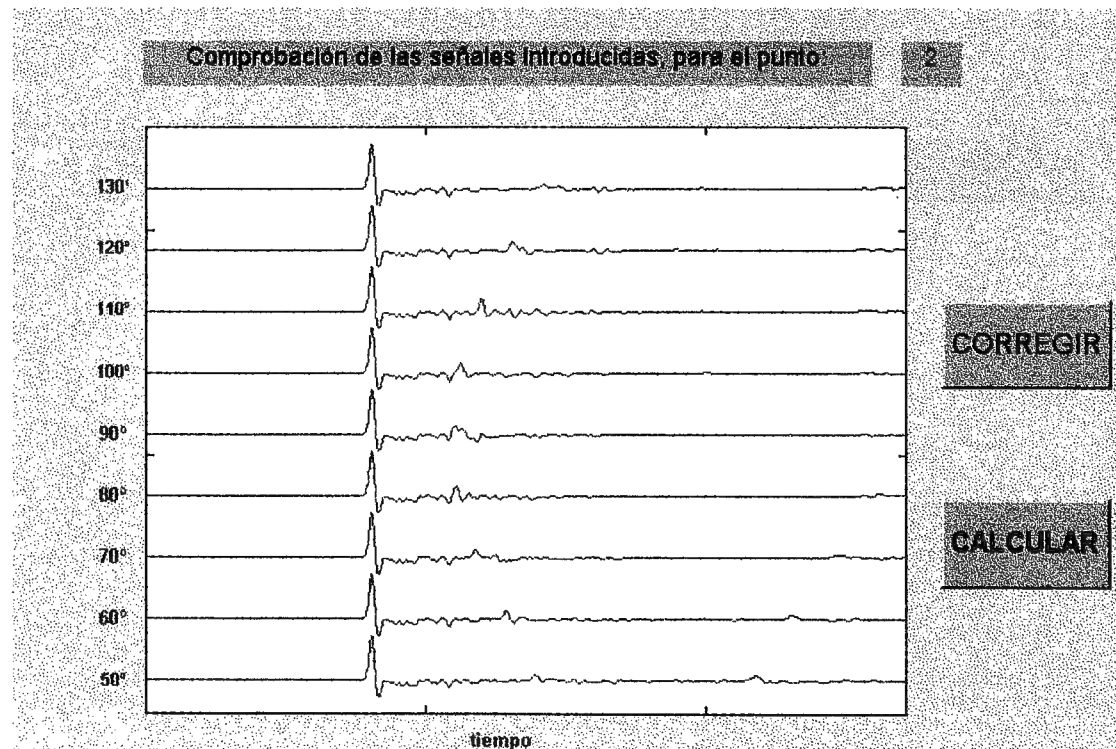
Figura 1.2.2.3.4.  
Orientación del  
equipo de medida  
para la obtención  
de la señal de  
referencia.

Una vez que se tienen las señales directa y reflejada se calculan las energías asociadas a cada una de ellas. Pasándolas al dominio de la frecuencia y dividiendo la energía reflejada entre la incidente, se obtiene la **pérdida acústica por reflexión (RL)** para cada banda de tercio de octava.

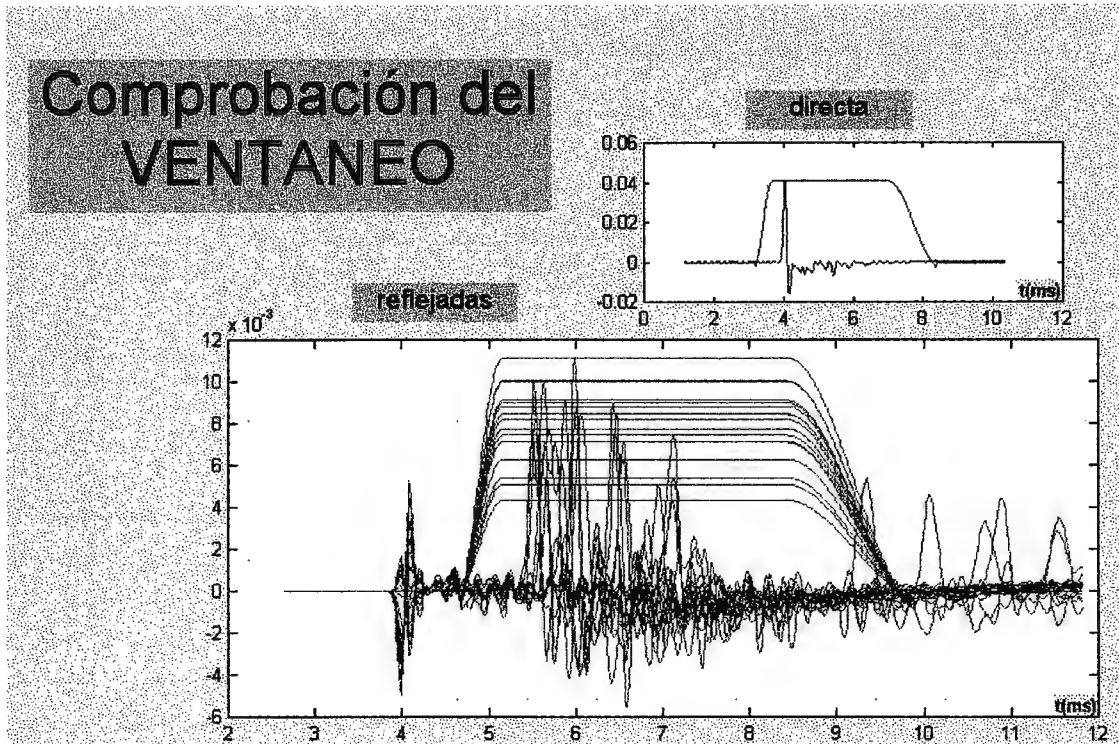
He aquí una **tabla-resumen** con las condiciones del ensayo:

<b>MUESTRA:</b> MICE	<b>TAMAÑO DE MUESTRA:</b> 4x4 m <sup>2</sup>
<b>NOTAS:</b> El sistema altavoz-micrófono se orienta hacia abajo (50°-90°) y hacia arriba (90°-130°). Para cada ángulo, el ensayo se hace para dos puntos: punto 1 (junta de dos paneles) y 2 (en el centro del panel).	
<b>CROQUIS:</b> 	

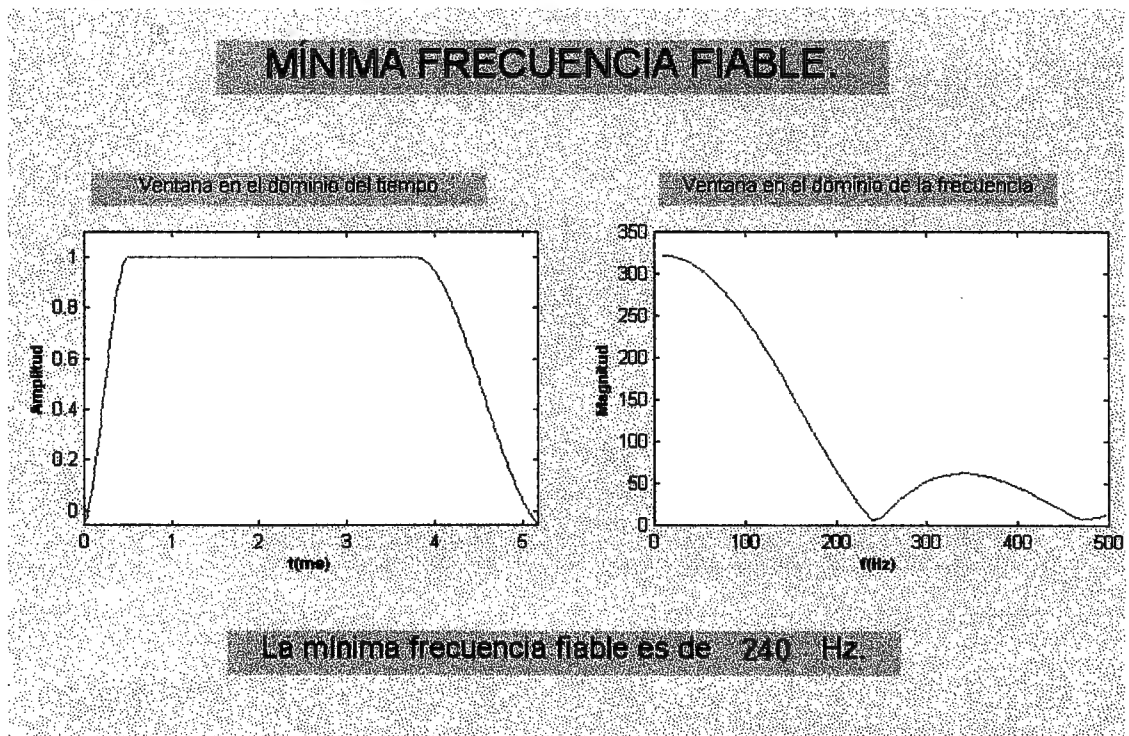
Las señales medidas según el procedimiento indicado anteriormente se introducen en el software realizado en este proyecto de fin de carrera. Se obtienen las siguientes salidas:

**a. Ecogramas de las señales introducidas para los tres puntos:****punto 1:****punto 2:**

**b. Comprobación de cómo se ha efectuado el ventaneo:**

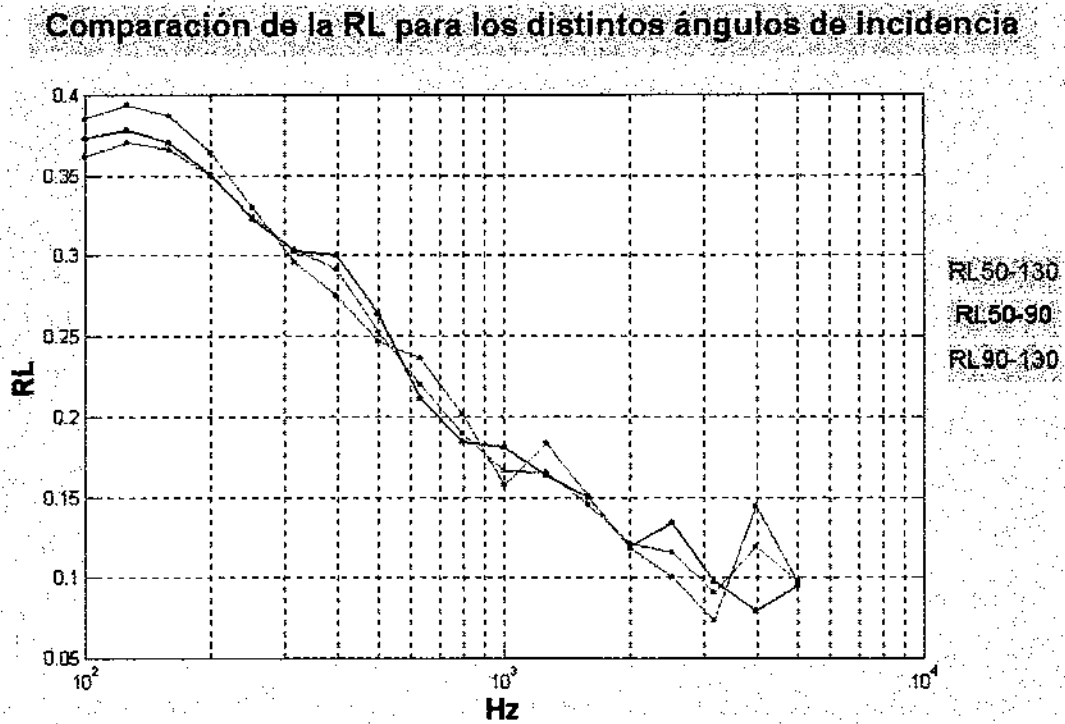


**c. Frecuencia por debajo de la cual los resultados no son fiables:**

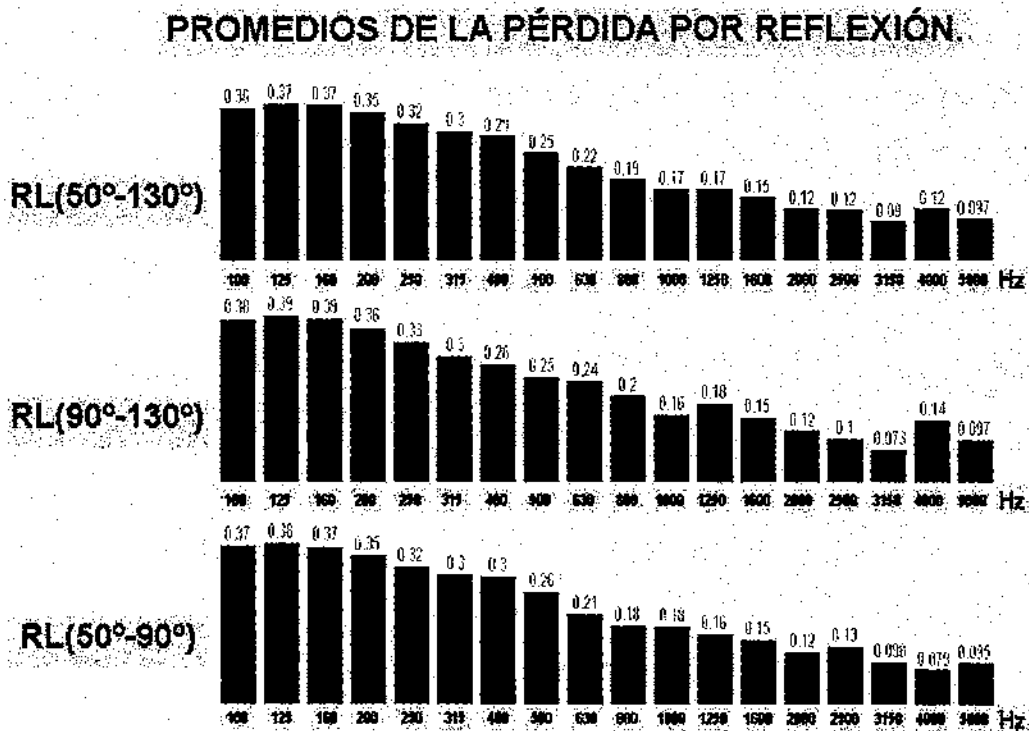




**d. Comparación de la pérdida por reflexión para los tres promedios angulares:**



**e. Valores numéricos de la pérdida por reflexión (RL) para cada promedio angular:**



### **1.2.2.3.1. RESUMEN Y CONCLUSIONES.**

La muestra ensayada se caracteriza por tener dos puntos de estudio, dada la forma de la superficie.

En el procesado de la señal se puede observar que el ventaneo ha sido realizado correctamente, ya que todos los ecos reflejados ( $9 \times 2 = 18$ ) se hallan dentro de la ventana temporal Adrienne. Esto es un **avance** con respecto al método Adrienne creado por el equipo investigador europeo. El método Adrienne, a la fecha de entrega de este proyecto de fin de carrera, usa siempre la **misma ventana** temporal, independientemente de los parámetros geométricos del ensayo. El programa desarrollado en este proyecto fin de carrera tiene también como entradas los **parámetros geométricos** (distancias relativas entre altavoz, micrófono y barrera, altura de la barrera). En función de dichos parámetros calcula la ventana Adrienne lo más grande posible (para una óptima resolución en frecuencia) pero evitando coger reflexiones indeseadas (lo que limita el tamaño de la ventana y no hace posible tener una frecuencia fiable de 100 Hz, que era el objetivo inicial).

La ventana Adrienne usada anteriormente por el equipo investigador Adrienne es demasiado larga y coge reflexiones indeseadas. Esto ocurre, sobre todo, cuando el equipo de medida se orienta hacia abajo ( $50^\circ$ - $90^\circ$ ). Si la ventana no tiene la longitud justa se procesa la señal con ecos provenientes del **suelo** y se obtiene una RL mayor que la real.

Esto se puede apreciar en la **figura 1.2.2.3.5.**, de la página siguiente. En ella aparecen los promedios angulares de la RL que obtuvo el equipo investigador Adrienne. Se puede observar cómo el **promedio para  $50^\circ$ - $90^\circ$**  sale mucho mayor que los demás debido a las reflexiones indeseadas provenientes del suelo.

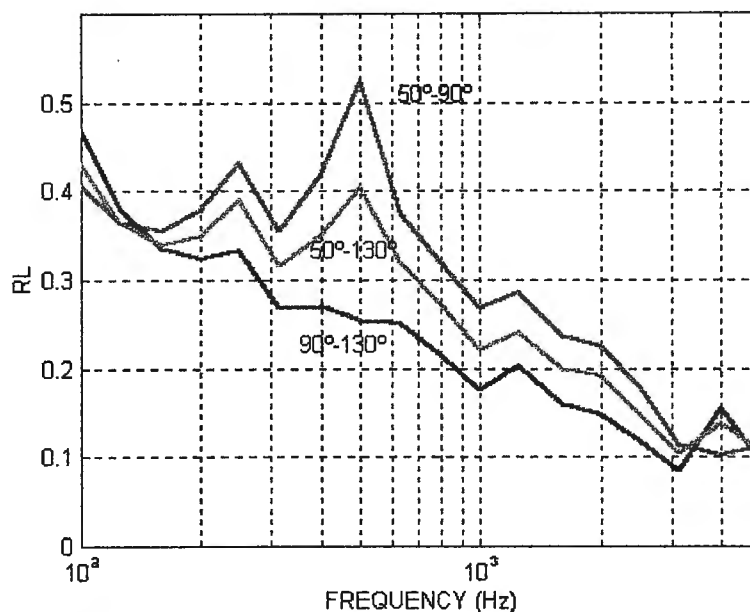


Figura 1.2.2.3.5. Promedios angulares de la pérdida por reflexión que obtuvo el equipo investigador Adrienne para la muestra MICE.

Esta barrera es de un material **absorbente** que se caracteriza por tener pérdidas por reflexión bajas.

No se aprecian diferencias significativas en los valores de la pérdida acústica por reflexión (RL) para los distintos promedios angulares.

La **mínima frecuencia** por debajo de la cual no se obtienen resultados fiables de la pérdida por reflexión es de **240 Hz**.

Una descripción más detallada del contenido de cada una de estas pantallas aparece en el apartado "**3.2.1.MANUAL DE USUARIO**", perteneciente al **pliego de condiciones**.

## **1.3. ANEJOS.**

## **1.3.0. ÍNDICE.**

**1.3.0. ÍNDICE.....118**

**1.3.1. LISTADO DEL PROGRAMA.....119**

1.3.1.1. FUNCIONES DE INTRODUCCIÓN DE DATOS.....119

1.3.1.2. FUNCIONES DE CÁLCULO.....147

1.3.1.3. FUNCIONES DE SALIDA DE RESULTADOS.....156

**1.3.2. PROYECTO DE NORMA EUROPEA PrEN 1793.....194**

1.3.2.1. PROYECTO DE NORMA EUROPEA PrEN 1793-1.....194

1.3.2.2. PROYECTO DE NORMA EUROPEA PrEN 1793-3.....205

### **1.3.1. LISTADO DEL PROGRAMA.**

### **1.3.1.1. FUNCIONES DE INTRODUCCIÓN DE DATOS.**

Estas funciones tienen como objetivo adquirir los datos necesarios para calcular la pérdida por reflexión (RL) de la barrera antirruído.

Todo el proceso de adquisición de datos se hace a través de G.U.I., interfaces gráficas, que resultan cómodas e intuitivas.

Dentro de las siguientes funciones hay algunas destinadas a que el usuario compruebe que los datos han sido correctamente introducidos. De este modo, se puede asegurar que los cálculos serán representativos de las propiedades absorbentes de la barrera antirruído.



## adrienne.m

```
%Inicia el programa. Carga el sonido que se oirá al apretar los  
pushbuttons  
%y llama a la pantalla de presentación.
```

```
load clic;  
portada;
```

## portada.m

```
function portada()
%[]=portada().
%
% Pantalla de presentación del programa, que aparece con música.
%
%No tiene parámetros de entrada.
%
%No tiene parámetros de salida.

%Se carga la música y se hace sonar.
load sonido;
sound(sonido,22500,16);

%Se carga la variable asociada a portada que contiene datos necesarios
para que
%aparezca correctamente la pantalla gráfica.
load portada

%Parámetros generales de la pantalla gráfica.
a = figure('Units','normalized', ...
    'Color',[0.95 0.95 0.8], ...
    'Colormap',mat0, ...
    'PointerShapeCData',mat1, ...
    'Position',[0 0.05 1 0.88125], ...
    'Tag','Fig1');

%Cuadro de texto que contiene el título del proyecto.
b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'FontName','Arial', ...
    'FontSize',0.189655, ...
    'FontWeight','bold', ...
    'Position',[0.0385572 0.294007 0.924129 0.413858], ...
    'String','MEDIDA IN SITU DE LA PÉRDIDA ACÚSTICA POR REFLEXIÓN DE
BARRERAS ANTIRRUIDO POR EL MÉTODO "ADRIENNE"', ...
    'Style','text', ...
    'Tag','StaticText1');

%Cuadro de texto que contiene el autor del proyecto.
b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'FontAngle','italic', ...
    'FontName','Arial', ...
    'FontSize',0.533333, ...
    'Position',[0.039801 0.105805 0.553483 0.0543071], ...
    'String','Realizado por: Adolfo Corral Quintana.', ...
    'Style','text', ...
    'Tag','StaticText2');

%Cuadro de texto de la institución donde se realizó el proyecto.
b = uicontrol('Parent',a, ...
    'Units','normalized', ...
```

## portada.m

```
'FontUnits','normalized', ...
'FontAngle','italic', ...
'FontName','Arial', ...
'FontSize',0.682171, ...
'Position',[0.0385572 0.857678 0.532338 0.0617978], ...
'String','Instituto de Acústica; C.S.I.C.', ...
'Style','text', ...
'Tag','StaticText3');

%Pushbutton que nos lleva a la siguiente pantalla gráfica, r11.
%El pushbutton no sale hasta que han pasado 6 segundos porque es
%el tiempo que tarda en sonar la música de presentación.
pause(6);

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'Callback','sound(clic,22050,16);r11;', ...
'FontName','Arial', ...
'FontSize',0.375, ...
'Position',[0.7699 0.082397 0.139303 0.101124], ...
'String','SEGUIR', ...
'Tag','Pushbutton1');
```

## r11.m

```
function r11();
%r11()
%
% Pantalla gráfica donde se introducen los siguientes datos necesarios
%para el cálculo de la pérdida por reflexión:
% -directa, señal directa medida.
% -n, número de puntos que vamos a estudiar en la barrera.
% -dt, periodo de muestreo, en ms.
% -2*dSE, altura de la barrera, en m.
% -dMS, distancia entre el micrófono y la barrera, en m.
%
%No tiene parámetros de entrada.
%
%En el workspace, quedan las siguientes variables:
% directa
% n
% nom_dir, variable necesaria para la siguiente pantalla gráfica, r11a.
%     Es el nombre de la variable que contiene la señal directa.
% dt
% dSE
% dMS

%Se cierra la anterior pantalla gráfica.
close;

%Se carga la variable que contiene datos necesarios para que se ejecute
correctamente
%la pantalla gráfica.
load r11 ;

%Parámetros generales de la pantalla gráfica.
a = figure('Units','normalized', ...
    'Color',[0.95 0.95 0.8], ...
    'Colormap',mat0, ...
    'PointerShapeCData',mat1, ...
    'Position',[0 0.05 1 0.88125], ...
    'Tag','Fig1');

%Cuadro de texto que contiene el título de la pantalla gráfica:
%"Pérdida por reflexión (RL):".
b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'FontName','Arial', ...
    'FontSize',0.547009, ...
    'Position',[0.195761 0.91573 0.587282 0.0543071], ...
    'String','Pérdida por reflexión (RL):', ...
    'Style','text', ...
    'Tag','StaticText1');

%Cuadro de texto y texto editable donde se introduce el número de puntos
de
%estudio en la barrera acústica, n.
b = uicontrol('Parent',a, ...
```

## rl1.m

```
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'FontName','Arial', ...
'FontSize',0.3, ...
'Position',[0.05 0.568998 0.35125 0.0982987], ...
'String','Introduce el número de puntos de medida:', ...
'Style','text', ...
'Tag','StaticText2');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[1 1 1], ...
'Callback','n=str2num(get(gcbo, 'String'));', ...
'FontName','Arial', ...
'FontSize',0.35, ...
'Position',[0.4325 0.586011 0.12 0.0586011], ...
'String','', ...
'Style','edit', ...
'Tag','EditText1');

%Cuadro de texto y texto editable donde se introduce la señal directa
%medida, directa. Para almacenarla se usa la función fila, de modo que,
%independientemente de si se ha introducido como vector fila o como
vector
%columna, se transforma en fila. También se almacena el nombre de la
variable que lo
%contiene, nom_dir.
b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'FontName','Arial', ...
'FontSize',0.3, ...
'Position',[0.05 0.716446 0.35125 0.100189], ...
'String','Introduce la variable que contiene la señal directa:', ...
'Style','text', ...
'Tag','StaticText2');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[1 1 1], ...
'Callback','directa=fila(eval(get(gcbo,
'String')));nom_dir=get(gcbo, 'String');', ...
'FontName','Arial', ...
'FontSize',0.35, ...
'Position',[0.4325 0.73535 0.12 0.0604915], ...
'String','', ...
'Style','edit', ...
'Tag','EditText1');

%Cuadro de texto y texto editable donde se introduce el periodo de
muestreo
%en ms, dt.
b = uicontrol('Parent',a, ...
'Units','normalized', ...
```

```

'FontUnits','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'FontName','Arial', ...
'FontSize',0.3, ...
'Position',[0.05 0.417769 0.35125 0.100189], ...
'String','Introduce el periodo de muestreo, en ms:', ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[1 1 1], ...
'Callback','dt=str2num(get(gcbo, 'String'))';', ...
'FontName','Arial', ...
'FontSize',0.35, ...
'Position',[0.4325 0.434783 0.12 0.0604915], ...
'String','', ...
'Style','edit', ...
'Tag','EditText2');

%Cuadro de texto y texto editable donde se introduce la altura de la
barrera
%acústica en m, 2*dSE.
b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'FontName','Arial', ...
'FontSize',0.3, ...
'Position',[0.05 0.266541 0.35125 0.100189], ...
'String','Introduce la altura de la barrera, en m:', ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[1 1 1], ...
'Callback','dSE=0.5*(str2num(get(gcbo, 'String')));', ...
'FontName','Arial', ...
'FontSize',0.35, ...
'Position',[0.4325 0.283554 0.12 0.0604915], ...
'String','', ...
'Style','edit', ...
'Tag','EditText2');

%Cuadro de texto y texto editable donde se introduce la distancia entre
el
micrófono y al barrera en m, dMS.
b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'FontName','Arial', ...
'FontSize',0.28, ...
'Position',[0.05 0.115312 0.35125 0.100189], ...

```

## r11.m

```
'String','Introduce, en m, la distancia entre el micrófono y la
barrera:', ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[1 1 1], ...
'Callback','dMS=str2num(get(gcbo, 'String')); ', ...
'FontName','Arial', ...
'FontSize',0.35, ...
'Position',[0.4325 0.132325 0.12 0.0604915], ...
'String','', ...
'Style','edit', ...
'Tag','EditText2');

%Pushbutton que llama a la siguiente pantalla gráfica, r11a. Se debe
pulsar
%una vez que ya están introducidos todos los datos.
b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'Callback','sound(clic,22050,16);r11a(nom_dir,directa,n,dt,dSE,dMS);',
...
'FontName','Arial', ...
'FontSize',0.375, ...
'Position',[0.7699 0.082397 0.139303 0.101124], ...
'String','SEGUIR', ...
'Tag','Pushbutton1');
```

## fila.m

```
function fila = fila(vector);
%fila = fila(vector).
%
% Devuelve un vector en formato fila, independientemente de si se
%ha introducido como fila o como columna.
%
%Parámetro de entrada:
% vector, vector que puede ser introducido en formato fila o columna.
%
%Parámetro de salida:
% fila, vector que contiene la misma información que 'vector', pero
% en formato fila.

%Algoritmo de transformación.
if size(vector,1)>1
    fila=vector';
else
    fila=vector;
end;
```



## rlla.m

```
function rlla(nom_dir,directa,n,dt,dSE,dMS);
%rlla(nom_dir,directa,n,dt,dSE,dMS);
%
% Es la misma pantalla gráfica que la anterior, rll, pero con la
%diferencia de que en esta también se introduce la distancia entre el
%altavoz y el micrófono. Se informa en esta pantalla de la distancia
entre
%altavoz y micrófono ideal para una óptima mínima frecuencia fiable.
%
%Parámetros de entrada:
% -nom_dir, nombre de la variable que contiene la señal directa.
% -directa, señal directa medida.
% -n, número de puntos que vamos a estudiar en la barrera.
% -dt, periodo de muestreo, en ms.
% -2*dSE, altura de la barrera, en m.
% -dMS, distancia entre el micrófono y la barrera, en m.
%
%En el workspace quedan las siguientes variables:
% -directa
% -n
% -dt
% -dSE
% -dMS
% -dLM, distancia entre altavoz y micrófono.
% -i, variable necesaria para la siguiente pantalla gráfica, rl2.

%Se cierra la anterior pantalla gráfica.
close;

%Se carga la variable que contiene datos necesarios para que se ejecute
correctamente
%la pantalla gráfica.
load rlla;

%Parámetros generales de la pantalla gráfica.
a = figure('Units','normalized', ...
    'Color',[0.95 0.95 0.8], ...
    'Colormap',mat0, ...
    'PointerShapeCData',mat1, ...
    'Position',[0 0.05 1 0.88125], ...
    'Tag','Fig1');

%Cuadro de texto que contiene el título de la pantalla gráfica:
%"Pérdida por reflexión (RL):".
b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'FontName','Arial', ...
    'FontSize',0.547009, ...
    'Position',[0.195761 0.91573 0.587282 0.0543071], ...
    'String','Pérdida por reflexión (RL):', ...
    'Style','text', ...
    'Tag','StaticText1');
```

## rlla.m

%Cuadro de texto y texto editable donde se introduce el número de puntos de

%estudio en la barrera acústica, n.

```
b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'FontName','Arial', ...
    'FontSize',0.3, ...
    'Position',[0.05 0.568998 0.35125 0.0982987], ...
    'String','Introduce el número de puntos de medida:', ...
    'Style','text', ...
    'Tag','StaticText2');
```

```
b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'Callback','n=str2num(get(gcbo, 'String'));', ...
    'FontName','Arial', ...
    'FontSize',0.35, ...
    'Position',[0.4325 0.586011 0.12 0.0586011], ...
    'String',num2str(n), ...
    'Style','edit', ...
    'Tag','EditText1');
```

%Cuadro de texto y texto editable donde se introduce la señal directa  
%medida, directa.

```
b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'FontName','Arial', ...
    'FontSize',0.3, ...
    'Position',[0.05 0.716446 0.35125 0.100189], ...
    'String','Introduce la variable que contiene la señal directa:', ...
    'Style','text', ...
    'Tag','StaticText2');
```

```
b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'Callback','directa=fila(eval(get(gcbo, 'String')));', ...
    'FontName','Arial', ...
    'FontSize',0.35, ...
    'Position',[0.4325 0.73535 0.12 0.0604915], ...
    'String',nom_dir, ...
    'Style','edit', ...
    'Tag','EditText1');
```

%Cuadro de texto y texto editable donde se introduce el periodo de muestreo

%en ms, dt.

```
b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
```

## rl1a.m

```
'FontName','Arial', ...
'FontSize',0.3, ...
'Position',[0.05 0.417769 0.35125 0.100189], ...
'String','Introduce el periodo de muestreo, en ms:', ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[1 1 1], ...
'Callback','dt=str2num(get(gcbo, 'String'));', ...
'FontName','Arial', ...
'FontSize',0.35, ...
'Position',[0.4325 0.434783 0.12 0.0604915], ...
'String',num2str(dt), ...
'Style','edit', ...
'Tag','EditText2');

%Cuadro de texto y texto editable donde se introduce la altura de la
barrera
%acústica en m, 2*dSE.
b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'FontName','Arial', ...
'FontSize',0.3, ...
'Position',[0.05 0.266541 0.35125 0.100189], ...
'String','Introduce la altura de la barrera, en m:', ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[1 1 1], ...
'Callback','dSE=0.5*(str2num(get(gcbo, 'String'))); ', ...
'FontName','Arial', ...
'FontSize',0.35, ...
'Position',[0.4325 0.283554 0.12 0.0604915], ...
'String',num2str(2*dSE), ...
'Style','edit', ...
'Tag','EditText2');

%Cuadro de texto y texto editable donde se introduce la distancia entre
el
%micrófono y al barrera en m, dMS.
b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'FontName','Arial', ...
'FontSize',0.28, ...
'Position',[0.05 0.115312 0.35125 0.100189], ...
'String','Introduce, en m, la distancia entre el micrófono y la
barrera:', ...
'Style','text', ...
```

```

    'Tag','StaticText3');

b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'Callback','dMS=str2num(get(gcbo, 'String'));', ...
    'FontName','Arial', ...
    'FontSize',0.35, ...
    'Position',[0.4325 0.132325 0.12 0.0604915], ...
    'String',num2str(dMS), ...
    'Style','edit', ...
    'Tag','EditText2');

%Cuadro de texto y texto editable donde se introduce la distancia entre
el
%altavoz y el micrófono en m, dLM.
b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'FontName','Arial', ...
    'FontSize',0.15, ...
    'Position',[0.6425 0.374291 0.30875 0.249527], ...
    'String',mat2, ...
    'Style','text', ...
    'Tag','StaticText3');

b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'Callback','dLM=str2num(get(gcbo, 'String'));', ...
    'FontName','Arial', ...
    'FontSize',0.35, ...
    'Position',[0.74125 0.281664 0.12 0.0604915], ...
    'Style','edit', ...
    'Tag','EditText2');

%Cuadro de texto donde aparece la distancia entre altavoz y micrófono
óptima
%para obtener la mínima frecuencia fiable más baja. Esta distancia se
calcula
%con la función alt_mic.
b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'FontName','Arial', ...
    'FontSize',0.5, ...
    'FontWeight','bold', ...
    'ForegroundColor',[1 0 0], ...
    'Position',[0.84 0.375 0.07625 0.0548204], ...
    'String',num2str(alt_mic(dSE,dMS),3), ...
    'Style','text', ...
    'Tag','StaticText1');

```

## rla.m

```
%Pushbutton que llama a la siguiente pantalla gráfica, r12. Se debe
pulsar una
%vez que ya están introducidos todos los datos. Se inicializa la variable
i a 1.
b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'Callback','sound(clic,22050,16);i=1;r12(n,i);', ...
    'FontName','Arial', ...
    'FontSize',0.375, ...
    'Position',[0.73 0.10397 0.14 0.102079], ...
    'String','SEGUIR', ...
    'Tag','Pushbutton1');
```

## alt\_mic.m

```
function alt_mic= alt_mic(dSE,dMS);
%alt_mic= alt_mic(dSE, dMS)
%
% Cálculo de la distancia altavoz-micrófono óptima para el método
%Adrienne
%
%Parámetros de entrada:
% dSE, semialtura de la barrera, en metros.
% dMS, distancia micrófono-barrera, en m.
%
%Parámetro de salida:
% dLM, distancia altavoz-micrófono óptima, en m.

%Se calcula la distancia altavoz-micrófono para obtener una resolución en
%frecuencia óptima, según P.Cobo.

alt_mic= (sqrt((12*dMS+(4*sin(40*pi/180)*dSE)).^2+(128*(dSE^2-dMS.^2)))-
...
(12*dMS+4*sin(40*pi/180)*dSE))/16;
```

## rl2.m

```
function rl2(n,i)
%[]=rl2(n,i);
%
% Pantalla gráfica donde se introducen las distintas señales reflejadas
medidas,
%en función del punto de estudio y del ángulo de incidencia. En cada
punto
%característico se estudia la señal reflejada para ángulos de incidencia
de
%50, 60, 70, 80, 90, 100, 110, 120 y 130 grados sexagesimales.
%
%Parámetros de entrada:
% n, número de puntos que vamos a estudiar en la barrera.
% i, variable necesaria para la siguiente pantalla gráfica, rl2.
%     Es el punto "actual" en el que se está introduciendo las señales.
%
%Parámetros de salida:
% matr_ref, matriz que contiene, en filas, cada una de las señales
%     reflejadas medidas.

%Se cierra la anterior pantalla gráfica.
close;

%Se carga la variable asociada a rl2 que contiene datos necesarios para
la
%correcta ejecución de la pantalla gráfica.
load rl2 ;

%Parámetros generales de la pantalla gráfica.
a= figure('Units','normalized', ...
'Color',[0.95 0.95 0.8], ...
'Colormap',mat0, ...
'PointerShapeCData',mat1, ...
'Position',[0 0.05 1 0.88125], ...
'Tag','Fig1');

%Cuadro de texto que contiene el título de la pantalla gráfica.
b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'FontName','Arial', ...
'FontSize',0.547009, ...
'Position',[0.195761 0.91573 0.587282 0.0543071], ...
'String','Pérdida por reflexión (RL):', ...
'Style','text', ...
'Tag','StaticText1');

%Cuadro de texto(compuesto por dos cuadros de texto) donde se indica que
se
%introduzcan las señales reflejadas para los distintos ángulos. El
segundo
%cuadro de texto nos informa del punto de estudio.
b = uicontrol('Parent',a, ...
'Units','normalized', ...
```

```

'FontUnits','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'FontName','Arial', ...
'FontSize',0.18, ...
'Position',[0.07875 0.404537 0.3575 0.196597], ...
'String','mat2', ...
'Style','text', ...
'Tag','StaticText2');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.752941 0.75 0.752941], ...
'FontName','Arial', ...
'FontSize',0.6, ...
'FontWeight','bold', ...
'ForegroundColor',[1 0 0], ...
'Position',[0.38375 0.426415 0.02875 0.0471698], ...
'String',num2str(i), ...
'Style','text', ...
'Tag','StaticText2');

%Cuadros de texto donde aparecen los distintos ángulos de incidencia de
la
%señal acústica. Estos ángulos son:
% 50, 60, 70, 80, 90, 100, 110, 120 y 130 grados sexagesimales.
b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'FontName','Arial', ...
'FontSize',0.5, ...
'Position',[0.5075 0.759924 0.05 0.05], ...
'String','50°', ...
'Style','text', ...
'Tag','StaticText2');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'FontName','Arial', ...
'FontSize',0.5, ...
'Position',[0.5075 0.678639 0.05 0.05], ...
'String','60°', ...
'Style','text', ...
'Tag','StaticText2');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'FontName','Arial', ...
'FontSize',0.5, ...
'Position',[0.5075 0.597353 0.05 0.05], ...
'String','70°', ...
'Style','text', ...
'Tag','StaticText2');

```



```
b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'FontName','Arial', ...
    'FontSize',0.5, ...
    'Position',[0.5075 0.516068 0.05 0.05], ...
    'String','80°', ...
    'Style','text', ...
    'Tag','StaticText2');

b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'FontName','Arial', ...
    'FontSize',0.5, ...
    'Position',[0.5075 0.434783 0.05 0.05], ...
    'String','90°', ...
    'Style','text', ...
    'Tag','StaticText2');

b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'FontName','Arial', ...
    'FontSize',0.5, ...
    'Position',[0.5075 0.353497 0.05 0.05], ...
    'String','100°', ...
    'Style','text', ...
    'Tag','StaticText2');

b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'FontName','Arial', ...
    'FontSize',0.5, ...
    'Position',[0.5075 0.274102 0.05 0.05], ...
    'String','110°', ...
    'Style','text', ...
    'Tag','StaticText2');

b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'FontName','Arial', ...
    'FontSize',0.5, ...
    'Position',[0.5075 0.192817 0.05 0.05], ...
    'String','120°', ...
    'Style','text', ...
    'Tag','StaticText2');

b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'FontName','Arial', ...
```

```

'FontSize',0.5, ...
'Position',[0.5075 0.111531 0.05 0.05], ...
'String','130°', ...
'Style','text', ...
'Tag','StaticText2');

%Textos editables donde se introduce, en cada uno de ellos, la señal
medida
%en un punto determinado y para un ángulo de incidencia determinado. Cada
%señal introducida pasará a ser una fila de la matriz mat_ref,
independientemente
%de si se ha introducido como fila o como columna. Esto último se
consigue
%gracias a la función fila.m.
b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'Callback','matr_ref(9*i-8,:)=fila(eval(get(gcbo, 'String')));', ...
    'FontName','Arial', ...
    'FontSize',0.4, ...
    'Position',[0.5675 0.759546 0.09 0.05], ...
    'String','', ...
    'Style','edit', ...
    'Tag','EditText1');

b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'Callback','matr_ref(9*i-7,:)=fila(eval(get(gcbo, 'String')));', ...
    'FontName','Arial', ...
    'FontSize',0.4, ...
    'Position',[0.56875 0.678544 0.09 0.05], ...
    'String','', ...
    'Style','edit', ...
    'Tag','EditText1');

b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'Callback','matr_ref(9*i-6,:)=fila(eval(get(gcbo, 'String')));', ...
    'FontName','Arial', ...
    'FontSize',0.4, ...
    'Position',[0.56875 0.597543 0.09 0.05], ...
    'String','', ...
    'Style','edit', ...
    'Tag','EditText1');

b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'Callback','matr_ref(9*i-5,:)=fila(eval(get(gcbo, 'String')));', ...
    'FontName','Arial', ...
    'FontSize',0.4, ...
    'Position',[0.56875 0.516541 0.09 0.05], ...
    'String','', ...

```

```

'Style','edit', ...
'Tag','EditText1');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[1 1 1], ...
'Callback','matr_ref(9*i-4,:)=fila(eval(get(gcbo, 'String')));', ...
'FontName','Arial', ...
'FontSize',0.4, ...
'Position',[0.56875 0.435539 0.09 0.05], ...
'String','', ...
'Style','edit', ...
'Tag','EditText1');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[1 1 1], ...
'Callback','matr_ref(9*i-3,:)=fila(eval(get(gcbo, 'String')));', ...
'FontName','Arial', ...
'FontSize',0.4, ...
'Position',[0.56875 0.354537 0.09 0.05], ...
'String','', ...
'Style','edit', ...
'Tag','EditText1');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[1 1 1], ...
'Callback','matr_ref(9*i-2,:)=fila(eval(get(gcbo, 'String')));', ...
'FontName','Arial', ...
'FontSize',0.4, ...
'Position',[0.56875 0.273535 0.09 0.05], ...
'String','', ...
'Style','edit', ...
'Tag','EditText1');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[1 1 1], ...
'Callback','matr_ref(9*i-1,:)=fila(eval(get(gcbo, 'String')));', ...
'FontName','Arial', ...
'FontSize',0.4, ...
'Position',[0.56875 0.192533 0.09 0.05], ...
'String','', ...
'Style','edit', ...
'Tag','EditText1');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[1 1 1], ...
'Callback','matr_ref(9*i,:)=fila(eval(get(gcbo, 'String')));', ...
'FontName','Arial', ...
'FontSize',0.4, ...
'Position',[0.56875 0.111531 0.09 0.05], ...
'String','', ...

```

```
'Style','edit', ...  
'Tag','EditText1');
```

```
%Pushbutton gracias al cual podemos comprobar el aspecto de las señales  
que  
%hemos introducido en cada punto. Al pulsarlo, se llama la función  
ecograma,  
%que representa en un mismo gráfico todas las señales introducidas para  
un  
%mismo punto.
```

```
b = uicontrol('Parent',a, ...  
    'Units','normalized', ...  
    'FontUnits','normalized', ...  
    'Callback','sound(clic,22050,16); ecograma(matr_ref,i,dt,n);', ...  
    'FontName','Arial', ...  
    'FontSize',0.3, ...  
    'FontWeight','bold', ...  
    'Position',[0.76875 0.459357 0.15 0.0793951], ...  
    'String','COMPROBAR', ...  
    'Tag','Pushbutton1');
```

## ecograma.m

```
function ecograma(matr_ref,i,dt,n);
%ecograma(matr_ref,i,dt,n);
%
% Pantalla gráfica donde aparece el ecograma de las señales
introducidas,
%para un punto, en rl2. De este modo se puede comprobar si nos hemos
confundido
%al introducirlas. Aparecen, para un mismo punto, las señales medidas
para
%los ángulos de incidencia: 50°, 60°, 70°, 80°, 90°, 100°, 110°, 120° y
130°.
%
%Parámetros de entrada:
% matr_ref, matriz que contiene las señales medidas.
% i, punto donde se han realizado las medidas.
% dt, periodo de muestreo, en ms.
% n, número de puntos característicos de la barrera acústica.
%
%No tiene parámetros de salida.

%Se cierra la netrior pantalla gráfica.
close;

%Se carga la variable que contiene los datos necesario para que se
ejecute
%correctamnte la pantalla gráfica.
load ecograma

%Parámetros generales de la pntalla gráfica.
a = figure('Units','normalized', ...
'Color',[0.95 0.95 0.8], ...
'Colormap',mat0, ...
'PointerShapeCData',mat1, ...
'Position',[-0.0025 0.05 1.0025 0.89], ...
'RendererMode','manual', ...
'Tag','Fig1');

%Parámetros de los ejes de la representación gráfica.
b = axes('Parent',a, ...
'FontUnits','normalized', ...
'Color',[1 1 1], ...
'Position',[0.122195 0.0468165 0.678304 0.78839], ...
'Tag','Axes1', ...
'TickLength',[0 0]);

%Se eligen las filas de la matriz que se deben representar y se
transponen,
%ya que la función ecog.m representa columnas.
matriz=matr_ref((9*i-8):9*i,:);

%Se hace la representación gráfica del ecograma, gracias a la función
ecog.m.
ecog(matriz,dt,1,size(matr_ref,2)/2,100);
```

```

%Título del eje horizontal.
b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'BackgroundColor',[0.95 0.95 0.8], ...
    'FontName','Arial', ...
    'FontSize',0.5, ...
    'FontWeight','bold', ...
    'ForegroundColor',[0 0 1], ...
    'Position',[0.4 0 0.08 0.0299625], ...
    'String','tiempo', ...
    'Style','text', ...
    'Tag','StaticText3');

%Cuadros de texto que nos informan del punto en el que se está haciendo
la
%comprobación de las señales introducidas.
b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'FontName','Arial', ...
    'FontSize',0.42, ...
    'Position',[0.12 0.89 0.65 0.06], ...
    'String','Comprobación de las señales introducidas, para el punto:',
    ...
    'Style','text', ...
    'Tag','StaticText1');

b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'FontName','Arial', ...
    'FontSize',0.4, ...
    'ForegroundColor',[1 0 0], ...
    'Position',[0.796758 0.89 0.053616 0.06], ...
    'String',num2str(i), ...
    'Style','text', ...
    'Tag','StaticText2');

% Cuadros de texto en los que aparecen los ángulos correspondientes a la
%representación de cada señal. Cada señal corresponde a un ángulo de
inci-
%dencia determinado.
b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'BackgroundColor',[0.95 0.95 0.8], ...
    'FontName','Arial', ...
    'FontSize',0.5, ...
    'FontWeight','bold', ...
    'ForegroundColor',[0 0 1], ...
    'Position',[0.0798005 0.741573 0.0286783 0.0299625], ...
    'String','130°', ...
    'Style','text', ...
    'Tag','StaticText3');

```

```

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.95 0.95 0.8], ...
'FontName','Arial', ...
'FontSize',0.48, ...
'FontWeight','bold', ...
'ForegroundColor',[0 0 1], ...
'Position',[0.0798005 0.659176 0.0286783 0.0299625], ...
'String','120°', ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.95 0.95 0.8], ...
'FontName','Arial', ...
'FontSize',0.48, ...
'FontWeight','bold', ...
'ForegroundColor',[0 0 1], ...
'Position',[0.0798005 0.576779 0.0286783 0.0299625], ...
'String','110°', ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.95 0.95 0.8], ...
'FontName','Arial', ...
'FontSize',0.48, ...
'FontWeight','bold', ...
'ForegroundColor',[0 0 1], ...
'Position',[0.0798005 0.492509 0.0286783 0.0299625], ...
'String','100°', ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.95 0.95 0.8], ...
'FontName','Arial', ...
'FontSize',0.5, ...
'FontWeight','bold', ...
'ForegroundColor',[0 0 1], ...
'Position',[0.0798005 0.410112 0.0286783 0.0299625], ...
'String','90°', ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.95 0.95 0.8], ...
'FontName','Arial', ...
'FontSize',0.5, ...
'FontWeight','bold', ...
'ForegroundColor',[0 0 1], ...

```

## ecograma.m

```
'Position',[0.0798005 0.327715 0.0286783 0.0299625], ...
'String','80°', ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.95 0.95 0.8], ...
'FontName','Arial', ...
'FontSize',0.5, ...
'FontWeight','bold', ...
'ForegroundColor',[0 0 1], ...
'Position',[0.0798005 0.243446 0.0286783 0.0299625], ...
'String','70°', ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.95 0.95 0.8], ...
'FontName','Arial', ...
'FontSize',0.5, ...
'FontWeight','bold', ...
'ForegroundColor',[0 0 1], ...
'Position',[0.0798005 0.161049 0.0286783 0.0299625], ...
'String','60°', ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.95 0.95 0.8], ...
'FontName','Arial', ...
'FontSize',0.5, ...
'FontWeight','bold', ...
'ForegroundColor',[0 0 1], ...
'Position',[0.0798005 0.0786517 0.0286783 0.0299625], ...
'String','50°', ...
'Style','text', ...
'Tag','StaticText3');

%Pushbutton que hace que se retorne a la pantalla gráfica anterior, rl2,
%donde se introducen las señales medidas. Se debe pulsar en caso de que,
%viendo el ecograma, se tenga el conocimiento de que ha habido cualquier
%error a la hora de introducir las señales.
b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'FontName','Arial', ...
'FontSize',0.27, ...
'Position',[0.831671 0.485019 0.150873 0.117978], ...
'String','CORREGIR', ...
'Tag','Pushbutton1',...
'Callback','sound(clic,22050,16); rl2(n,i);');
```



## ecograma.m

```
%Algoritmo para introducir por pantalla las señales reflejadas medidas en
el
%siguiente punto, en caso de no haber introducido las señales medidas en
todos
%los puntos. En caso de haber introducido todas las señales para todos
los
%puntos, aparece un push button "CALCULAR" que hace que corran las
funciones
%necesarias para calcular la pérdida por reflexión (RL) del dispositivo
%antirruído. Mientras se están realizando los cálculos, aparecerá la
pantalla
%gráfica espere.m.
if i < n
    b = uicontrol('Parent',a, ...
        'Units','normalized', ...
        'FontUnits','normalized', ...
        'FontName','Arial', ...
        'FontWeight','Bold',...
        'FontSize',0.27, ...
        'Position',[0.830424 0.219101 0.150873 0.117978], ...
        'String','SIG.PUNTO', ...
        'Tag','Pushbutton1',...
        'Callback','sound(clic,22050,16); i=i+1; rl2(n,i);');
else
    b = uicontrol('Parent',a, ...
        'Units','normalized', ...
        'FontUnits','normalized', ...
        'FontName','Arial', ...
        'FontWeight','Bold',...
        'FontSize',0.27, ...
        'Position',[0.830424 0.219101 0.150873 0.117978], ...
        'String','CALCULAR', ...
        'Tag','Pushbutton1',...
        'Callback','sound(clic,22050,16); clear i nom_dir;
espere;pause(.01);[ref_loss,RL5090,RL50130,RL90130]=nucleo(matr_ref,directa,dSE,dMS,dLM,dt);');
end;
```

## ecog.m

```
function ecog(A,dt,i1,i2,solp)
%ecog(A,dt,i1,i2,solp)
%
%      Dada la matriz de datos A(m,n) dibuja el ecograma(para las
columnas)
%en el intervalo (i1,i2) ,con un solapamiento dado por solp (en tanto por
%ciento); dt es el periodo de muestreo (ms).
%
%Parámetros de entrada:
% A, matriz de datos.
% dt, periodo de muestreo, en ms.
% i1, fila inicial de la columna que queremos dibujar.
% i2, fila final de la columna que queremos dibujar.
% solp, solapamiento, en %.
%
%No tiene parámetros de salida.
%
%
%Pedro Cobo.

[m,n]=size(A);
xmed=zeros(m,1);

for i=1:n;
    xmed=xmed+A(:,i);
end

xmed=xmed/n;

m1=max(xmed);m2=min(xmed);m3=m1-m2;

i3=i2-i1+1;t=(1:i3)*dt;m4=n*solp*0.01*m3;

plot(t,A(i1:i2,1),'b'),hold on;

axis([0,i3*dt/2,2*m2,m4]);

for i=2:n
    x=A(:,i)+((i-1)*m3*solp*0.01);
    plot(t,x(i1:i2),'b')
end

hold off;
```

## espere.m

```
function espere()
%Pantalla gráfica en la que aparece el mensaje "Calculando. Espere, por
favor"
% Aparece mientras se están ejecutando los cálculos para la obtención de
la
% pérdida acústica por reflexión.(RL).

%Se cierra la netrior pantalla gráfica.
close;

%Se carga la variable que contiene los datos necesarios para que la
pantalla
%se ejecute adecuadamente.
load espere;

%Parámetros generales de la pantalla gráfica.
a = figure('Units','normalized', ...
    'Color',[0.95 0.95 .8], ...
    'Colormap',mat0, ...
    'PointerShapeCData',mat1, ...
    'Position',[-0.0025 0.05 1.005 0.89], ...
    'Tag','Fig1');

%Cuadro de texto en el que se comunica al usuario que sea tan amable de
%esperar mientras se ejecutan los cálculos.
b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'FontName','Arial', ...
    'FontSize',0.25, ...
    'Position',[0.298507 0.303371 0.400498 0.395131], ...
    'String','Calculando. Espere, por favor.', ...
    'Style','text', ...
    'Tag','StaticText1');
```

### 1.3.1.2. FUNCIONES DE CÁLCULO.

Estas funciones son las que realmente implementan el método Adrienne descrito en el apartado "1.1.3.2. MÉTODO ADRIENNE".

**nucleo** es la función principal de cálculo. El resto de las funciones que aparecen en este apartado son subfunciones de **nucleo**.

## nucleo.m

```
function
[ref_loss,RL5090,RL50130,RL90130,MinFrec]=nucleo(matr_ref,directa,dSE,dMS
,dLM,dt);
%[ref_loss,RL5090,RL50130,RL90130,MinFrec]=nucleo(matr_ref,directa,dSE,dM
S,dLM,dt);
%
% Calcula la pérdida por reflexión de nuestro dispositivo antirruído
%aplicando el algoritmo Adrienne.
%
%Parámetros de entrada:
% matr_ref; matriz que contiene, en filas, cada una de las señales
% reflejadas medidas para un punto determinado, en ángulo de
% incidencia determinado.
% directa; vector fila que contiene la señal directa.
% dSE; semialtura de la barrera, en m.
% dMS; distancia entre el micrófono y la barrera, en m.
% dLM; distancia entre el altavoz y el micrófono, en m.
% dt; periodo de muestreo en ms.
%
%Parámetros de salida:
% ref_loss; pérdida por reflexión para cada una de las frecuencias dadas
% por las siguientes 18 bandas separadas un tercio de octava:
% [100 125 160 200 250 315 400 500 630 800 1000 1250 1600
% 2000 2500 3150 4000 5000] Hz.
% RL5090; promedio de la pérdida por reflexión (RL), para las distintas
% bandas de tercio de octava, para los ángulos comprendidos
% entre 50 y 90 grados.
% RL50130; promedio de la pérdida por reflexión (RL), para las distintas
% bandas de tercio de octava, para los ángulos comprendidos
% entre 50 y 130 grados.
% RL90130; promedio de la pérdida por reflexión (RL), para las distintas
% bandas de tercio de octava, para los ángulos comprendidos
% entre 90 y 130 grados.
% MinFrec; mínima frecuencia para la cual los valores obtenidos para
% la pérdida por reflexión son válidos.
%
% 1999. Adolfo Corral.

%Nº de columnas, número de puntos que contiene el vector medido.
num_datos=size(matr_ref,2);

%Velocidad del sonido, en m/ms
c=.34;

%Se calcula el vector para realizar la corrección por divergencia de la
%onda acústica. La intensidad acústica decrece con (1/(distancia
%avanzada)).
%El vector de corrección por divergencia es "div".
v_tiempo=[];
for i=1:num_datos
    v_tiempo = [v_tiempo, i*dt];
end;
div=c*v_tiempo;

%Se halla el incremento de frecuencia(Hz) entre dos puntos de la función
%de la frecuencia en banda estrecha.
```

## nucleo.m

```
delta=1/(num_datos*dt*10^(-3));
```

```
%Se calcula, en función de la disposición geométrica de la barrera, la  
%longitud temporal de la ventana Adrienne, el tiempo que tarda en llegar  
%la onda directa al altavoz y el tiempo que tarda en llegar la señal  
%reflejada. Todos estos tiempos están en ms. Esto se consigue con la  
%función tdel.
```

```
[delay,taud,taur]=tdel(dSE,dLM,dMS);
```

```
%Se calcula la ventana definida en el método Adrienne por la función  
%wadri.m
```

```
vent=wadri(delay,dt);
```

```
%Se calculan las ventanas para la señal directa y para la señal refle-  
%jada. Se hace con la función ventana.
```

```
vent_dir=ventana(vent,taud,num_datos,dt);
```

```
vent_ref=ventana(vent,taur,num_datos,dt);
```

```
%Se define el vector tiempos, necesario posteriormente para realizar  
%una representación gráfica de las señales con la ventana, mediante  
%la función ventaneo.m.
```

```
tiempos=[delay, taud, taur, dt];
```

```
%Algoritmo definido en el método Adrienne para calcular la pérdida por  
%reflexión en un dispositivo antirruído. Se aplica para cada punto de  
estudio
```

```
%determinado de la barrera, con un ángulo de incidencia determinado.
```

```
for i=1:size(matr_ref,1)
```

```
    %Se alinean la señal reflejada y la directa, con la función alinear.m.
```

```
    %Es decir, se hacen coincidir los máximos.
```

```
    directa_d=alinear(directa, matr_ref(i,:));
```

```
    %Se aplica el método de la sustracción para obtener la señal reflejada  
    %pura. Basta, simplemente, con restar la señal directa de la reflejada  
    %en cuestión.
```

```
    reflejada= matr_ref(i,:)-directa_d;
```

```
    %Se realiza la representación gráfica de las señales con sus  
    %ventanas. Se hace dentro del bucle porque cada vez tenemos una  
    %señal reflejada distinta.
```

```
    ventaneo(i,directa,vent_dir,reflejada,vent_ref,v_tiempo, tiempos);
```

```
    %Se calcula el parámetro RL como viene definido en el método Adrienne.
```

```
    %Lo calculamos para cada señal, luego ya se harán los promedios
```

```
    %oportunos. Se calculan el numerador y el denominador por separado y
```

```
    %luego se dividen. Se dividen las respuestas en frecuencia en bandas
```

```
    %de tercio de octava, como dicta el método Adrienne. Para hacer la
```

```
    %transformación de funciones de la frecuencia en banda estrecha a
```

```
bandas
```

```
    %de tercio de octava se usa la función bandas.m.
```

## nucleo.m

```
num= abs(fft(div.*reflejada.*vent_ref)).^2;
den= abs(fft(div.*directa.*vent_dir)).^2;
rl(i,:)= bandas(num,delta)./ bandas(den,delta);
end;

%A partir de la matriz rl, que contiene tantas filas como medidas
%realizadas, se calcula la matriz ref_loss, que tendrá sólo nueve filas,
%cada una representativa del ángulo de incidencia de la señal acústica.
%Tiene 18 columnas, tantas como bandas de tercio de octava hay definidas
%en el método Adrienne.

%Se inicializa la matriz ref_loss.
ref_loss(9,18)=0;

%Se aplica el algoritmo para construirla. Consiste, simplemente, en
realizar
%un promedio de la pérdida por reflexión para los distintos ángulos. Se
hace
%el promedio entre señales medidas, para el mismo ángulo, en distintos
puntos
%de la barrera acústica.
for i=1:9
    for j=1:(size(matr_ref,1)/9)
        ref_loss(i,:)=ref_loss(i,:)+rl(9*j-9+i,:);
    end;
end;

ref_loss= ref_loss/(size(matr_ref,1)/9);

%A su vez, se hace el promedio de la pérdida por reflexión para los
ángulos
%comprendidos entre 50 y 90 grados, entre 50 y 130 grados y entre 90 y
130
%grados. Este es el resultado final de todo el proceso de cálculo.
RL5090=(1/5)*(ref_loss(1,:)+ref_loss(2,:)+ref_loss(3,:)+ref_loss(4,:)+ref
_loss(5,:));
RL50130=(1/9)*(ref_loss(1,:)+ref_loss(2,:)+ref_loss(3,:)+ref_loss(4,:)+..
.
ref_loss(5,:)+ref_loss(6,:)+ref_loss(7,:)+ref_loss(8,:)+ref_loss(9,:));
RL90130=(1/5)*(ref_loss(5,:)+ref_loss(6,:)+ref_loss(7,:)+ref_loss(8,:)+re
f_loss(9,:));

%Se realiza un gráfico en representación continua, comparando los tres
%promedios. Se hace con la función graf_con.m.
graf_con(RL5090,RL90130,RL50130);

%Se hace el gráfico de barras para los distintos promedios de pérdida por
%reflexión. Se hace con la función graf_bar.m.
graf_bar(RL5090,RL90130,RL50130);

%Se calcula la mínima frecuencia fiable, que es la mínima frecuencia para
%la cual los resultados obtenidos son válidos. La función min_frec.m
```

## nucleo.m

```
%también representa la ventana Adrienne y su transformada de Fourier, a  
%partir de la cual se obtiene el valor de la mínima frecuencia.  
MinFrec=min_frec(vent,dt);
```



## aLinear.m

```
function Ad=aLinear(A,B)
%Ad=aLinear(A,B)
%
%  Dados dos vectores A y B, desplaza el vector A, de modo que los
máximos
%de los dos vectores se dan para la misma posición dentro del vector.
%Los vectores A y B han de tener el mismo número de datos.
%
%Parámetros de entrada:
% A.
% B.
%
%Parámetros de salida:
% Ad, vector A desplazado.

%Cálculo de la posición que ocupa, en el vector A, el valor máximo.
posmaxA=1;
for i= 1:length(A)
    if A(i)>A(posmaxA)
        posmaxA=i;
    end
end;

%Cálculo de la posición que ocupa, en el vector B, el valor máximo.
posmaxB=1;
for i= 1:length(B)
    if B(i)>B(posmaxB)
        posmaxB=i;
    end
end;

%Algoritmo para desplazar el vector A a la derecha. Los valores que
ocupaban
%antes las últimas posiciones pasan a ocupar las primeras.
if (posmaxA-posmaxB)<0
    for i=1:(posmaxB-posmaxA)
        Ad(i)=A(length(A)-(posmaxB-posmaxA-i));
    end
    for i= (1+posmaxB-posmaxA):length(A)
        Ad(i)=A(i+posmaxA-posmaxB);
    end
end

%Algoritmo para desplazar el vector A a la izquierda. Los valores que
ocupaban
%antes las primeras posiciones pasan a ocupar las últimas. También se
contempla
%aquí el caso de que los vectores A y B tengan el máximo para la misma
posición
%del vector, en cuyo caso A permanece igual.
if (posmaxA-posmaxB)>=0
    for i=1:(length(A)-(posmaxA-posmaxB))
        Ad(i)=A(i+posmaxA-posmaxB);
    end
    for i=(length(A)-(posmaxA-posmaxB)+1):length(A)
        Ad(i)= A(posmaxA-posmaxB-(length(A)-i));
    end
end
```

```
end  
end
```

## tdel.m

```
function [tdel,taud,taur]=tdel(dSE,dLM,dMS);
%[tdel,taud,taur]=tdel(dSE,dLM,dMS);
%
% Halla la longitud temporal de la ventana Adrienne, así como el tiempo,
%en ms, que tarda en llegar la onda reflejada en incidencia normal. Estos
%parámetros se calculan a partir de los parámetros geométricos del
%dispositivo antirruído.
%
%Parámetros de entrada:
% dSE, semialtura de la barrera acústica, en m.
% dLM, distancia altavoz-micrófono, en m.
% dMS, distancia micrófono-barrera, en m.
%
%Parámetros de salida:
% tdel, longitud temporal de la ventana Adrienne, en ms.
% taud, tiempo que tarda en llegar la onda directa, en ms.
% taur, tiempo que tarda en llegar la onda reflejada en incidencia
% normal, en ms.

%Distancia entre el altavoz y la barrera antirruído.
dLS=dLM+dMS;

%Velocidad del sonido en m/ms.
c=.34;

%Ángulo alpha.
alpha=atan(dSE/dLS);

%Vector de ángulos de incidencia de la señal acústica, en rad.
theta=[-40 -30 -20 -10 0 10 20 30 40]*(pi/180);

%Tiempo que tarda en llegar la onda directa, en ms.
taud=dLM/c;

%Tiempo que tarda en llegar la onda reflejada en incidencia normal, en
ms.
taur=sqrt(4*(dLS^2)+(dLM^2)-(4*dLS*dLM))/c;

%Tiempo que tarda en llegar la onda difractada en el altavoz, en ms.
taul=(2*dLS+dLM)/c;

%Tiempos que tardan en llegar la onda difractada en el borde de la
barrera
%y la onda reflejada en el suelo, en ms. Estos tiempos se calculan para
los
%distintos ángulos de incidencia de la señal acústica.
for i=1:9
    taue(i)=(sqrt(dLS^2+dSE^2)+sqrt(dLS^2+dSE^2+dLM^2-...
        2*dLM*sqrt(dLS^2+dSE^2)*cos(alpha+theta(i))))/c; %criterio de
ángulos.
    tauf(i)=sqrt(4*(dSE^2)+(dLM^2)-(4*dSE*dLM*sin(theta(i))))/c;
end;
```

**tdel.m**

```
%Longitud temporal de la ventana Adrienne, en ms.  
tdel=min([taul,taue,tauf])-taur;
```

## wadri.m

```
function w=wadri(tdel,dt)
% w=wadri(tdel,dt)
%
% Cálculo de la ventana Adrienne, wadri, a partir de los datos del
% periodo de muestreo, dt (en ms), y retardo temporal, tdel (en ms)
%
%Parámetros de entrada:
% tdel, longitud temporal de la ventana Adrienne, en ms.
% dt, periodo de muestreo, en ms.
%
%Parámetros de salida:
% w, vector fila que contiene la ventana Adrienne.

%Cálculo del número de puntos en cada tramo de la ventana.
n1=round(0.5/dt);
n2=round((0.7*tdel)/dt);
n3=round((0.3*tdel)/dt);

%Cálculo de cada tramo de la ventana.
w1=nutwin(2*n1,4);
w2=nutwin(n2,1);
w3=nutwin(2*n3,4);

%Ventana Adrienne.
w=[w1(1:n1)' w2' w3(n3+1:2*n3)'];
```

## NUTWIN.m

```

function x=nutwin(nw,iw)
%x=nutwin(nw,iw)
%
%  Calcula una ventana de Nuttall de nw puntos definida por:
%
%   $x(t)=a_0+a_1\cos(\pi 2t/T)+a_2\cos(2\pi 2t/T)+a_3\cos(3\pi 2t/T)$ ,
%
%  donde  $|t| > T/2$ , siendo  $nw*dt = T$  la longitud de la ventana.
%
%  Los coeficientes {a0,a1,a2,a3} dependen del tipo de ventana
%
%iw          a0          a1          a2          a3
%-----
%1 (RECT)    1          0          0          0
%2 (HANN)    0.5        0.5        0          0
%3 (HAMM)    0.53836   0.461614  0          0
%4 (BLACK)   0.42       0.5        0.08       0
%5 (NUT98)   0.3635819  0.4891775  0.1365995  0.00106411
%
%
%      Pedro Cobo Parra. Instituto de Acústica.
%      22 de Noviembre de 1993.

n2=round(nw/2);pi2n=pi/nw;x=zeros(nw,1);

if iw==1
    a0=1;a1=0;a2=0;a3=0;
elseif iw==2
    a0=0.5;a1=0.5;a2=0;a3=0;
elseif iw==3
    a0=0.53836;a1=0.461614;a2=0;a3=0;
elseif iw==4
    a0=0.42;a1=0.5;a2=0.08;a3=0;
elseif iw==5
    a0=0.3635819;a1=0.4891775;a2=0.1365995;a3=0.00106411;
else
    error('iw debe ser un entero entre 1 y 5');
end

for i=1:nw
    fase=(i-n2)*pi2n;
    x(i)=a0+(a1*cos(fase))+(a1*cos(2*fase))+(a3*cos(3*fase));
end

x=x/(max(x));

```

## ventana.m

```
function ventana=ventana(vent,tau,num_datos,dt);
%ventana=ventana(vent,tau,num_datos,dt);
%
% Sitúa la ventana obtenida por el método Adrienne 0.5 ms antes del
% tiempo dado por tau, en un vector con tantos elementos como num_datos.
El
% resto de los elementos serán ceros. De este modo, se recogerá el evento
% que comienza en tau y el vector tendrá la misma longitud que las señales
% medidas, de modo que dichos vectores se podrán multiplicar.
%
% Parámetros de entrada:
% vent, ventana obtenida por el método Adrienne.
% tau, tiempo de origen del evento que se quiere que recoja la ventana
% Adrienne.
% num_datos, longitud del vector que contiene la señal.
% dt, periodo de muestreo, en ms.
%
% Parámetros de salida:
% ventana, vector que contiene la ventana Adrienne en el lugar adecuado
% para recoger el evento que tiene su origen en tau.

% Se inicializa el vector ventana con los mismos elementos que las
% señales medidas y todos igual a cero.
ventana(num_datos)=0;

% Se coloca la ventana adecuadamente en el tiempo. Para ello, se calcula
% dónde ha de estar su origen y se empiezan a colocar los elementos de la
% ventana a partir de ahí. La ventana resultante queda en la variable
% "ventana".
origen=round((tau-0.5)/dt);
for j=1:length(vent);
    ventana(origen+j)=vent(j);
end;
```

## bandas.m

```
function tercio_oct=bandas(banda_estr,delta);
%tercio_oct=bandas(banda_estr,delta)
%
% Pasa una función de la frecuencia en banda estrecha a su equivalente
%en bandas de tercio de octava, desde 100Hz hasta 5000Hz
%
%Parámetros de entrada:
% banda_estr, función de la frecuencia en banda estrecha.
% delta, incremento de frecuencia entre dos puntos en
%banda estrecha.
%
%Parámetro de salida:
% tercio_oct, función equivalente en 18 bandas de tercio de octava de
% valores nominales [100,125,160,200,250,315,400,500,630,800,1000,
% 1250,1600,2000,2500,3150,4000,5000] Hz

%Valores reales de las bandas de tercio de octava.
bandas_exactas=[100 125.8925 158.4893 199.5262 251.1886 316.2278 ...
                398.1072 501.1872 630.9573 794.3282 1000 1258.9254 1584.8932 ...
                1995.2623 2511.8864 3162.2777 3981.0717 5011.8723];

%Valor de un sexto de octava.
sexto_oct=1.1220184543;

%Algoritmo de transformación de una función en banda estrecha a una
función
%en bandas de tercio de octava.
for i=1:18

    %Creación de variables necesarias
    acumulado=0;
    frec_inf = bandas_exactas(i)/sexto_oct;
    frec_sup = bandas_exactas(i)*sexto_oct;
    punto_inic_banda = floor(frec_inf/delta)+1;
    punto_fin_banda = floor(frec_sup/delta)+1;
    first_in = banda_estr(punto_inic_banda);
    prim_delta = delta*punto_inic_banda - frec_inf;
    last_in = banda_estr(punto_fin_banda);
    ult_delta = frec_sup - delta*(punto_fin_banda - 1);

    %Cálculo de los valores de cada banda
    for j=(punto_inic_banda+1):(punto_fin_banda-1)
        acumulado=acumulado+banda_estr(j);
    end;
    acumulado = acumulado + (prim_delta/delta)*first_in;
    acumulado = acumulado + (ult_delta/delta)*last_in;

    %Resultado
    tercio_oct(i)= acumulado;

end;
```



### 1.3.1.3. FUNCIONES DE SALIDA DE RESULTADOS.

El programa da como resultado 4 pantallas gráficas, donde el usuario puede extraer conclusiones. Estas 4 pantallas gráficas son:

- **ventaneo**, donde se puede comprobar si el ventaneo ha sido correctamente realizado.
- **graf\_con**, donde se comparan las pérdidas por reflexión para los tres promedios angulares.
- **graf\_bar**, donde se obtienen los valores de la pérdida por reflexión, que serán usados para obtener las propiedades absorbentes de la barrera, según EN-1793.
- **min\_frec**, donde se obtiene la mínima frecuencia pro debajo de la cual los resultados no son fiables.

## ventaneo.m

```
function
ventaneo(i,directa,vent_dir,reflejada,vent_ref,v_tiempo,tiempos);
%ventaneo(i,directa,vent_dir,reflejada,vent_ref,v_tiempo,tiempos);
%
% Pantalla gráfica donde se representan las señales con sus
%correspondientes ventanas superpuestas. El objetivo es que se
%compruebe que se ha hecho el ventaneo correctamente. Es una
%subfunción de la función núcleo.m.
%
%Parámetros de entrada:
% i, número que nos da la iteración de la función nucleo.m.
% directa, señal directa medida.
% vent_dir, ventana de la señal directa.
% reflejada, señal reflejada medida(distinta en cada iteración).
% vent_ref, ventana de la señal reflejada.
% v_tiempo, vector que contiene el tiempo transcurrido
%correspondiente a cada punto de las señales medidas.
% tiempos, vector que contiene variables temporales
%necesarias.
%
%No tiene parámetros de salida.

%Se carga la variable asociada a la pantalla gráfica, con variables
%asociadas a distintos parámetros gráficos.
load ventaneo;

%Se definen los orígenes y finales de las representaciones gráficas,
%de modo que, independientemente de la disposición geométrica de la
%barrera y del equipo de medida, se vea claramente la señal con
%su correspondiente ventana.
%tiempos=(delay, taud, taur, dt);

%Se define el origen y el fin para la señal directa.
or_dir=round((tiempos(2)-2.5)/tiempos(4));
fin_dir=round((tiempos(1)+tiempos(2)+2)/tiempos(4));

%Se define el origen y el fin para la señal reflejada.
or_ref=round((tiempos(3)-2.5)/tiempos(4));
fin_ref=round((tiempos(1)+tiempos(3)+2)/tiempos(4));

%Se definen ahora una serie de objetos que basta con definirlos
%en la primera iteración.
if i==1

    %Parámetros generales de la pantalla gráfica.
    a = figure('Color',[0.95 0.95 0.8], ...
        'Colormap',mat0, ...
        'PointerShapeCData',mat1, ...
        'Position',[-1 31 804 534], ...
        'Tag','Fig1');

    %Título de la pantalla gráfica.
    b = uicontrol('Parent',a, ...
```

```

'Units','normalized', ...
'FontUnits','normalized', ...
'FontName','Arial', ...
'FontSize',0.37, ...
'Position',[0.0223881 0.715356 0.466418 0.205993], ...
'String','Comprobación del VENTANEO', ...
'Style','text', ...
'Tag','StaticText1');

%Título de la representación para la señal directa.
b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'FontName','Arial', ...
'FontSize',0.5, ...
'Position',[0.662935 0.827715 0.14801 0.0524345], ...
'String','directa', ...
'Style','text', ...
'Tag','StaticText2');

%Título de la representación para la señal reflejada.
b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'FontName','Arial', ...
'FontSize',0.5, ...
'Position',[0.312189 0.569288 0.14801 0.0524345], ...
'String','reflejadas', ...
'Style','text', ...
'Tag','StaticText2');

%Se definen los ejes para la representación de la señal directa.
b = axes(...
'CameraUpVector',[0 1 0], ...
'Color',[1 1 1], ...
'ColorOrder',mat2, ...
'Position',[0.572139 0.593633 0.371891 0.215356], ...
'Tag','Axes2', ...
'XColor',[0 0 0], ...
'YColor',[0 0 0], ...
'ZColor',[0 0 0]);

%Se dibuja la señal directa y su ventana correpondiente, desde
%el punto or_dir hasta fin_dir.
plot(v_tiempo(or_dir:fin_dir) ,directa(or_dir:fin_dir) ,'b');hold on;
plot(v_tiempo(or_dir:fin_dir) ,max(directa)*vent_dir(or_dir:fin_dir)
,'r');

%Título del eje horizontal.
b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.95 0.95 0.8], ...
'FontName','Arial', ...

```

```

'FontSize',0.5, ...
'FontWeight','bold', ...
'ForegroundColor',[0 0 1], ...
'Position',[0.904 0.597 0.04 0.0299625], ...
'String','t(ms)', ...
'Style','text', ...
'Tag','StaticText3');

%Título del eje vertical.
b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'BackgroundColor',[0.95 0.95 0.8], ...
    'FontName','Arial', ...
    'FontSize',0.5, ...
    'FontWeight','bold', ...
    'ForegroundColor',[0 0 1], ...
    'Position',[0.574 0.778 0.1 0.0299625], ...
    'String','presión(bar)', ...
    'Style','text', ...
    'Tag','StaticText3');

%Se definen los ejes para la representación de las señales reflejadas.
b = axes(...
    'CameraUpVector',[0 1 0], ...
    'Color',[1 1 1], ...
    'ColorOrder',mat3, ...
    'Position',[0.200249 0.0374532 0.774876 0.507491], ...
    'Tag','Axes1', ...
    'XColor',[0 0 0], ...
    'YColor',[0 0 0], ...
    'ZColor',[0 0 0]);

%Título del eje horizontal.
b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'BackgroundColor',[0.95 0.95 0.8], ...
    'FontName','Arial', ...
    'FontSize',0.5, ...
    'FontWeight','bold', ...
    'ForegroundColor',[0 0 1], ...
    'Position',[0.935 0.04 0.04 0.0299625], ...
    'String','t(ms)', ...
    'Style','text', ...
    'Tag','StaticText3');

%Título del eje vertical.
b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'BackgroundColor',[0.95 0.95 0.8], ...
    'FontName','Arial', ...
    'FontSize',0.5, ...
    'FontWeight','bold', ...
    'ForegroundColor',[0 0 1], ...

```

## ventaneo.m

```
'Position',[0.202 0.513 0.1 0.0299625], ...
'String','presión(bar)', ...
'Style','text', ...
'Tag','StaticText3');

end;

%Para iteraciones posteriores a la primera se conserva la
%representación gráfica de la señal/es reflejada/s.
if i>1
    hold on;
end

%Se superpone la nueva representación de la señal reflejada
%a la/s que ya había (en el,caso de la primera no había ninguna)
%La representación se hace desde al punto or_ref hasta fin_ref.
plot(v_tiempo(or_ref:fin_ref),reflejada(or_ref:fin_ref),'b');hold on;
plot(v_tiempo(or_ref:fin_ref),max(reflejada)*vent_ref(or_ref:fin_ref),'r'
);
```

## graf\_con.m

```
function graf_con(RL5090,RL90130,RL50130);
%graf_con(RL5090,RL90130,RL50130)
%Realiza un gráfico en representación continua, comparando los tres
%promedios de la pérdida por reflexión(RL).
%
%Parámetros de entrada:
% RL5090; promedio de la pérdida por reflexión (RL), para las distintas
%   bandas de tercio de octava, para los ángulos comprendidos
%   entre 50 y 90 grados.
% RL50130;promedio de la pérdida por reflexión (RL), para las distintas
%   bandas de tercio de octava, para los ángulos comprendidos
%   entre 50 y 130 grados.
% RL90130;promedio de la pérdida por reflexión (RL), para las distintas
%   bandas de tercio de octava, para los ángulos comprendidos
%   entre 90 y 130 grados.
%
%No tiene parámetros de salida.

%La representación se tiene que hacer con los valores reales de la
%frecuencia central de cada uno de los tercios de banda. Aquí se
%definen.
bandas_exactas=[100 125.8925 158.4893 199.5262 251.1886 316.2278 ...
    398.1072 501.1872 630.9573 794.3282 1000 1258.9254 1584.8932 ...
    1995.2623 2511.8864 3162.2777 3981.0717 5011.8723];

%Parámetros generales de la figura.
a = figure('Units','normalized', ...
    'Color',[0.95 0.95 0.8], ...
    'Position',[-0.0025 0.0433333 1.0025 0.898333], ...
    'Tag','Fig1');

%Se definen los ejes.
axes('Units','Normalized','Position',[.1,.1,.75,.75]);

%Se hacen las representaciones gráficas de los tres promedios
%en representación logarítmica para las frecuencias.
semilogx(bandas_exactas,RL50130,'r.-');hold on;
semilogx(bandas_exactas,RL5090,'k.-');
semilogx(bandas_exactas,RL90130,'b.-');hold off;

%Se hace un mallado de la representación gráfica para poder leer
%más cómodamente los valores
grid on;

%Título del eje horizontal.
xlabel('Hz', ...
    'FontName','Arial', ...
    'FontUnits','Normalized', ...
    'FontWeight','Bold',...
    'FontSize',.05);

%Título del eje vertical.
ylabel('RL',...
```

## graf\_con.m

```
'FontName','Arial', ...
'FontUnits','Normalized', ...
'FontWeight','Bold', ...
'FontSize',.05);

%Título del gráfico.
b=uicontrol('Parent',a,...
  'String','Comparación de la RL para los distintos ángulos de
incidencia', ...
  'FontName','Arial', ...
  'FontUnits','Normalized', ...
  'FontWeight','Bold', ...
  'FontSize',.49,...
  'BackgroundColor',[0.752941 0.752941 0.752941], ...
  'Style','text',...
  'Units','Normalized',...
  'Position',[.05,.9,.9,.07]);

%Cuadros de texto donde aparecen los nombres de cada gráfica
%representada, con la letra del mismo color que la representación.
b = uicontrol('Parent',a, ...
  'Units','normalized', ...
  'FontUnits','normalized', ...
  'BackgroundColor',[0.752941 0.752941 0.752941], ...
  'FontName','Arial', ...
  'FontSize',0.48, ...
  'ForegroundColor',[1 0 0], ...
  'Position',[0.87 0.583146 0.100746 0.0505618], ...
  'String','RL50-130', ...
  'Style','text', ...
  'Tag','StaticText1');

b = uicontrol('Parent',a, ...
  'Units','normalized', ...
  'FontUnits','normalized', ...
  'BackgroundColor',[0.752941 0.752941 0.752941], ...
  'FontName','Arial', ...
  'FontSize',0.48, ...
  'ForegroundColor',[0 0 0], ...
  'Position',[0.87 0.525094 0.100746 0.0505618], ...
  'String','RL50-90', ...
  'Style','text', ...
  'Tag','StaticText1');

b = uicontrol('Parent',a, ...
  'Units','normalized', ...
  'FontUnits','normalized', ...
  'BackgroundColor',[0.752941 0.752941 0.752941], ...
  'FontName','Arial', ...
  'FontSize',0.47, ...
  'ForegroundColor',[0 0 1], ...
  'Position',[0.87 0.467041 0.100746 0.0505618], ...
  'String','RL90-130', ...
  'Style','text', ...
  'Tag','StaticText1');
```

## graf\_bar.m

```
function graf_bar(RL5090,RL90130,RL50130);
%graf_bar(RL5090,RL90130,RL50130);
%
% Pantalla gráfica en la que aparecen representados los promedios de
% las pérdidas por reflexión, para los distintos rangos angulares de
% incidencia. Se representan en forma de gráficos de barras.
%
%Parámetros de entrada:
% RL5090; promedio de la pérdida por reflexión (RL), para las distintas
% bandas de tercio de octava, para los ángulos comprendidos
% entre 50 y 90 grados.
% RL50130; promedio de la pérdida por reflexión (RL), para las distintas
% bandas de tercio de octava, para los ángulos comprendidos
% entre 50 y 130 grados.
% RL90130; promedio de la pérdida por reflexión (RL), para las distintas
% bandas de tercio de octava, para los ángulos comprendidos
% entre 90 y 130 grados.
%
%No tiene parámetros de salida.

%Se cierra la anterior pantalla gráfica.
close(1);

%Se carga la variable para que la pantalla se ejecute adecuadamente.
load graf_bar;

%Parámetros generales de la pantalla gráfica.
a = figure('Units','normalized', ...
    'Color',[0.95 0.95 0.8], ...
    'Colormap',mat0, ...
    'PointerShapeCData',mat1, ...
    'Position',[-0.0025 0.0433333 1.0025 0.898333], ...
    'Tag','Fig1');

%Cuadro de texto que contiene el título de la pantalla gráfica:
%"PROMEDIOS DE LA PÉRDIDA POR REFLEXIÓN."
b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'FontName','Arial', ...
    'FontSize',0.55, ...
    'FontWeight','bold', ...
    'Position',[0.167082 0.914657 0.724439 0.0649351], ...
    'String','PROMEDIOS DE LA PÉRDIDA POR REFLEXIÓN.', ...
    'Style','text', ...
    'Tag','StaticText2');

%Parámetros generales de los ejes de la representación gráfica del
% promedio de la pérdida por reflexión entre 50° y 130°.
b = axes('Parent',a, ...
    'Box','on', ...
    'CameraUpVector',[0 1 0], ...
    'CameraUpVectorMode','manual', ...
    'CLim',[1 2], ...
```



## graf\_bar.m

```
'CLimMode','manual', ...
'Color',[1 1 1], ...
'ColorOrder',mat2, ...
'NextPlot','replacechildren', ...
'Position',[0.198254 0.643785 0.774314 0.22449], ...
'Tag','Axes3', ...
'Visible','off', ...
'XColor',[0 0 0], ...
'YColor',[0 0 0], ...
'ZColor',[0 0 0]);
```

%Se realiza la representación de barras de la pérdida por reflexión entre  
%50° y 130°, para las distintas bandas de tercio de octava.

```
bar(RL50130);
max50130=get(gca,'Ylim');
```

%Parámetros generales de los ejes de la representación gráfica del  
%promedio de la pérdida por reflexión entre 90° y 130°.

```
b = axes('Parent',a, ...
'Box','on', ...
'CameraUpVector',[0 1 0], ...
'CameraUpVectorMode','manual', ...
'CLim',[1 2], ...
'CLimMode','manual', ...
'Color',[1 1 1], ...
'ColorOrder',mat7, ...
'NextPlot','replacechildren', ...
'Position',[0.198254 0.350649 0.774314 0.22449], ...
'Tag','Axes2', ...
'Visible','off', ...
'XColor',[0 0 0], ...
'YColor',[0 0 0], ...
'ZColor',[0 0 0]);
```

%Se realiza la representación de barras de la pérdida por reflexión entre  
%90° y 130°, para las distintas bandas de tercio de octava.

```
bar(RL90130);
max90130=get(gca,'Ylim');
```

%Parámetros generales de los ejes de la representación gráfica del  
%promedio de la pérdida por reflexión entre 50° y 90°.

```
b = axes('Parent',a, ...
'Box','on', ...
'CameraUpVector',[0 1 0], ...
'CameraUpVectorMode','manual', ...
'CLim',[1 2], ...
'CLimMode','manual', ...
'Color',[1 1 1], ...
'ColorOrder',mat12, ...
'NextPlot','replacechildren', ...
'Position',[0.198254 0.0556586 0.774314 0.222635], ...
'Tag','Axes1', ...
'Visible','off', ...
'XColor',[0 0 0], ...
'YColor',[0 0 0], ...
'ZColor',[0 0 0]);
```

## graf\_bar.m

%Se realiza la representación de barras de la pérdida por reflexión entre  
%50° y 90°, para las distintas bandas de tercio de octava.

```
bar(RL5090);  
max5090=get(gca,'Ylim');
```

%Cuadros de texto en los que aparecen las frecuencias correspondientes a  
%cada banda de tercio de octava. Estos son los correspondientes a la  
%pérdida por reflexión entre 50° y 130°.

```
b = uicontrol('Parent',a, ...  
    'Units','normalized', ...  
    'FontUnits','normalized', ...  
    'BackgroundColor',[0.752941 0.752941 0.752941], ...  
    'FontName','Arial narrow', ...  
    'FontSize',0.4, ...  
    'FontWeight','bold', ...  
    'Position',[0.493766 0.602968 0.0286783 0.03], ...  
    'String','500', ...  
    'Style','text', ...  
    'Tag','StaticText3');
```

```
b = uicontrol('Parent',a, ...  
    'Units','normalized', ...  
    'FontUnits','normalized', ...  
    'BackgroundColor',[0.752941 0.752941 0.752941], ...  
    'FontName','Arial narrow', ...  
    'FontSize',0.4, ...  
    'FontWeight','bold', ...  
    'Position',[0.532419 0.602968 0.0286783 0.03], ...  
    'String','630', ...  
    'Style','text', ...  
    'Tag','StaticText3');
```

```
b = uicontrol('Parent',a, ...  
    'Units','normalized', ...  
    'FontUnits','normalized', ...  
    'BackgroundColor',[0.752941 0.752941 0.752941], ...  
    'FontName','Arial narrow', ...  
    'FontSize',0.4, ...  
    'FontWeight','bold', ...  
    'Position',[0.455112 0.602968 0.0286783 0.03], ...  
    'String','400', ...  
    'Style','text', ...  
    'Tag','StaticText3');
```

```
b = uicontrol('Parent',a, ...  
    'Units','normalized', ...  
    'FontUnits','normalized', ...  
    'BackgroundColor',[0.752941 0.752941 0.752941], ...  
    'FontName','Arial narrow', ...  
    'FontSize',0.4, ...  
    'FontWeight','bold', ...  
    'Position',[0.377805 0.602968 0.0286783 0.03], ...  
    'String','250', ...  
    'Style','text', ...  
    'Tag','StaticText3');
```

```
b = uicontrol('Parent',a, ...
```

```

'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'FontName','Arial narrow', ...
'FontSize',0.4, ...
'FontWeight','bold', ...
'Position',[0.416459 0.602968 0.0286783 0.03], ...
'String','315', ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'FontName','Arial narrow', ...
'FontSize',0.4, ...
'FontWeight','bold', ...
'Position',[0.339152 0.602968 0.0286783 0.03], ...
'String','200', ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'FontName','Arial narrow', ...
'FontSize',0.4, ...
'FontWeight','bold', ...
'Position',[0.300499 0.602968 0.0286783 0.03], ...
'String','160', ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'FontName','Arial narrow', ...
'FontSize',0.4, ...
'FontWeight','bold', ...
'Position',[0.261845 0.602968 0.0286783 0.03], ...
'String','125', ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'FontName','Arial narrow', ...
'FontSize',0.4, ...
'FontWeight','bold', ...
'Position',[0.223192 0.602968 0.0286783 0.03], ...
'String','100', ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...

```

```

'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'FontName','Arial narrow', ...
'FontSize',0.4, ...
'FontWeight','bold', ...
'Position',[0.571072 0.602968 0.0286783 0.03], ...
'String','800', ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'FontName','Arial narrow', ...
'FontSize',0.4, ...
'FontWeight','bold', ...
'Position',[0.609726 0.602968 0.0286783 0.03], ...
'String','1000', ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'FontName','Arial narrow', ...
'FontSize',0.4, ...
'FontWeight','bold', ...
'Position',[0.648379 0.602968 0.0286783 0.03], ...
'String','1250', ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'FontName','Arial narrow', ...
'FontSize',0.4, ...
'FontWeight','bold', ...
'Position',[0.687032 0.602968 0.0286783 0.03], ...
'String','1600', ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'FontName','Arial narrow', ...
'FontSize',0.4, ...
'FontWeight','bold', ...
'Position',[0.725686 0.602968 0.0286783 0.03], ...
'String','2000', ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...

```

## graf\_bar.m

```
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'FontName','Arial narrow', ...
'FontSize',0.4, ...
'FontWeight','bold', ...
'Position',[0.764339 0.602968 0.0286783 0.03], ...
'String','2500', ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'FontName','Arial narrow', ...
'FontSize',0.4, ...
'FontWeight','bold', ...
'Position',[0.802993 0.602968 0.0286783 0.03], ...
'String','3150', ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'FontName','Arial narrow', ...
'FontSize',0.4, ...
'FontWeight','bold', ...
'Position',[0.841646 0.602968 0.0286783 0.03], ...
'String','4000', ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'FontName','Arial narrow', ...
'FontSize',0.4, ...
'FontWeight','bold', ...
'Position',[0.879052 0.601428 0.0286783 0.0315399], ...
'String','5000', ...
'Style','text', ...
'Tag','StaticText3');

%Cuadros de texto en los que aparecen las frecuencias correspondientes a
%cada banda de tercio de octava. Estos son los correspondientes a la
%pérdida por reflexión entre 90° y 130°.
b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'FontName','Arial narrow', ...
'FontSize',0.4, ...
'FontWeight','bold', ...
'Position',[0.877805 0.309833 0.0286783 0.0315399], ...
'String','5000', ...
```

## graf\_bar.m

```
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'FontName','Arial narrow', ...
'FontSize',0.4, ...
'FontWeight','bold', ...
'Position',[0.839152 0.309833 0.0286783 0.0315399], ...
'String','4000', ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'FontName','Arial narrow', ...
'FontSize',0.4, ...
'FontWeight','bold', ...
'Position',[0.800499 0.309833 0.0286783 0.0315399], ...
'String','3150', ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'FontName','Arial narrow', ...
'FontSize',0.4, ...
'FontWeight','bold', ...
'Position',[0.763092 0.309833 0.0286783 0.0315399], ...
'String','2500', ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'FontName','Arial narrow', ...
'FontSize',0.4, ...
'FontWeight','bold', ...
'Position',[0.723192 0.309833 0.0286783 0.0315399], ...
'String','2000', ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'FontName','Arial narrow', ...
'FontSize',0.4, ...
'FontWeight','bold', ...
'Position',[0.684539 0.309833 0.0286783 0.0315399], ...
'String','1600', ...
```

```

'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'FontName','Arial narrow', ...
'FontSize',0.4, ...
'FontWeight','bold', ...
'Position',[0.645885 0.309833 0.0286783 0.0315399], ...
'String','1250', ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'FontName','Arial narrow', ...
'FontSize',0.4, ...
'FontWeight','bold', ...
'Position',[0.607232 0.309833 0.0286783 0.0315399], ...
'String','1000', ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'FontName','Arial narrow', ...
'FontSize',0.4, ...
'FontWeight','bold', ...
'Position',[0.568579 0.309833 0.0286783 0.0315399], ...
'String','800', ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'FontName','Arial narrow', ...
'FontSize',0.4, ...
'FontWeight','bold', ...
'Position',[0.220698 0.309833 0.0286783 0.0315399], ...
'String','100', ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'FontName','Arial narrow', ...
'FontSize',0.4, ...
'FontWeight','bold', ...
'Position',[0.259352 0.309833 0.0286783 0.0315399], ...
'String','125', ...

```

```
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'FontName','Arial narrow', ...
'FontSize',0.4, ...
'FontWeight','bold', ...
'Position',[0.299252 0.309833 0.0286783 0.0315399], ...
'String','160', ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'FontName','Arial narrow', ...
'FontSize',0.4, ...
'FontWeight','bold', ...
'Position',[0.336658 0.309833 0.0286783 0.0315399], ...
'String','200', ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'FontName','Arial narrow', ...
'FontSize',0.4, ...
'FontWeight','bold', ...
'Position',[0.413965 0.309833 0.0286783 0.0315399], ...
'String','315', ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'FontName','Arial narrow', ...
'FontSize',0.4, ...
'FontWeight','bold', ...
'Position',[0.375312 0.309833 0.0286783 0.0315399], ...
'String','250', ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'FontName','Arial narrow', ...
'FontSize',0.4, ...
'FontWeight','bold', ...
'Position',[0.453865 0.309833 0.0286783 0.0315399], ...
'String','400', ...
```



## graf\_bar.m

```
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'FontName','Arial narrow', ...
'FontSize',0.4, ...
'FontWeight','bold', ...
'Position',[0.531172 0.309833 0.0286783 0.0315399], ...
'String','630', ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'FontName','Arial narrow', ...
'FontSize',0.4, ...
'FontWeight','bold', ...
'Position',[0.491272 0.309833 0.0286783 0.0315399], ...
'String','500', ...
'Style','text', ...
'Tag','StaticText3');

%Cuadros de texto en los que aparecen las frecuencias correspondientes a
%cada banda de tercio de octava. Estos son los correspondientes a la
%pérdida por reflexión entre 50° y 90°.
b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'FontName','Arial narrow', ...
'FontSize',0.4, ...
'FontWeight','bold', ...
'Position',[0.492519 0.0185529 0.0286783 0.0315399], ...
'String','500', ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'FontName','Arial narrow', ...
'FontSize',0.4, ...
'FontWeight','bold', ...
'Position',[0.532419 0.0185529 0.0286783 0.0315399], ...
'String','630', ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'FontName','Arial narrow', ...
```

## graf\_bar.m

```
'FontSize',0.4, ...
'FontWeight','bold', ...
'Position',[0.455112 0.0185529 0.0286783 0.0315399], ...
'String','400', ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'FontName','Arial narrow', ...
'FontSize',0.4, ...
'FontWeight','bold', ...
'Position',[0.376559 0.0185529 0.0286783 0.0315399], ...
'String','250', ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'FontName','Arial narrow', ...
'FontSize',0.4, ...
'FontWeight','bold', ...
'Position',[0.415212 0.0185529 0.0286783 0.0315399], ...
'String','315', ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'FontName','Arial narrow', ...
'FontSize',0.4, ...
'FontWeight','bold', ...
'Position',[0.337905 0.0185529 0.0286783 0.0315399], ...
'String','200', ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'FontName','Arial narrow', ...
'FontSize',0.4, ...
'FontWeight','bold', ...
'Position',[0.300499 0.0185529 0.0286783 0.0315399], ...
'String','160', ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'FontName','Arial narrow', ...
```

## graf\_bar.m

```
'FontSize',0.4, ...
'FontWeight','bold', ...
'Position',[0.260599 0.0185529 0.0286783 0.0315399], ...
'String','125', ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'FontName','Arial narrow', ...
'FontSize',0.4, ...
'FontWeight','bold', ...
'Position',[0.221945 0.0185529 0.0286783 0.0315399], ...
'String','100', ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'FontName','Arial narrow', ...
'FontSize',0.4, ...
'FontWeight','bold', ...
'Position',[0.569825 0.0185529 0.0286783 0.0315399], ...
'String','800', ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'FontName','Arial narrow', ...
'FontSize',0.4, ...
'FontWeight','bold', ...
'Position',[0.608479 0.0185529 0.0286783 0.0315399], ...
'String','1000', ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'FontName','Arial narrow', ...
'FontSize',0.4, ...
'FontWeight','bold', ...
'Position',[0.647132 0.0185529 0.0286783 0.0315399], ...
'String','1250', ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'FontName','Arial narrow', ...
```

## graf\_bar.m

```
'FontSize',0.4, ...
'FontWeight','bold', ...
'Position',[0.685786 0.0185529 0.0286783 0.0315399], ...
'String','1600', ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'FontName','Arial narrow', ...
'FontSize',0.4, ...
'FontWeight','bold', ...
'Position',[0.724439 0.0185529 0.0286783 0.0315399], ...
'String','2000', ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'FontName','Arial narrow', ...
'FontSize',0.4, ...
'FontWeight','bold', ...
'Position',[0.764339 0.0185529 0.0286783 0.0315399], ...
'String','2500', ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'FontName','Arial narrow', ...
'FontSize',0.4, ...
'FontWeight','bold', ...
'Position',[0.801746 0.0185529 0.0286783 0.0315399], ...
'String','3150', ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'FontName','Arial narrow', ...
'FontSize',0.4, ...
'FontWeight','bold', ...
'Position',[0.840399 0.0185529 0.0286783 0.0315399], ...
'String','4000', ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'FontName','Arial narrow', ...
```

## graf\_bar.m

```
'FontSize',0.4, ...
'FontWeight','bold', ...
'Position',[0.879052 0.0185529 0.0286783 0.0315399], ...
'String','5000', ...
'Style','text', ...
'Tag','StaticText3');
```

%Cuadro de texto en el que aparece lo que se está representando en el  
%gráfico que aparece a su derecha. En este caso, es la pérdida por  
%reflexión entre 50° y 130°, "RL(50°-130°)".

```
b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'FontName','Arial', ...
'FontWeight','Bold',...
'FontSize',0.5, ...
'Position',[0.03 0.688312 0.18 0.07], ...
'String','RL(50°-130°)', ...
'Style','text', ...
'Tag','StaticText1');
```

%Cuadro de texto en el que aparece lo que se está representando en el  
%gráfico que aparece a su derecha. En este caso, es la pérdida por  
%reflexión entre 90° y 130°, "RL(90°-130°)".

```
b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'FontName','Arial', ...
'FontWeight','Bold',...
'FontSize',0.5, ...
'Position',[0.03 0.395176 0.18 0.07], ...
'String','RL(90°-130°)', ...
'Style','text', ...
'Tag','StaticText1');
```

%Cuadro de texto en el que aparece lo que se está representando en el  
%gráfico que aparece a su derecha. En este caso, es la pérdida por  
%reflexión entre 50° y 90°, "RL(50°-90°)".

```
b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[0.752941 0.752941 0.752941], ...
'FontName','Arial', ...
'FontWeight','Bold',...
'FontSize',0.5, ...
'Position',[0.03 0.102041 0.18 0.07], ...
'String','RL(50°-90°)', ...
'Style','text', ...
'Tag','StaticText1');
```

%Cuadro de texto en el que aparecen las unidades representadas en el eje  
%horizontal, "Hz". En este caso, es para la pérdida por reflexión entre  
%50° y 130°.

```
b = uicontrol('Parent',a, ...
```

## graf\_bar.m

```
'Units','normalized', ...
'FontUnits','normalized', ...
'FontName','Arial narrow', ...
'FontSize',0.6, ...
'FontWeight','bold', ...
'Position',[0.912718 0.601428 0.0286783 0.04], ...
'String','Hz', ...
'Style','text', ...
'Tag','StaticText4');

%Cuadro de texto en el que aparecen las unidades representadas en el eje
%horizontal, "Hz". En este caso, es para la pérdida por reflexión entre
%90° y 130°.
b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'FontName','Arial narrow', ...
'FontSize',0.6, ...
'FontWeight','bold', ...
'Position',[0.912718 0.309833 0.0286783 0.038961], ...
'String','Hz', ...
'Style','text', ...
'Tag','StaticText4');

%Cuadro de texto en el que aparecen las unidades representadas en el eje
%horizontal, "Hz". En este caso, es para la pérdida por reflexión entre
%50° y 90°.
b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'FontName','Arial narrow', ...
'FontSize',0.6, ...
'FontWeight','bold', ...
'Position',[0.912718 0.0185529 0.0286783 0.038961], ...
'String','Hz', ...
'Style','text', ...
'Tag','StaticText4');

%Cuadros de texto en los que aparecen los valores de la pérdida por
%reflexión(RL) correspondientes a cada banda de tercio de octava.
%La posición del cuadro de texto es variable con el valor de la RL
%de cada banda. Independientemente del valor de la RL, siempre está
%encima de la banda. En este caso, son para la pérdida por reflexión
%entre 50° y 130°.
b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[1 1 1], ...
'ForegroundColor',[.2 .2 1],...
'FontName','Arial narrow', ...
'FontSize',0.45, ...
'Position',[0.223192 ((RL50130(1)/max50130(2))* .22449 + .646)
0.0286783 0.03], ...
'String',num2str(RL50130(1),2), ...
'Style','text', ...
```

```

    'Tag','StaticText3');

b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'ForegroundColor',[.2 .2 1],...
    'FontName','Arial narrow', ...
    'FontSize',0.45, ...
    'Position',[0.261845 ((RL50130(2)/max50130(2))* .22449 + .646)
0.0286783 0.03], ...
    'String',num2str(RL50130(2),2), ...
    'Style','text', ...
    'Tag','StaticText3');

b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'ForegroundColor',[.2 .2 1],...
    'FontName','Arial narrow', ...
    'FontSize',0.45, ...
    'Position',[0.300499 ((RL50130(3)/max50130(2))* .22449 + .646)
0.0286783 0.03], ...
    'String',num2str(RL50130(3),2), ...
    'Style','text', ...
    'Tag','StaticText3');

b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'ForegroundColor',[.2 .2 1],...
    'FontName','Arial narrow', ...
    'FontSize',0.45, ...
    'Position',[0.339152 ((RL50130(4)/max50130(2))* .22449 + .646)
0.0286783 0.03], ...
    'String',num2str(RL50130(4),2), ...
    'Style','text', ...
    'Tag','StaticText3');

b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'ForegroundColor',[.2 .2 1],...
    'FontName','Arial narrow', ...
    'FontSize',0.45, ...
    'Position',[0.377805 ((RL50130(5)/max50130(2))* .22449 + .646)
0.0286783 0.03], ...
    'String',num2str(RL50130(5),2), ...
    'Style','text', ...
    'Tag','StaticText3');

b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'ForegroundColor',[.2 .2 1],...
    'FontName','Arial narrow', ...

```

## graf\_bar.m

```
'FontSize',0.45, ...
'Position',[0.416459 ((RL50130(6)/max50130(2))*0.22449 + .646)
0.0286783 0.03], ...
'String',num2str(RL50130(6),2), ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[1 1 1], ...
'ForegroundColor',[.2 .2 1],...
'FontName','Arial narrow', ...
'FontSize',0.45, ...
'Position',[0.455112 ((RL50130(7)/max50130(2))*0.22449 + .646)
0.0286783 0.03], ...
'String',num2str(RL50130(7),2), ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[1 1 1], ...
'ForegroundColor',[.2 .2 1],...
'FontName','Arial narrow', ...
'FontSize',0.45, ...
'Position',[0.493766 ((RL50130(8)/max50130(2))*0.22449 + .646)
0.0286783 0.03], ...
'String',num2str(RL50130(8),2), ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[1 1 1], ...
'ForegroundColor',[.2 .2 1],...
'FontName','Arial narrow', ...
'FontSize',0.45, ...
'Position',[0.532419 ((RL50130(9)/max50130(2))*0.22449 + .646)
0.0286783 0.03], ...
'String',num2str(RL50130(9),2), ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[1 1 1], ...
'ForegroundColor',[.2 .2 1],...
'FontName','Arial narrow', ...
'FontSize',0.45, ...
'Position',[0.571072 ((RL50130(10)/max50130(2))*0.22449 + .646)
0.0286783 0.03], ...
'String',num2str(RL50130(10),2), ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
```



## graf\_bar.m

```
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[1 1 1], ...
'ForegroundColor',[.2 .2 1],...
'FontName','Arial narrow', ...
'FontSize',0.45, ...
'Position',[0.609726 ((RL50130(11)/max50130(2))* .22449 + .646)
0.0286783 0.03], ...
'String',num2str(RL50130(11),2), ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[1 1 1], ...
'ForegroundColor',[.2 .2 1],...
'FontName','Arial narrow', ...
'FontSize',0.45, ...
'Position',[0.648379 ((RL50130(12)/max50130(2))* .22449 + .646)
0.0286783 0.03], ...
'String',num2str(RL50130(12),2), ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[1 1 1], ...
'ForegroundColor',[.2 .2 1],...
'FontName','Arial narrow', ...
'FontSize',0.45, ...
'Position',[0.687032 ((RL50130(13)/max50130(2))* .22449 + .646)
0.0286783 0.03], ...
'String',num2str(RL50130(13),2), ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[1 1 1], ...
'ForegroundColor',[.2 .2 1],...
'FontName','Arial narrow', ...
'FontSize',0.45, ...
'Position',[0.725686 ((RL50130(14)/max50130(2))* .22449 + .646)
0.0286783 0.03], ...
'String',num2str(RL50130(14),2), ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[1 1 1], ...
'ForegroundColor',[.2 .2 1],...
'FontName','Arial narrow', ...
'FontSize',0.45, ...
'Position',[0.764339 ((RL50130(15)/max50130(2))* .22449 + .646)
0.0286783 0.03], ...
```

## graf\_bar.m

```
'String',num2str(RL50130(15),2), ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[1 1 1], ...
'ForegroundColor',[.2 .2 1],...
'FontName','Arial narrow', ...
'FontSize',0.45, ...
'Position',[0.802993 ((RL50130(16)/max50130(2))* .22449 + .646)
0.0286783 0.03], ...
'String',num2str(RL50130(16),2), ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[1 1 1], ...
'ForegroundColor',[.2 .2 1],...
'FontName','Arial narrow', ...
'FontSize',0.45, ...
'Position',[0.841646 ((RL50130(17)/max50130(2))* .22449 + .646)
0.0286783 0.03], ...
'String',num2str(RL50130(17),2), ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[1 1 1], ...
'ForegroundColor',[.2 .2 1],...
'FontName','Arial narrow', ...
'FontSize',0.45, ...
'Position',[0.879052 ((RL50130(18)/max50130(2))* .22449 + .646)
0.0286783 0.03], ...
'String',num2str(RL50130(18),2), ...
'Style','text', ...
'Tag','StaticText3');

%Cuadros de texto en los que aparecen los valores de la pérdida por
%reflexión(RL) correspondientes a cada banda de tercio de octava.
%En este caso, son para la pérdida por reflexión entre 90° y 130°.
b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[1 1 1], ...
'ForegroundColor',[.2 .2 1],...
'FontName','Arial narrow', ...
'FontSize',0.45, ...
'Position',[0.223192 ((RL90130(1)/max90130(2))* .22449 + .352)
0.0286783 0.03], ...
'String',num2str(RL90130(1),2), ...
'Style','text', ...
'Tag','StaticText3');
```

```

b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'ForegroundColor',[.2 .2 1],...
    'FontName','Arial narrow', ...
    'FontSize',0.45, ...
    'Position',[0.261845 ((RL90130(2)/max90130(2))* .22449 + .352)
0.0286783 0.03], ...
    'String',num2str(RL90130(2),2), ...
    'Style','text', ...
    'Tag','StaticText3');

b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'ForegroundColor',[.2 .2 1],...
    'FontName','Arial narrow', ...
    'FontSize',0.45, ...
    'Position',[0.300499 ((RL90130(3)/max90130(2))* .22449 + .352)
0.0286783 0.03], ...
    'String',num2str(RL90130(3),2), ...
    'Style','text', ...
    'Tag','StaticText3');

b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'ForegroundColor',[.2 .2 1],...
    'FontName','Arial narrow', ...
    'FontSize',0.45, ...
    'Position',[0.339152 ((RL90130(4)/max90130(2))* .22449 + .352)
0.0286783 0.03], ...
    'String',num2str(RL90130(4),2), ...
    'Style','text', ...
    'Tag','StaticText3');

b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'ForegroundColor',[.2 .2 1],...
    'FontName','Arial narrow', ...
    'FontSize',0.45, ...
    'Position',[0.377805 ((RL90130(5)/max90130(2))* .22449 + .352)
0.0286783 0.03], ...
    'String',num2str(RL90130(5),2), ...
    'Style','text', ...
    'Tag','StaticText3');

b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'ForegroundColor',[.2 .2 1],...
    'FontName','Arial narrow', ...
    'FontSize',0.45, ...

```

## graf\_bar.m

```
'Position',[0.416459 ((RL90130(6)/max90130(2))* .22449 + .352)
0.0286783 0.03], ...
'String',num2str(RL90130(6),2), ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[1 1 1], ...
'ForegroundColor',[.2 .2 1],...
'FontName','Arial narrow', ...
'FontSize',0.45, ...
'Position',[0.455112 ((RL90130(7)/max90130(2))* .22449 + .352)
0.0286783 0.03], ...
'String',num2str(RL90130(7),2), ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[1 1 1], ...
'ForegroundColor',[.2 .2 1],...
'FontName','Arial narrow', ...
'FontSize',0.45, ...
'Position',[0.493766 ((RL90130(8)/max90130(2))* .22449 + .352)
0.0286783 0.03], ...
'String',num2str(RL90130(8),2), ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[1 1 1], ...
'ForegroundColor',[.2 .2 1],...
'FontName','Arial narrow', ...
'FontSize',0.45, ...
'Position',[0.532419 ((RL90130(9)/max90130(2))* .22449 + .352)
0.0286783 0.03], ...
'String',num2str(RL90130(9),2), ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[1 1 1], ...
'ForegroundColor',[.2 .2 1],...
'FontName','Arial narrow', ...
'FontSize',0.45, ...
'Position',[0.571072 ((RL90130(10)/max90130(2))* .22449 + .352)
0.0286783 0.03], ...
'String',num2str(RL90130(10),2), ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
```

```

    'FontUnits','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'ForegroundColor',[.2 .2 1],...
    'FontName','Arial narrow', ...
    'FontSize',0.45, ...
    'Position',[0.609726 ((RL90130(11)/max90130(2))* .22449 + .352)
0.0286783 0.03], ...
    'String',num2str(RL90130(11),2), ...
    'Style','text', ...
    'Tag','StaticText3');

b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'ForegroundColor',[.2 .2 1],...
    'FontName','Arial narrow', ...
    'FontSize',0.45, ...
    'Position',[0.648379 ((RL90130(12)/max90130(2))* .22449 + .352)
0.0286783 0.03], ...
    'String',num2str(RL90130(12),2), ...
    'Style','text', ...
    'Tag','StaticText3');

b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'ForegroundColor',[.2 .2 1],...
    'FontName','Arial narrow', ...
    'FontSize',0.45, ...
    'Position',[0.687032 ((RL90130(13)/max90130(2))* .22449 + .352)
0.0286783 0.03], ...
    'String',num2str(RL90130(13),2), ...
    'Style','text', ...
    'Tag','StaticText3');

b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'ForegroundColor',[.2 .2 1],...
    'FontName','Arial narrow', ...
    'FontSize',0.45, ...
    'Position',[0.725686 ((RL90130(14)/max90130(2))* .22449 + .352)
0.0286783 0.03], ...
    'String',num2str(RL90130(14),2), ...
    'Style','text', ...
    'Tag','StaticText3');

b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'ForegroundColor',[.2 .2 1],...
    'FontName','Arial narrow', ...
    'FontSize',0.45, ...
    'Position',[0.764339 ((RL90130(15)/max90130(2))* .22449 + .352)
0.0286783 0.03], ...
    'String',num2str(RL90130(15),2), ...

```

```

    'Style','text', ...
    'Tag','StaticText3');

b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'ForegroundColor',[.2 .2 1],...
    'FontName','Arial narrow', ...
    'FontSize',0.45, ...
    'Position',[0.802993 ((RL90130(16)/max90130(2))* .22449 + .352)
0.0286783 0.03], ...
    'String',num2str(RL90130(16),2), ...
    'Style','text', ...
    'Tag','StaticText3');

b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'ForegroundColor',[.2 .2 1],...
    'FontName','Arial narrow', ...
    'FontSize',0.45, ...
    'Position',[0.841646 ((RL90130(17)/max90130(2))* .22449 + .352)
0.0286783 0.03], ...
    'String',num2str(RL90130(17),2), ...
    'Style','text', ...
    'Tag','StaticText3');

b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'ForegroundColor',[.2 .2 1],...
    'FontName','Arial narrow', ...
    'FontSize',0.45, ...
    'Position',[0.879052 ((RL90130(18)/max90130(2))* .22449 + .352)
0.0286783 0.03], ...
    'String',num2str(RL90130(18),2), ...
    'Style','text', ...
    'Tag','StaticText3');

%Cuadros de texto en los que aparecen los valores de la pérdida por
%reflexión(RL) correspondientes a cada banda de tercio de octava.
%En este caso, son para la pérdida por reflexión entre 50° y 90°.
b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'ForegroundColor',[.2 .2 1],...
    'FontName','Arial narrow', ...
    'FontSize',0.45, ...
    'Position',[0.223192 ((RL5090(1)/max5090(2))* .22449 + .057) 0.0286783
0.03], ...
    'String',num2str(RL5090(1),2), ...
    'Style','text', ...
    'Tag','StaticText3');

b = uicontrol('Parent',a, ...

```

## graf\_bar.m

```
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[1 1 1], ...
'ForegroundColor',[.2 .2 1],...
'FontName','Arial narrow', ...
'FontSize',0.45, ...
'Position',[0.261845 ((RL5090(2)/max5090(2))*0.22449 + .057) 0.0286783
0.03], ...
'String',num2str(RL5090(2),2), ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[1 1 1], ...
'ForegroundColor',[.2 .2 1],...
'FontName','Arial narrow', ...
'FontSize',0.45, ...
'Position',[0.300499 ((RL5090(3)/max5090(2))*0.22449 + .057) 0.0286783
0.03], ...
'String',num2str(RL5090(3),2), ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[1 1 1], ...
'ForegroundColor',[.2 .2 1],...
'FontName','Arial narrow', ...
'FontSize',0.45, ...
'Position',[0.339152 ((RL5090(4)/max5090(2))*0.22449 + .057) 0.0286783
0.03], ...
'String',num2str(RL5090(4),2), ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[1 1 1], ...
'ForegroundColor',[.2 .2 1],...
'FontName','Arial narrow', ...
'FontSize',0.45, ...
'Position',[0.377805 ((RL5090(5)/max5090(2))*0.22449 + .057) 0.0286783
0.03], ...
'String',num2str(RL5090(5),2), ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[1 1 1], ...
'ForegroundColor',[.2 .2 1],...
'FontName','Arial narrow', ...
'FontSize',0.45, ...
'Position',[0.416459 ((RL5090(6)/max5090(2))*0.22449 + .057) 0.0286783
0.03], ...
```

## graf\_bar.m

```
'String',num2str(RL5090(6),2), ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[1 1 1], ...
'ForegroundColor',[.2 .2 1],...
'FontName','Arial narrow', ...
'FontSize',0.45, ...
'Position',[0.455112 ((RL5090(7)/max5090(2))*0.22449 + .057) 0.0286783
0.03], ...
'String',num2str(RL5090(7),2), ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[1 1 1], ...
'ForegroundColor',[.2 .2 1],...
'FontName','Arial narrow', ...
'FontSize',0.45, ...
'Position',[0.493766 ((RL5090(8)/max5090(2))*0.22449 + .057) 0.0286783
0.03], ...
'String',num2str(RL5090(8),2), ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[1 1 1], ...
'ForegroundColor',[.2 .2 1],...
'FontName','Arial narrow', ...
'FontSize',0.45, ...
'Position',[0.532419 ((RL5090(9)/max5090(2))*0.22449 + .057) 0.0286783
0.03], ...
'String',num2str(RL5090(9),2), ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[1 1 1], ...
'ForegroundColor',[.2 .2 1],...
'FontName','Arial narrow', ...
'FontSize',0.45, ...
'Position',[0.571072 ((RL5090(10)/max5090(2))*0.22449 + .057) 0.0286783
0.03], ...
'String',num2str(RL5090(10),2), ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[1 1 1], ...
```



## graf\_bar.m

```
'ForegroundColor',[.2 .2 1],...
'FontName','Arial narrow', ...
'FontSize',0.45, ...
'Position',[0.609726 ((RL5090(11)/max5090(2))* .22449 + .057) 0.0286783
0.03], ...
'String',num2str(RL5090(11),2), ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[1 1 1], ...
'ForegroundColor',[.2 .2 1],...
'FontName','Arial narrow', ...
'FontSize',0.45, ...
'Position',[0.648379 ((RL5090(12)/max5090(2))* .22449 + .057) 0.0286783
0.03], ...
'String',num2str(RL5090(12),2), ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[1 1 1], ...
'ForegroundColor',[.2 .2 1],...
'FontName','Arial narrow', ...
'FontSize',0.45, ...
'Position',[0.687032 ((RL5090(13)/max5090(2))* .22449 + .057) 0.0286783
0.03], ...
'String',num2str(RL5090(13),2), ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[1 1 1], ...
'ForegroundColor',[.2 .2 1],...
'FontName','Arial narrow', ...
'FontSize',0.45, ...
'Position',[0.725686 ((RL5090(14)/max5090(2))* .22449 + .057) 0.0286783
0.03], ...
'String',num2str(RL5090(14),2), ...
'Style','text', ...
'Tag','StaticText3');

b = uicontrol('Parent',a, ...
'Units','normalized', ...
'FontUnits','normalized', ...
'BackgroundColor',[1 1 1], ...
'ForegroundColor',[.2 .2 1],...
'FontName','Arial narrow', ...
'FontSize',0.45, ...
'Position',[0.764339 ((RL5090(15)/max5090(2))* .22449 + .057) 0.0286783
0.03], ...
'String',num2str(RL5090(15),2), ...
'Style','text', ...
'Tag','StaticText3');
```

## graf\_bar.m

```
b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'ForegroundColor',[.2 .2 1],...
    'FontName','Arial narrow', ...
    'FontSize',0.45, ...
    'Position',[0.802993 ((RL5090(16)/max5090(2))*0.22449 + .057) 0.0286783
0.03], ...
    'String',num2str(RL5090(16),2), ...
    'Style','text', ...
    'Tag','StaticText3');

b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'ForegroundColor',[.2 .2 1],...
    'FontName','Arial narrow', ...
    'FontSize',0.45, ...
    'Position',[0.841646 ((RL5090(17)/max5090(2))*0.22449 + .057) 0.0286783
0.03], ...
    'String',num2str(RL5090(17),2), ...
    'Style','text', ...
    'Tag','StaticText3');

b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'BackgroundColor',[1 1 1], ...
    'ForegroundColor',[.2 .2 1],...
    'FontName','Arial narrow', ...
    'FontSize',0.45, ...
    'Position',[0.879052 ((RL5090(18)/max5090(2))*0.22449 + .057) 0.0286783
0.03], ...
    'String',num2str(RL5090(18),2), ...
    'Style','text', ...
    'Tag','StaticText3');
```

## min\_frec.m

```
function min_frec=min_frec(vent,dt);
%min_frec=min_frec(vent,dt);
% Pantalla gráfica donde se calcula la mínima frecuencia
%por debajo de la cual los resultados obtenidos de la
%pérdida por reflexión (RL) no son fiables. Se calcula
%obteniendo el primer mínimo en la transformada de Fourier
%de la ventana Adrienne.
%
%Parámetros de entrada:
% vent, ventana Adrienne.
% dt, periodo de muestreo, en ms.
%
%Parámetro de salida:
% min_frec, mínima frecuencia fiable, en Hz.

%Se carga la variable asociada a la pantalla gráfica.
load min_frec

%Se definen los parámetros generales de la pantalla.
a = figure('Units','normalized', ...
    'Color',[0.95 0.95 0.8], ...
    'Colormap',mat0, ...
    'PointerShapeCData',mat1, ...
    'Position',[-0.0025 0.05 1.005 0.89], ...
    'Tag','Fig1');

%Título de la pantalla.
b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'BackgroundColor',[0.752941 0.752941 0.752941], ...
    'FontName','Arial', ...
    'FontSize',0.6, ...
    'Position',[0.19 0.88 0.6 0.072], ...
    'String','MÍNIMA FRECUENCIA FIABLE.', ...
    'Style','text', ...
    'Tag','StaticText1');

%Título de la representación de la ventana Adrienne en el
%dominio del tiempo.
b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'FontName','Arial', ...
    'FontSize',0.43, ...
    'Position',[0.0777 0.73 0.3545 0.045], ...
    'String','Ventana en el dominio del tiempo', ...
    'Style','text', ...
    'Tag','StaticText1');

%Ejes para la representación en el dominio del tiempo.
b = axes('Parent',a, ...
    'Units','normalized', ...
    'Position',[0.0722 0.3 0.4055 0.3951]);
```

## min\_frec.m

```
%Se define el vector de tiempos del eje horizontal.
tiempo(length(vent))=0;

for i=1:length(vent)
    tiempo(i)=i*dt;
end;

%Se dibuja la ventana frente al tiempo.
plot(tiempo,vent);

%Se definen los valores numéricos de los extremos de
%los dos ejes.
axis([0 tiempo(length(vent)) min(vent) 1.1]);

%Título del eje horizontal.
xlabel('t(ms)', ...
    'FontName','Arial', ...
    'FontUnits','Normalized', ...
    'FontWeight','Bold',...
    'FontSize',.05);

%Título del eje vertical.
ylabel('Amplitud',...
    'FontName','Arial', ...
    'FontUnits','Normalized', ...
    'FontWeight','Bold', ...
    'FontSize',.05);

%Título de la representación de la ventana Adrienne en el
%dominio de la frecuencia.
b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'FontName','Arial', ...
    'FontSize',0.43, ...
    'Position',[0.5852 0.73 0.3545 0.045], ...
    'String','Ventana en el dominio de la frecuencia', ...
    'Style','text', ...
    'Tag','StaticText1');

%Ejes para la representación en el dominio de la frecuencia.
b = axes('Parent',a, ...
    'Units','normalized', ...
    'Position',[0.58 0.3 0.4055 0.3951]);

%Se hace la transformada de Fourier de la ventana Adrienne.
transf=abs(fft(vent,8192));

%Se calcula el incremento en frecuencia entre dos puntos de
%la transformada.
df=1/(8192*dt*.001);
```

## min\_frec.m

```
%Se define el vector de frecuencia del eje horizontal.
frecuencia(length(vent))==0;

for i=1:length(vent)
    frecuencia(i)=i*df;
end;

%Dibujaremos hasta una frecuencia de 500Hz. Se calcula
%el número de puntos necesario en el vector de frecuencia
%para tener hasta 500Hz.
n=round(500/df);

%Se representa la ventana Adrienne frente a la frecuencia
%hasta 500Hz.
plot(frecuencia(1:n),transf(1:n));

%Se halla el mínimo a la frecuencia más baja. Para ello
%empezamos a buscar a partir del punto de mínima frecuencia
%fiable como si la ventana fuera rectangular. La mínima
%frecuencia fiable siempre será mayor que si la ventana
%fuera rectangular.
i=round(1/(length(vent)*dt*.001)/df);

%Se inicializa la variable inc, con valor negativo.
inc=-1;

%Algoritmo en el que la variable inc nos indica el valor de
%la pendiente. Se detiene cuando la pendiente sea positiva.
%Se empieza a iterar por el punto i.
while inc<0
    inc=transf(i+1)-transf(i);
    i=i+1;
end;

%Asignamos el punto de mínima frecuencia fiable como aquel
%en el que la pendiente cambia de negativa a positiva.
min_frec=frecuencia(i-1);

%Título del eje horizontal.
xlabel('f(Hz)', ...
    'FontName','Arial', ...
    'FontUnits','Normalized', ...
    'FontWeight','Bold',...
    'FontSize',.05);

%Título del eje vertical.
ylabel('Magnitud',...
    'FontName','Arial', ...
    'FontUnits','Normalized', ...
    'FontWeight','Bold', ...
```

## min\_frec.m

```
'FontSize',.05);

%Dos cuadros de texto a través de los cuales se nos informa
%del valor de la mínima frecuencia fiable
b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'FontName','Arial', ...
    'FontSize',0.5, ...
    'Position',[0.2 0.1 0.6 0.06], ...
    'String','La mínima frecuencia fiable es de           Hz.', ...
    'Style','text', ...
    'Tag','StaticText1');

b = uicontrol('Parent',a, ...
    'Units','normalized', ...
    'FontUnits','normalized', ...
    'FontName','Arial', ...
    'FontSize',0.5, ...
    'FontWeight','Bold', ...
    'ForegroundColor',[1 0 0], ...
    'Position',[0.645 0.1 0.07 0.06], ...
    'String',num2str(min_frec,3), ...
    'Style','text', ...
    'Tag','StaticText1');
```

## **1.3.2. PROYECTO DE NORMA EUROPEA PrEN 1793.**

<b>1.3.2.1. PROYECTO DE NORMA EUROPEA PrEN 1793-1.</b>
------------------------------------------------------------



CEN/TC226/WG6/TG1

**NORMA EUROPEA**  
PrEN 1793-1/1995  
(PROYECTO)

**Dispositivos reductores de ruido de tráfico en carreteras**  
**Método de ensayo para determinar el comportamiento acústico**  
Parte 1 : Características intrínsecas - Absorción sonora

**CEN**

Comité Europeo para la Normalización

## Sumario

Prólogo .....	3
0 Introducción .....	3
1 Ambito de aplicación .....	4
2 Referencias normativas .....	4
3 Símbolos .....	4
4 Disposición del ensayo .....	5
5 Procedimiento de ensayo y evaluación .....	7
5.1 Método de ensayo .....	7
5.2 Índice de evaluación de la absorción acústica $DL\alpha$ .....	7
6 Informe de los ensayos .....	7
6.1 Expresión de los resultados .....	7
6.2 Información adicional .....	8
Anejo A (normativo). Clasificación según el índice de evaluación $DL\alpha$ .....	9
Anejo B (informativo). Nota de recomendación relativa al uso del índice de evaluación $DL\alpha$ .....	10

## Prólogo

Este proyecto de norma europea ha sido elaborado por el Comité Técnico CEN/TC 226 "Equipos para carreteras", cuyo Secretariado ostenta AFNOR.

El Comité Técnico CEN/TC 226 ha decidido someter este proyecto de norma europea a la encuesta pública CEN.

Esta norma europea incluye las partes siguientes, bajo el título genérico:

### Dispositivos reductores de ruido de tráfico en carreteras - Método de ensayo para determinar el comportamiento acústico

- Parte 1: Características intrínsecas
  - Absorción sonora.
- Parte 2: Características intrínsecas
  - Aislamiento a ruido aéreo.
- Parte 3: Espectro normalizado de ruido de tráfico.

El Anejo A de la presente parte de esta norma europea tiene carácter normativo y el Anejo B sólo se da a título informativo.

De conformidad con el Reglamento Interior del CEN/CENELEC, los países siguientes tienen la obligación de aplicar esta norma europea:

## 0 Introducción

Cuando haya que instalar una superficie reflectante del sonido a lo largo de una carretera, puede resultar eficaz emplear unos dispositivos de absorción acústica del lado del tráfico, con el objetivo de reducir el ruido ambiental añadido como consecuencia del sonido reflejado. Las circunstancias en las que este tratamiento puede llegar a ser necesario son las siguientes:

- pantallas anti ruido, paredes rocosas o muros de contención susceptibles de reflejar los rayos sonoros hacia zonas no protegidas;
- trincheras verticales o superficies reflectantes en paralelo;
- túneles y sus accesos;
- cuando el tráfico rodado pasa cerca de una pantalla donde existen reflexiones entre los vehículos y la pantalla que pueden reducir la eficacia de ésta.

El objeto de la presente parte de esta norma consiste en especificar un método de ensayo para evaluar las prestaciones de absorción acústica de los dispositivos reductores de ruido diseñados para las carreteras (medición de las prestaciones intrínsecas). Esta norma no se refiere a la determinación de las prestaciones in situ (prestaciones extrínsecas), que dependen además de factores que no están relacionados con el producto en sí mismo, como, por ejemplo, las dimensiones de la pantalla y los factores que se refieren al emplazamiento, tales como la impedancia del terreno, la geometría del emplazamiento, etc. Se ha diseñado este ensayo para establecer la medición de las prestaciones de absorción acústica

intrínsecas del dispositivo, resultando de éstas la clasificación que permita a su vez seleccionar los dispositivos a emplear en aplicaciones específicas a lo largo de las vías.

## 1 Ambito de aplicación

Esta parte de la norma especifica el método de laboratorio que permite medir la absorción acústica de las pantallas anti ruido planas o de los revestimientos para muros de contención o de los túneles. Se debe utilizar para evaluar las prestaciones de absorción acústica intrínsecas de los dispositivos reductores de ruido de tráfico en carreteras que puedan razonablemente ensamblarse dentro de las instalaciones de ensayo descritas en la norma EN 20354.

NOTA: El ensayo ISO según esta norma sólo es estrictamente válido para los absorbentes planos y excluye en especial los dispositivos que actúan como resonadores débilmente amortiguados. Numerosos dispositivos se apartan significativamente de estas exigencias, razón por la que, en este caso, la interpretación de los resultados tendrá que ser realizada con prudencia.

## 2 Referencias normativas

Este documento incluye, mediante referencia fechada o no fechada, algunas disposiciones de otras publicaciones. Estas referencias normativas se citan en los lugares adecuados a lo largo del texto y las publicaciones se enumeran a continuación.

Para las referencias fechadas, las enmiendas o las revisiones ulteriores de cualquiera de estas publicaciones sólo se aplican a este documento si han sido expresamente incorporadas mediante enmienda o revisión. Para las referencias no fechadas, se aplica la última edición de la publicación a la que se hace referencia.

EN 20354: 1993 Acústica - Medición de la absorción acústica en cámara reverberante

pr EN1793-3 Dispositivos reductores de ruido de tráfico en carreteras - Método de ensayo para determinar el comportamiento acústico - Parte 3: Espectro normalizado de ruido de tráfico.

## 3 Símbolos

A efectos de la presente norma, se aplican los símbolos siguientes:

$\alpha_{si}$  Coeficiente de absorción acústica dentro de la iésima banda de tercio de octava.

$L_i$  Nivel de presión sonora normalizado ponderado A (dB) de ruido de tráfico rodado dentro de la iésima banda de tercio de octava.

$DL_{ii}$  Índice de evaluación de la absorción acústica.

#### 4 Disposición de ensayo

La disposición de ensayo se describe en la norma EN 20354, con los añadidos siguientes:

- la muestra a ensayar debe montarse en la cámara de ensayo de la misma manera que el dispositivo propuesto, con los mismos ensamblajes y juntas de estanquidad entre los componentes;
- todas las partes reflectantes del material que estén expuestas del lado del tráfico rodado (postes, patillas de enganche y otros elementos) tienen que estar presentes sobre la muestra que se utiliza para el ensayo;
- cuando se empleen postes en la construcción del dispositivo, se tiene que incluir al menos un poste dentro de la muestra que se utiliza para el ensayo, con paneles fijados a ambos lados. La longitud de los paneles de un lado del poste tiene que ser superior o igual a 2 m (ver figura 1). La cara del panel que se encuentra del lado del tráfico rodado tiene que estar situada haciendo frente a la parte interna de la sala.

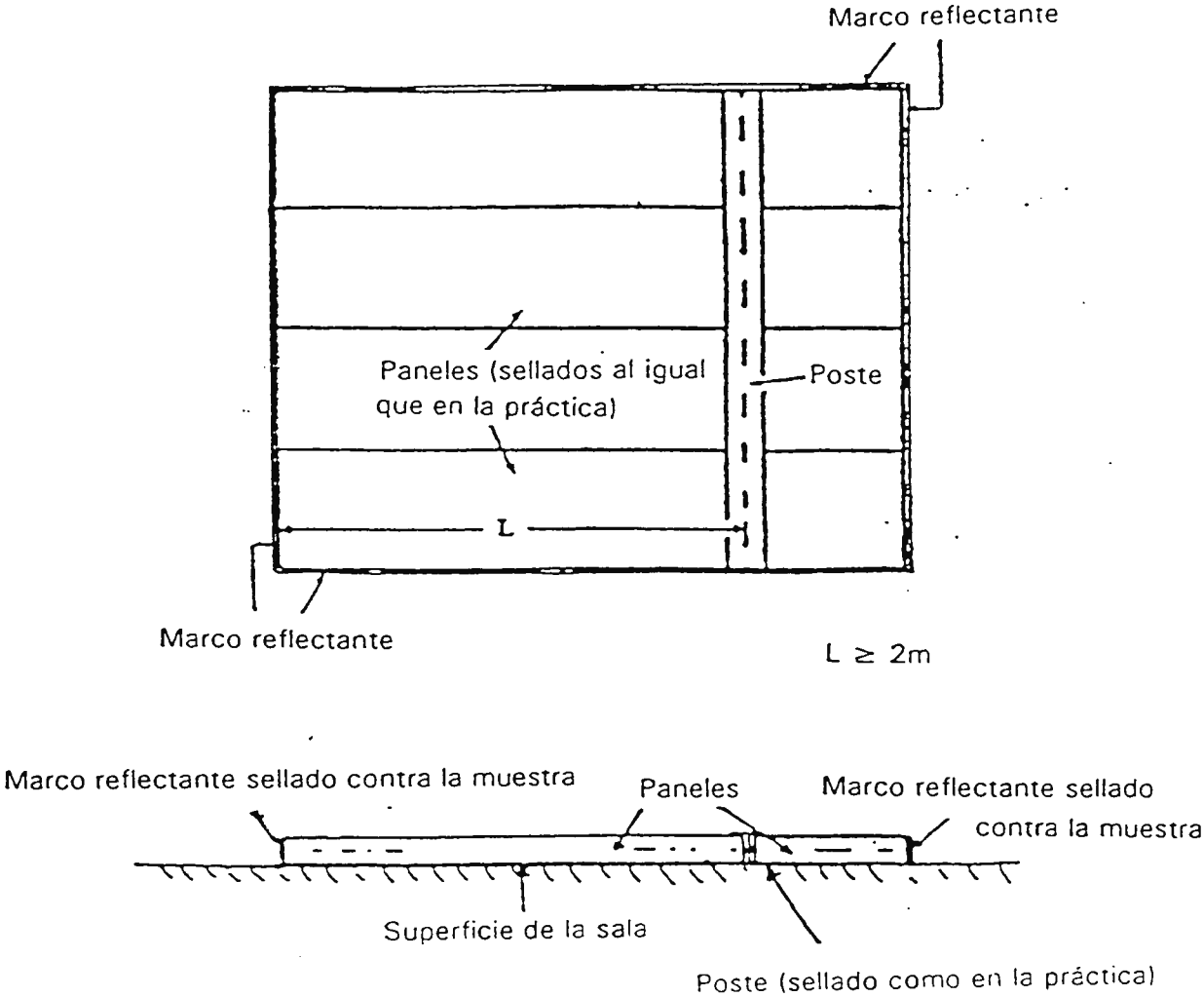


Figura 1: Ilustración de la instalación con poste

- para efectuar el ensayo con pantallas anti ruido, la muestra debe estar situada directamente contra una de las superficies (paredes, techo o suelo) de la sala, sin ningún espacio intermedio, en tanto como sea posible;

- para efectuar el ensayo con un revestimiento absorbente destinado a utilizarse en muros de contención, en túneles, etc., se tiene que montar la muestra a ensayar colocándola contra una de las superficies de la cámara, dejando el mismo espacio que el que se propone para la construcción real. En estas condiciones, la distancia existente entre la parte trasera de la muestra y la superficie de la sala debe quedar claramente consignada.

## 5 Procedimiento de ensayo y evaluación

### 5.1 Método de ensayo

Los coeficientes de absorción acústica  $\alpha_{si}$  dentro de cada banda de tercio de octava en el margen de 100 Hz a 5 kHz deben determinarse utilizando el método descrito en la norma EN 20354.

### 5.2 Índice de evaluación de la absorción acústica $DL_{\alpha}$

Se debe determinar un índice de evaluación para indicar las prestaciones del producto. Los índices de absorción acústica individuales se tienen que ponderar en conformidad con el espectro normalizado de ruido de tráfico rodado definido en la parte 3 de la presente norma.

El índice de evaluación de la absorción acústica  $DL_{\alpha}$  se calcula mediante:

$$DL_{\alpha} = - 10 \lg \left[ 1 - \sum_{i=1}^{18} \alpha_{si} \cdot 10^{0,1Li} \right]$$

NOTA: En casos excepcionales, el valor del término totalizador dentro de la expresión de  $DL_{\alpha}$  puede ser superior a 1, lo que impide el cálculo de  $DL_{\alpha}$ . Por esta razón, el valor máximo del término totalizador se tiene que limitar a 0,99.

Una nota de recomendación sobre la utilización de este índice de evaluación se da en el Anejo B.

## 6 Informe sobre los ensayos

### 6.1 Expresión de los resultados

Los valores por banda de tercio de octava de los coeficientes de absorción acústica  $\alpha_{si}$  para todas las frecuencias de medición se tienen que dar bajo la forma de un cuadro y de un gráfico. Los coeficientes deben redondearse a la segunda cifra decimal.

El índice de evaluación de la absorción acústica  $DL_{\alpha}$  se tiene que consignar tras haber sido redondeado al entero más próximo.

Si es necesario clasificar las prestaciones de absorción, esto debe efectuarse de conformidad con el Anejo A.

Los productos que no se sometan a ensayo se tienen que designar mediante la categoría A0.

## 6.2 Información adicional

El informe sobre el ensayo tiene que incluir:

- a) la referencia de la presente norma;
- b) una descripción de las condiciones del ensayo, procedimiento operativo y equipos utilizados de acuerdo con la norma EN 20354;
- c) la descripción completa de la muestra utilizada, incluyendo el nombre del fabricante y la identificación del producto con los croquis en sección y las condiciones de montaje, incluyendo las masas, las dimensiones, las especificaciones de los paneles, postes, juntas de estanquidad y componentes internos;
- d) el nombre y la dirección del organismo que ha efectuado las mediciones.



Anejo A (normativo)

Clasificación según el índice de evaluación  $DL_{\alpha}$

La exigencia principal de esta parte de la norma consiste en especificar el valor de  $DL_{\alpha}$ . Si además es necesario clasificar las prestaciones de absorción, hay que utilizar entonces las categorías dadas a continuación en el cuadro A.1.

Cuadro A1: Categorías de prestaciones de absorción

Categoría	$DL_{\alpha}$
A0	no ensayado
A1	< 4
A2	4 a 7
A3	8 a 11
A4	> 11

## Anejo B (informativo)

### Nota de recomendación relativa al uso del índice de evaluación $DL_{\alpha}$

El índice de evaluación  $DL_{\alpha}$  es el más directamente adecuado para caracterizar las prestaciones de absorción en situaciones donde la radiación de ruido proveniente del tráfico rodado se refleja sobre la superficie absorbente y se propaga directamente hasta el receptor, sin otras reflexiones adicionales sobre superficies o bien difracciones por encima de pantallas anti ruido o de obstáculos. En el caso de situaciones más complejas en que se producen reflexiones múltiples, o cuando las ondas sonoras se difractan, el espectro original del ruido de tráfico rodado se alterará de forma que los componentes de baja frecuencia pueden resultar reforzados en los puntos receptores. Bajo estas últimas condiciones, es importante considerar las prestaciones de los materiales absorbentes como una función de la frecuencia.

**1.3.2.2. PROYECTO DE NORMA EUROPEA PrEN  
1793-3.**

CEN/TC226/WG6/TG1

## **NORMA EUROPEA**

PrEN 1793-3/1995

(PROYECTO)

Dispositivos reductores de ruido de tráfico en carreteras  
Método de ensayo para determinar el comportamiento acústico  
Parte 3 : Espectro normalizado de ruido de tráfico

**CEN**

Comité Europeo para la Normalización

Sumario

Prólogo .....	3
0    Introducción .....	3
1    Ambito de aplicación .....	3
2    Definiciones .....	3
2.1    Espectro de referencia .....	3
2.2    Nivel por banda de tercio de octava $L_1$ .....	4
3    Espectro de referencia .....	4

## Prólogo

Este proyecto de norma europea ha sido elaborado por el Comité Técnico CEN/TC 226 "Equipos para carreteras", cuyo Secretariado ostenta AFNOR .

El Comité Técnico CEN/TC 226 ha decidido someter este proyecto de norma europea a la encuesta pública CEN.

Esta norma europea incluye las partes siguientes, bajo el título genérico:

### Dispositivos reductores de ruido de tráfico en carreteras - Método de ensayo para determinar el comportamiento acústico

- Parte 1: Características intrínsecas  
- Absorción sonora
- Parte 2: Características intrínsecas  
- Aislamiento a ruido aéreo
- Parte 3: Espectro normalizado de ruido de tráfico.

De conformidad con el Reglamento Interior del CEN/CENELEC, los países siguientes tienen la obligación de aplicar esta norma europea:

## 0 Introducción

Las principales propiedades acústicas de los dispositivos reductores de ruido de tráfico en carreteras, la pérdida por transmisión y la absorción, dependen de la frecuencia. La presente parte de esta norma define las propiedades fundamentales del ruido de tráfico rodado mediante un característico espectro sonoro normalizado del tráfico, que es necesario para determinar unos índices de evaluación.

## 1 Ambito de aplicación

La presente parte de esta norma se tiene que utilizar para evaluar y estimar el comportamiento acústico de los dispositivos anti ruido diseñados para reducir el ruido que genera tráfico en las inmediaciones de las carreteras.

## 2 Definiciones

A efectos de la presente norma, se aplican las definiciones siguientes:

### 2.1 Espectro de referencia

Espectro que se debe utilizar para calcular las prestaciones acústicas de los dispositivos reductores de ruido de tráfico en las inmediaciones de las carreteras, mediante el índice de evaluación de la absorción acústica y el índice de aislamiento a ruido aéreo. Este espectro se expresa en niveles de presión acústica por bandas de tercio de octava dentro del margen de frecuencias de 100 Hz hasta 5 kHz, ponderados A y normalizados a un valor global de 0 dB(A).

## 2.2 Nivel por banda de tercio de octava $L_f$

Nivel de presión sonora en dB(A) del espectro de referencia para la banda de frecuencias con frecuencia central  $f_i$ .

## 3 Espectro de referencia

El espectro normalizado de ruido de tráfico que se tiene que utilizar como espectro de referencia para la evaluación del comportamiento acústico de los dispositivos reductores de ruido de tráfico se da a continuación en el cuadro 1.

Cuadro 1: Espectro de referencia

Frecuencia central $f_i$	Nivel de presión acústica normalizado ponderado A, $L_i$
Hz	dB(A)
100	-20
125	-20
160	-18
200	-16
250	-15
315	-14
400	-13
500	-12
630	-11
800	-9
1000	-8
1250	-9
1600	-10
2000	-11
2500	-13
3150	-15
4000	-16
5000	-18



# PLANOS

Este proyecto, dado su carácter informático, no incorpora planos.

# **PLIEGO DE CONDICIONES**

## **3.0. ÍNDICE.**

**3.0. ÍNDICE.....1**

**3.1 PLIEGO DE CONDICIONES GENERALES Y ECONÓMICAS.....2**

3.1.1. CONDICIONES GENERALES.....2

3.1.2. CONDICIONES ECONÓMICAS.....3

**3.2. PLIEGO DE CONDICIONES TÉCNICAS Y PARTICULARES.....4**

3.2.1. MANUAL DE USUARIO.....6

## **3.1 PLIEGO DE CONDICIONES GENERALES Y ECONÓMICAS.**

### **3.1.1. CONDICIONES GENERALES.**

Las condiciones y cláusulas que se establecen en este documento tratan sobre la contratación por parte de persona física o jurídica del procedimiento de cálculo, implementado en un programa informático, desarrollado en este proyecto.

El cumplimiento de estas condiciones obliga a ambas partes y son las que, a continuación, se exponen:

1. Las dos partes se comprometen desde la fecha de la firma del contrato a cumplir todo lo que se estipula a continuación.
2. El proyectista queda obligado a facilitar a la otra parte cualquier información que contribuya a mejorar la instalación y funcionamiento del sistema, siempre que se le requiera para ello.
3. El proyectista no se responsabiliza del incorrecto funcionamiento del programa si se demuestra que éste ha sido objeto de manipulación indebida.



### **3.1.2. CONDICIONES ECONÓMICAS.**

1. Ambas partes se comprometen de mutuo acuerdo a respetar plazos y cuantías o, en su defecto, satisfacer las penalidades correspondientes expresadas en este pliego de condiciones.
2. El tiempo de realización de este proyecto se fija en nueve meses.
3. El valor total de la ejecución del proyecto será el que figure en el Documento nº1: Memoria, en la última hoja de su apartado 1.1. Memoria descriptiva, detallándose las diferentes partidas en el Documento nº4: Presupuesto.
4. Los pagos se satisfarán de la siguiente forma:
  - 20% a la firma del contrato.
  - 50% en el momento de la entrega.
  - 20% a los 60 días.
5. La empresa contratante se compromete a retribuir las penalidades o aumentos siguientes en caso de incumplimiento de las condiciones fijadas:
  - Un incremento del 6% anual en concepto de intereses por retraso en los pagos.
  - Un incremento del 5% anual por penalidad por retraso los pagos.
6. La empresa contratada asumirá una reducción en sus pagos del 5% anual por incumplimiento del plazo establecido para la realización del proyecto.
7. Los demás conflictos que surgieren serán dirimidos de acuerdo con las leyes vigentes en el Estado Español.

## **3.2. PLIEGO DE CONDICIONES TÉCNICAS Y PARTICULARES.**

1. El sistema desarrollado en este proyecto de fin de carrera debe ser capaz de obtener:

- La pérdida acústica por reflexión (RL) de una determinada barrera antirruido expresada en bandas de tercio de octava para cada promedio angular (50°-90°, 90°-130°, 50°-130°).
- Distancia óptima entre altavoz y micrófono para que la frecuencia fiable sea lo más baja posible.
- La mínima frecuencia por debajo de la cual los resultados obtenidos para la pérdida por reflexión no son fiables.

a partir de los siguientes datos introducidos por el usuario:

- Número de puntos característicos de la barrera.
- Periodo de muestreo de las señales de presión acústica.
- Parámetros geométricos asociados a la medida: altura de la barrera, distancia entre micrófono y barrera, distancia entre altavoz y micrófono.
- Señales medidas de presión acústica para el eco directo.
- Señales medidas de presión acústica para el eco directo + reflejado.

2. El manejo del programa informático desarrollado se ajustará a lo indicado en el apartado "3.2.1. MANUAL DE USUARIO".

3. Si el usuario quisiera obtener el parámetro  $DL_{\alpha}$  a partir de los datos de pérdida acústica por reflexión calculados por el sistema desarrollado en este proyecto, se debería atener a lo indicado en las futuras normas europeas EN 1793-1 y EN 1793-3. A fecha de entrega de este proyecto de fin de carrera no existe una redacción definitiva de dichas normas. En el

apartado 1.3.2. de este proyecto de fin de carrera aparecen como anejos los proyectos de normas **PrEN 1793-1** y **PrEN 1793-3**.

4. Los equipos informáticos han de estar homologados conforme a la reglamentación vigente en España en el día de la fecha del presente proyecto.
5. El idioma de representación será el castellano.
6. El lugar de instalación del equipo de ajustarse a los niveles de temperatura y humedad indicados para el equipo informático.
7. La alimentación eléctrica del sistema debe ser de 220V y 50Hz. Deberá ajustarse a todo lo estipulado en el reglamento de baja tensión.
8. La versión de Matlab<sup>®</sup> usada será a las 5.1 o superior.
9. El sistema operativo usado será el MS-DOS (versión 6 o superior).

### **3.2.1. MANUAL DE USUARIO.**

### **3.2.1.0. INTRODUCCIÓN.**

El programa ha sido diseñado de modo que la comunicación entre el usuario y el sistema informático sea cómoda e intuitiva.

Para usar adecuadamente el programa se han de seguir los siguientes **pasos**:

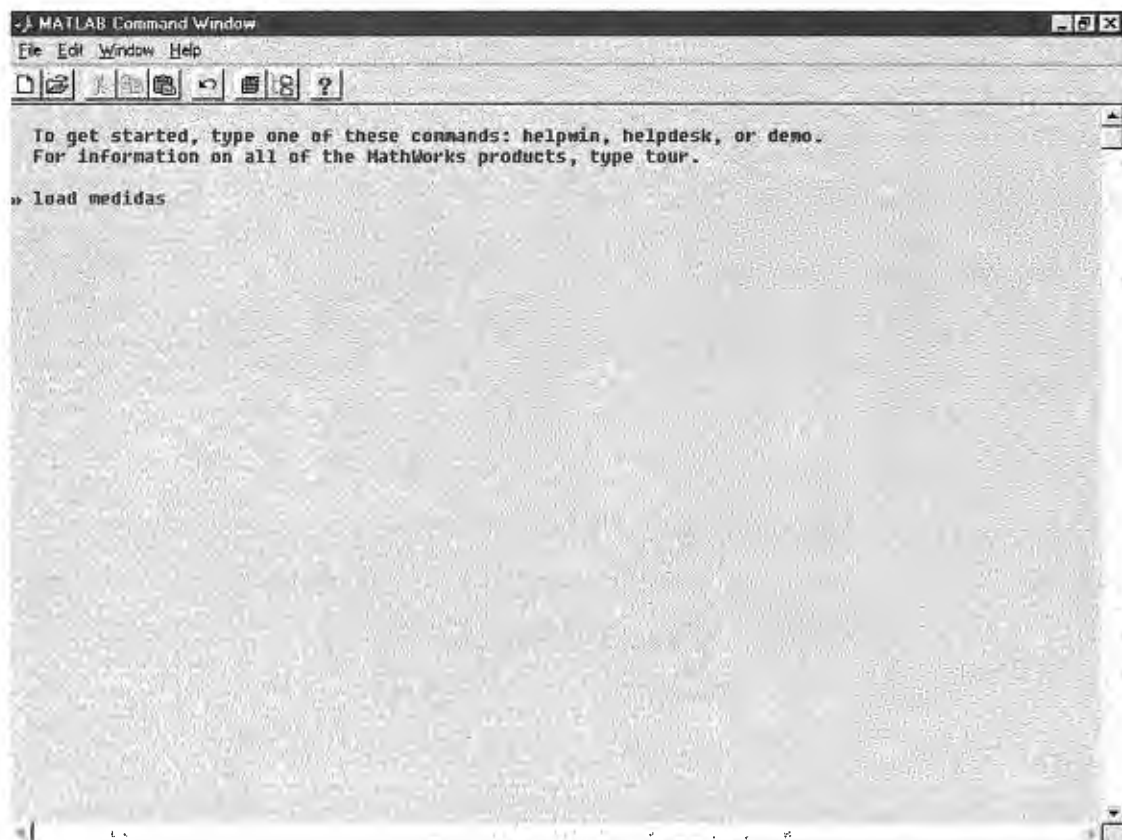
### 3.2.1.1. CARGA DE LAS VARIABLES QUE CONTIENEN LAS SEÑALES MEDIDAS.

Partimos de que las señales tomadas en el ensayo ya están en el formato adecuado para trabajar directamente en Matlab.

Estas señales se encontrarán en una variable de datos (con extensión **.mat**). Supongamos que la señal directa y las señales reflejadas se hallan en la variable **medidas.mat**.

Para cargarlas escribiríamos en la Matlab Command Window:

```
>> load medidas
```

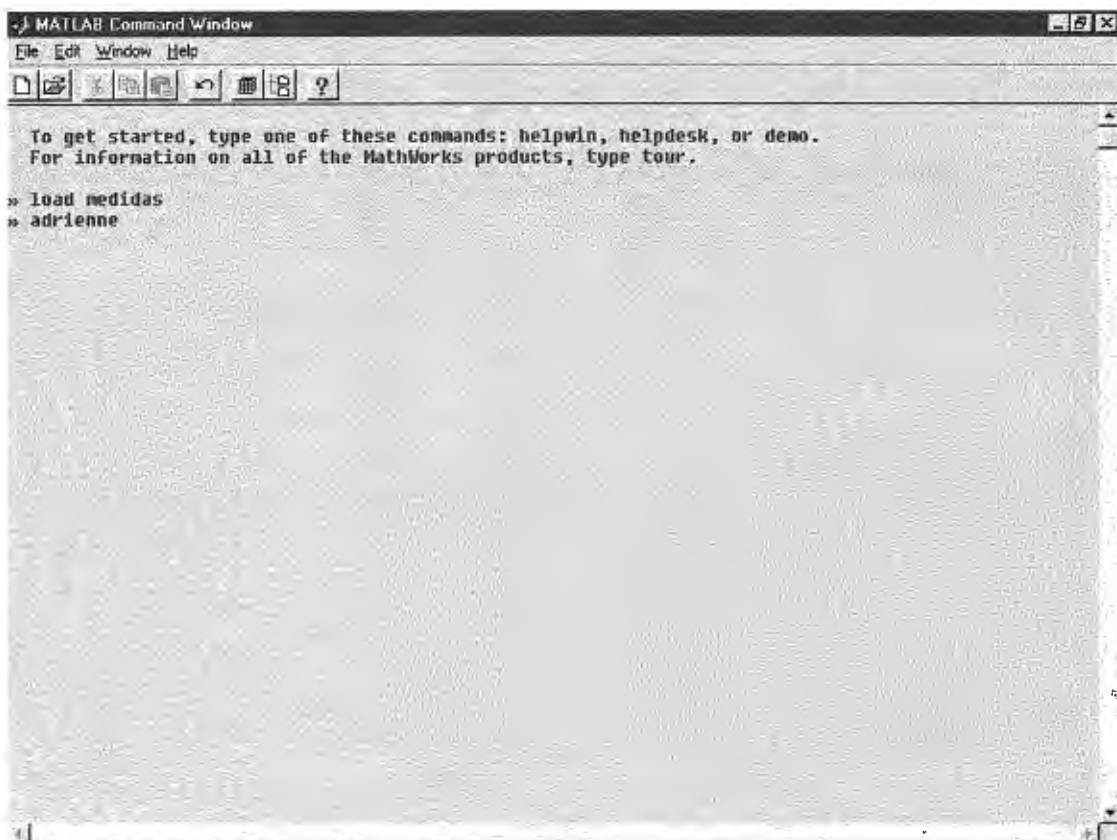




### 3.2.1.2. EJECUCIÓN DEL PROGRAMA.

Para ejecutar el programa basta con escribir en la Matlab Command Window la palabra "Adrienne".

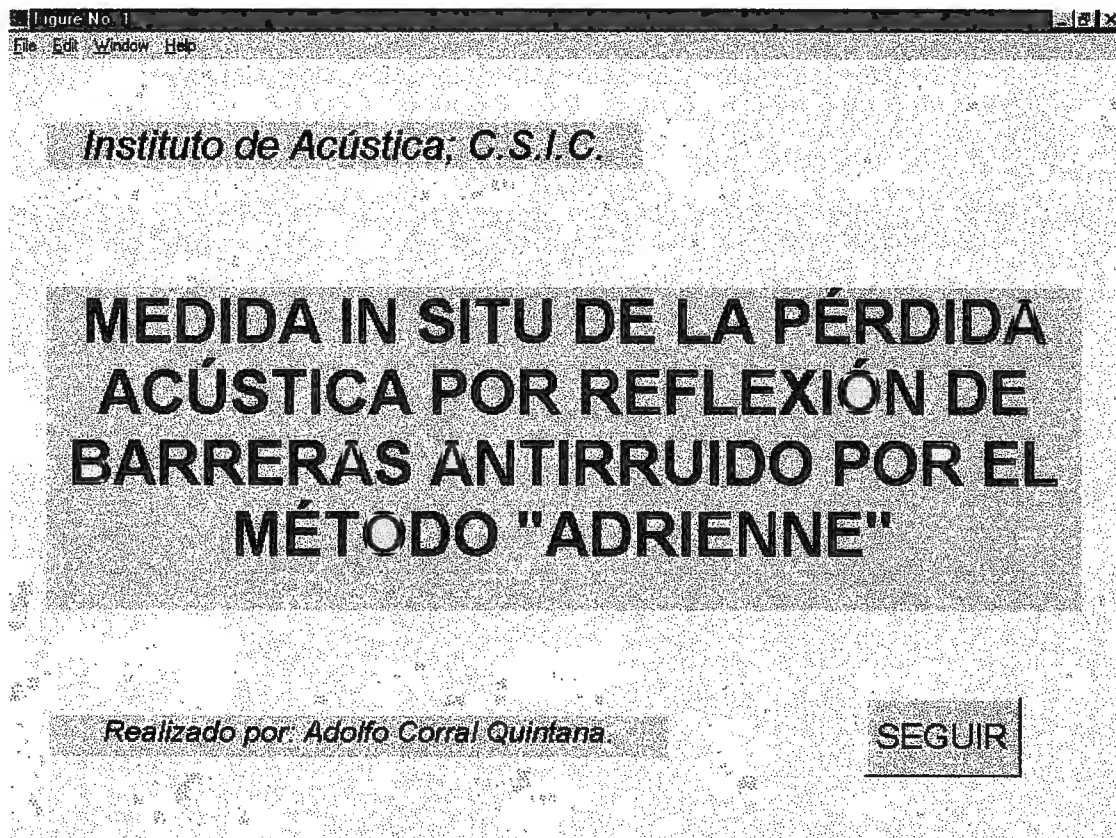
```
>> adrienne
```



Inmediatamente después aparece la primera pantalla gráfica, la portada de presentación. A partir de aquí toda la comunicación entre el usuario y el sistema informático se hará a través de **Interfaces Gráficas de Usuario (G.U.I.)**.

### 3.2.1.3. TÍTULO DEL PROYECTO.

Aparece una pantalla gráfica que contiene el título del proyecto de fin de carrera, la persona que lo ha realizado y la institución donde se ha llevado a cabo el proyecto. Viene acompañada con una música que dura, aproximadamente, 5 segundos. Transcurrido ese tiempo aparece un botón en el que pone "SEGUIR". Lo pulsamos con el ratón y pasamos a la siguiente pantalla gráfica.



### 3.2.1.4. INTRODUCCIÓN DE DATOS GENERALES.

En esta pantalla el usuario ha de introducir varios datos. Se debe pinchar con el ratón en cada uno de los campos e introducir el dato correspondiente. Se debe alimentar al programa con los siguientes parámetros (entre paréntesis aparecen los datos que se introducen en el ejemplo):

- nombre de la variable que contiene la señal directa. **(directa)**
- número de puntos de medida en la barrera. **(3)**
- periodo de muestreo, en milisegundos. **(0.01325)**
- altura de la barrera, en metros. **(4)**
- distancia entre el micrófono y la barrera, en metros. **(0.25)**

Figure No. 1

File Edit Window Help

Pérdida por reflexión (RL)

Introduce la variable que contiene la señal directa: directa

Introduce el número de puntos de medida: 3

Introduce el periodo de muestreo, en ms. 0.01325

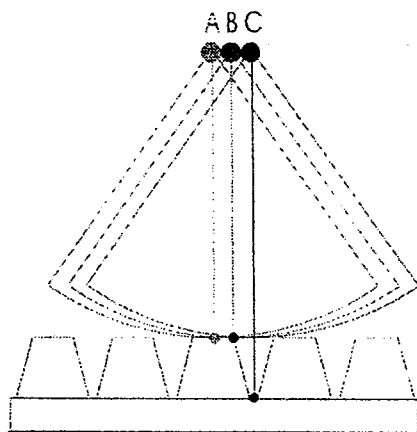
Introduce la altura de la barrera, en m. 4

Introduce, en m, la distancia entre el micrófono y la barrera 0.25

SEGUIR

La variable que contiene la señal **directa** debe contener el registro del micrófono cuando el equipo se orienta al cielo, de modo que simplemente se mide el sonido directo, al tener, prácticamente, condiciones de campo libre.

El **número de puntos** de medida nos da el número de puntos necesario para caracterizar acústicamente la barrera. Para barreras planas es suficiente con un solo punto. Para barreras con un perfil determinado se necesita más de un punto. Para este ejemplo se toma una barrera de un material llamado *Beton Bois*, que se caracteriza por tener el siguiente perfil, con tres puntos característicos.



**Figura 3.2.1.4.1. Los tres puntos característicos en el material Beton Bois.**

El **periodo de muestreo** es el tiempo que transcurre entre dos puntos medidos de la señal. Se tienen señales muestreadas, en forma discreta.

La distancia entre el micrófono y la barrera es la distancia entre estos dos elementos cuando el equipo de medida está orientado **perpendicularmente** a la barrera.

Una vez introducidos todos estos datos se pulsa con el ratón en el botón **"SEGUIR"**.

### 3.2.1.5. INTRODUCCIÓN DE LA DISTANCIA ENTRE EL ALTAVOZ Y EL MICRÓFONO.

La pantalla es la misma que en el caso anterior, con la diferencia de que hay un nuevo campo, la distancia entre el altavoz y el micrófono.

En este nuevo campo se informa al usuario del valor de esta distancia para obtener una frecuencia mínima fiable lo más baja posible (P. Cobo, 1999). El usuario debe introducir la distancia con la que realmente se ha llevado a cabo el ensayo. Para este ejemplo se tomó una distancia entre altavoz y micrófono de 1.25 m.

Figure No. 1

File Edit Window Help

Pérdida por reflexión (RL)

Introduce la variable que contiene la señal directa:	directa
Introduce el número de puntos de medida:	3
Introduce el periodo de muestreo, en ms.	0.01325
Introduce la altura de la barrera, en m.	4
Introduce, en m, la distancia entre el micrófono y la barrera:	0.25

Introduce, en m, la distancia entre el altavoz y el micrófono. La distancia óptima calculada es de 0.984 m

1.25

SEGUIR

Una vez introducida esta distancia se ha de pulsar con el ratón en el botón "SEGUIR".

### 3.2.1.6. INTRODUCCIÓN DE LAS SEÑALES MEDIDAS EN EL ENSAYO.

Se trata ahora de introducir los vectores que contienen las señales medidas de presión acústica para las ondas reflejadas. Estos vectores se deben cargar previamente cuando el usuario esté en la Command Window de Matlab. Este proceso se describió en el apartado "3.2.1.1. CARGA DE LAS VARIABLES QUE CONTIENEN LAS SEÑALES MEDIDAS".

Supongamos que la matriz que contiene estas señales se llama **bb**. Las señales aparecen en esta matriz por columnas, es decir, cada columna representa una señal reflejada para un punto determinado de la barrera y con un ángulo de incidencia determinado.

Figure No. 1

File Edit Window Help

Pérdida por reflexión (RL)

Introduce las variables que contienen la señal medida para los distintos ángulos de incidencia, en el punto 2

50°	bb(:,2)
60°	bb(:,5)
70°	bb(:,8)
80°	bb(:,11)
90°	bb(:,14)
100°	bb(:,17)
110°	bb(:,20)
120°	bb(:,23)
130°	bb(:,26)

COMPROBAR

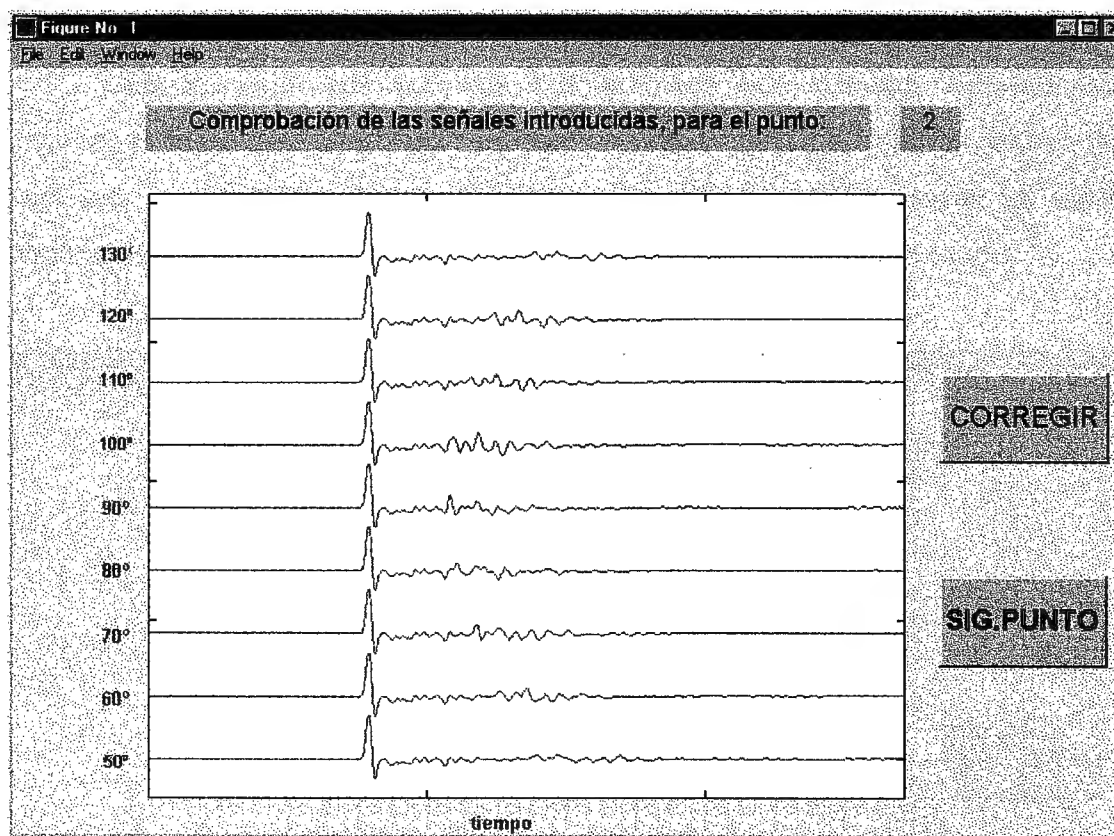
En la anterior pantalla se puede apreciar cómo se deben introducir las señales medidas. En este caso, se considera que las señales medidas corresponden al punto 2 de estudio. Las señales medidas se hallan en las columnas 2, 5, 8, 11, 14, 17, 20, 23 y 26 (que, en "formato Matlab", se han de introducir como `bb(:,2)`, `bb(:,5)`, ...); corresponde a las señales medidas para los nueve ángulos de incidencia: 40°, 50°, 60°, 70°, 80°, 90°, 100°, 110°, 120° y 130°.

Como este proceso de introducción de las distintas variables puede resultar algo tedioso podría dar lugar a introducir algún error. Esto se evita con la siguiente pantalla gráfica donde se puede comprobar que los datos introducidos son los adecuados.

Para ver la representación gráfica de las señales introducidas, debemos pulsar el botón "**COMPROBAR**".

### 3.2.1.7. REPRESENTACIÓN DE LAS SEÑALES INTRODUCIDAS.

En esta pantalla gráfica aparecen representadas las señales de presión acústica para todos los ángulos de incidencia en un punto determinado. Aparecen todas las señales en un mismo gráfico y a la izquierda de cada una de ellas al ángulo de incidencia al que corresponden. En la parte superior de la pantalla figura el punto del cual se están representando las señales.



De esta manera, se puede saber si hemos introducido los datos correctamente. El eco en incidencia normal es el primero en ser recogido por el micrófono y por tanto, la alteración, en este caso, es la primera en desaparecer. Esta figura recibe el nombre de ecograma.

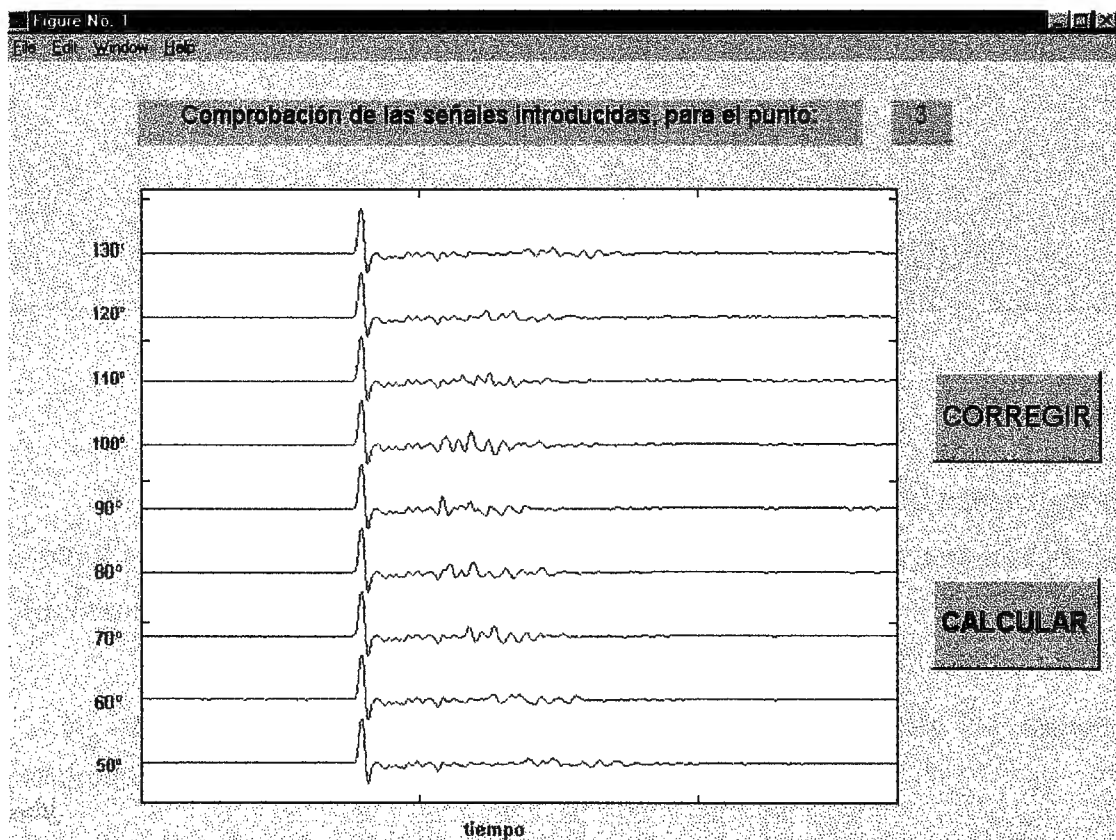


Si el ecograma presenta un aspecto similar al de la figura anterior entonces las señales han sido correctamente introducidas. El eco para incidencia normal es el primero en llegar y también el primero en desaparecer. Según nos alejamos de la incidencia normal los ecos tardan más en llegar y también más en desaparecer.

Si el ecograma es correcto el usuario debe pulsar el botón **"SIG.PUNTO"** y se introducirían las señales para el punto 3.

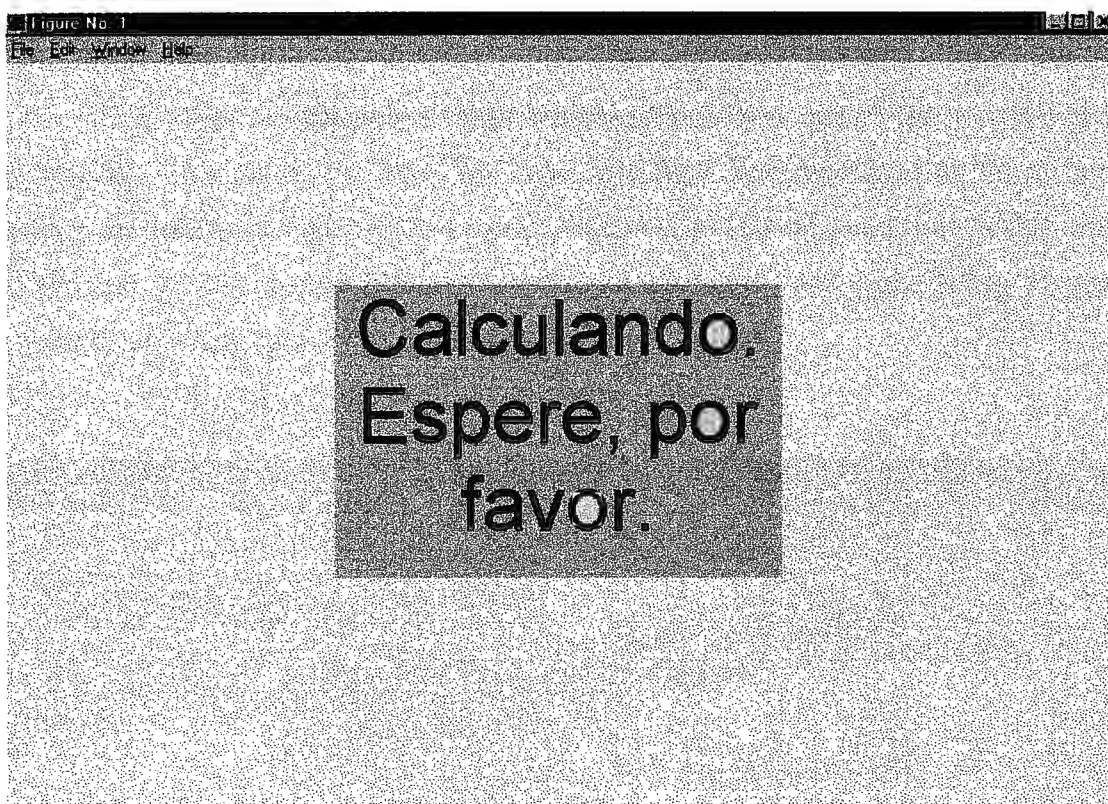
Si la forma del ecograma hace sospechar que los datos son erróneos, entonces el usuario debe pulsar el botón **"CORREGIR"**, que nos devuelve a la pantalla anterior y se debería volver a introducir los datos.

Cuando nos encontramos ya en el último punto (en el ejemplo, el 3) en vez de aparecer el botón **"SIG.PUNTO"** aparece un botón **"CALCULAR"** y, cuando lo pulsamos, el programa comienza a realizar los cálculos, ya que todos los datos necesarios han sido introducidos. Esta última pantalla aparece a continuación.



### 3.2.1.8. PANTALLA DE ESPERA.

Esta pantalla aparece mientras se están realizando los cálculos. En ella aparece el mensaje "**Calculando. Espere por favor**".



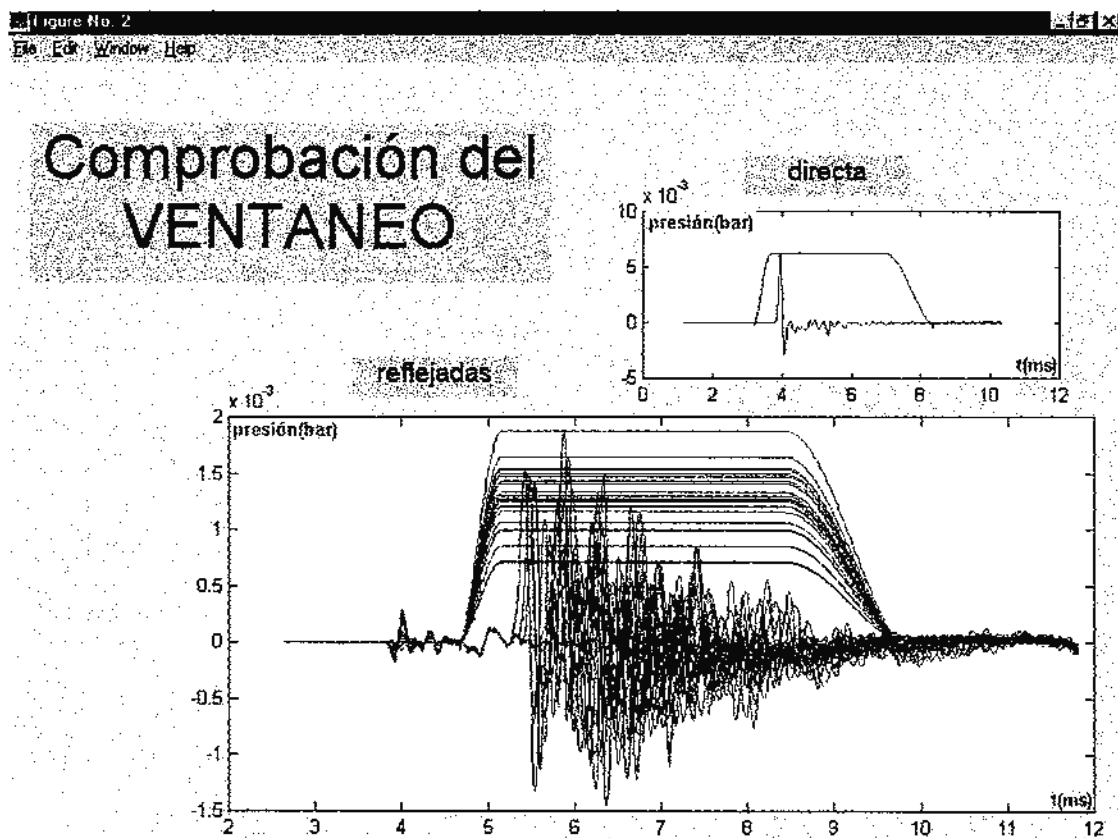
El tiempo de ejecución de los cálculos depende del equipo informático en el que esté corriendo el programa pero no será superior a los 5 minutos para equipos ya algo obsoletos en la fecha de realización del proyecto.

Tras realizarse los cálculos aparecen 4 pantallas gráficas con los distintos **resultados** obtenidos. Estas pantallas se describen a continuación.

### 3.2.1.9. COMPROBACIÓN DEL VENTANEO.

En esta pantalla, se representan tanto la señal directa como las reflejadas con sus correspondientes ventanas. La técnica de ventaneo forma parte de la "Técnica de substracción", usada para obtener, únicamente, los ecos reflejados a partir de una señal que también contiene el eco directo. La **ventana Adrienne** está descrita en el apartado "1.1.3.2.2. VENTANA ADRIENNE."

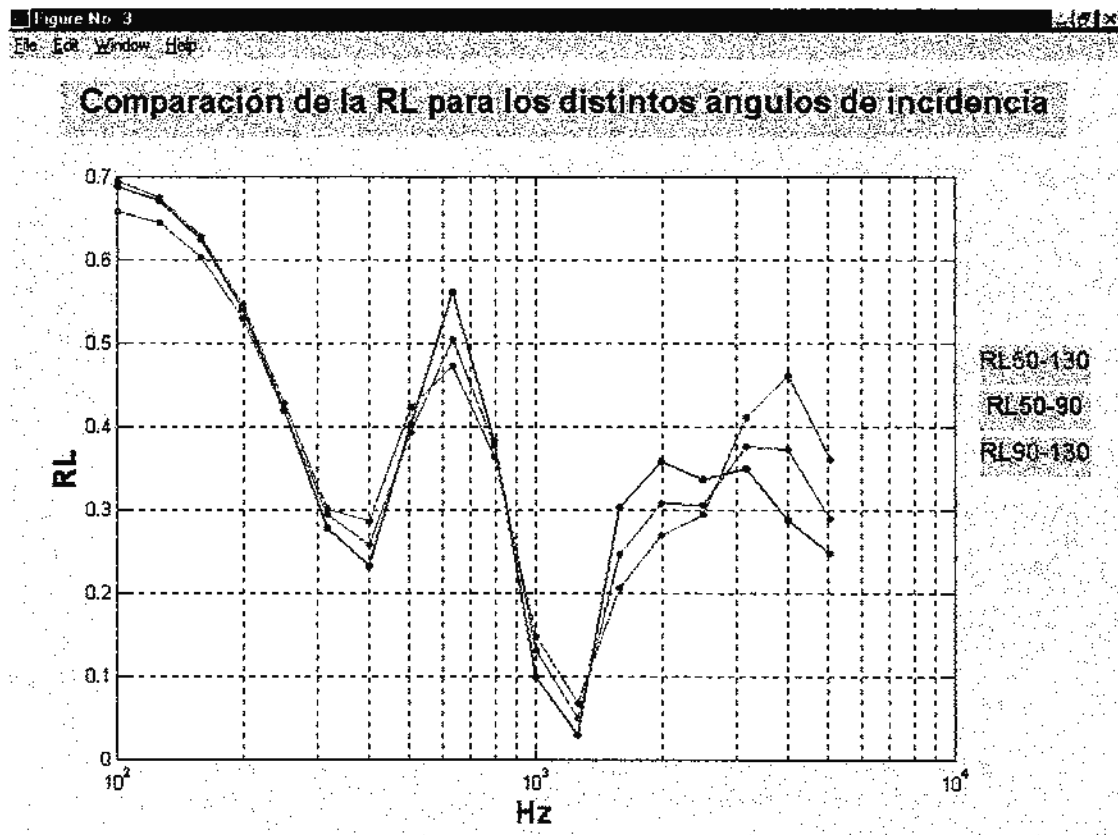
El proceso de ventaneo es especialmente delicado ya que es fundamental que la ventana Adrienne recoja todos los ecos reflejados. Representando en un mismo gráfico las señales reflejadas y la ventana Adrienne, el usuario puede comprobar que el proceso de ventaneo ha sido correctamente realizado (o no) y que si hay alguna anomalía no puede (puede) ser debida al ventaneo.



### 3.2.1.10. COMPARACIÓN DE LA PÉRDIDA POR REFLEXIÓN (RL) PARA LOS DISTINTOS ÁNGULOS DE INCIDENCIA.

En esta pantalla gráfica se pueden apreciar los tres promedios de la pérdida por reflexión para los siguientes rangos angulares:  $50^{\circ}$ - $90^{\circ}$ ,  $90^{\circ}$ - $130^{\circ}$  y  $50^{\circ}$ - $130^{\circ}$ . Como se puede observar en la figura estos promedios son función de la frecuencia. En el eje horizontal está representada la frecuencia en hercios, en escala logarítmica.

Esta gráfica sólo tiene valor para establecer una **comparación cualitativa** entre los valores de la pérdida por reflexión para los distintos ángulos de incidencia.



### **3.2.1.11. VALORES DE LA PÉRDIDA POR REFLEXIÓN (RL) PARA LOS DISTINTOS PROMEDIOS ANGULARES CON RESULTADOS NUMÉRICOS EN BANDAS DE TERCIO DE OCTAVA.**

Aparecen en esta pantalla los valores de la pérdida por reflexión (RL) en cada banda de tercio de octava para cada uno de los promedios angulares.

Estos valores son los que serán posteriormente utilizados para el cálculo del parámetro que determina las propiedades absorbentes de la barrera antirruído, el  $DL\alpha$ . Este parámetro se ha de obtener según los pasos indicados en la norma europea EN 1793, partes 1 y 3. Estas dos partes de la norma se pueden consultar en el apartado "1.3.2. PROYECTO DE NORMA EUROPEA PrEN 1793", perteneciente al documento "ANEJOS".

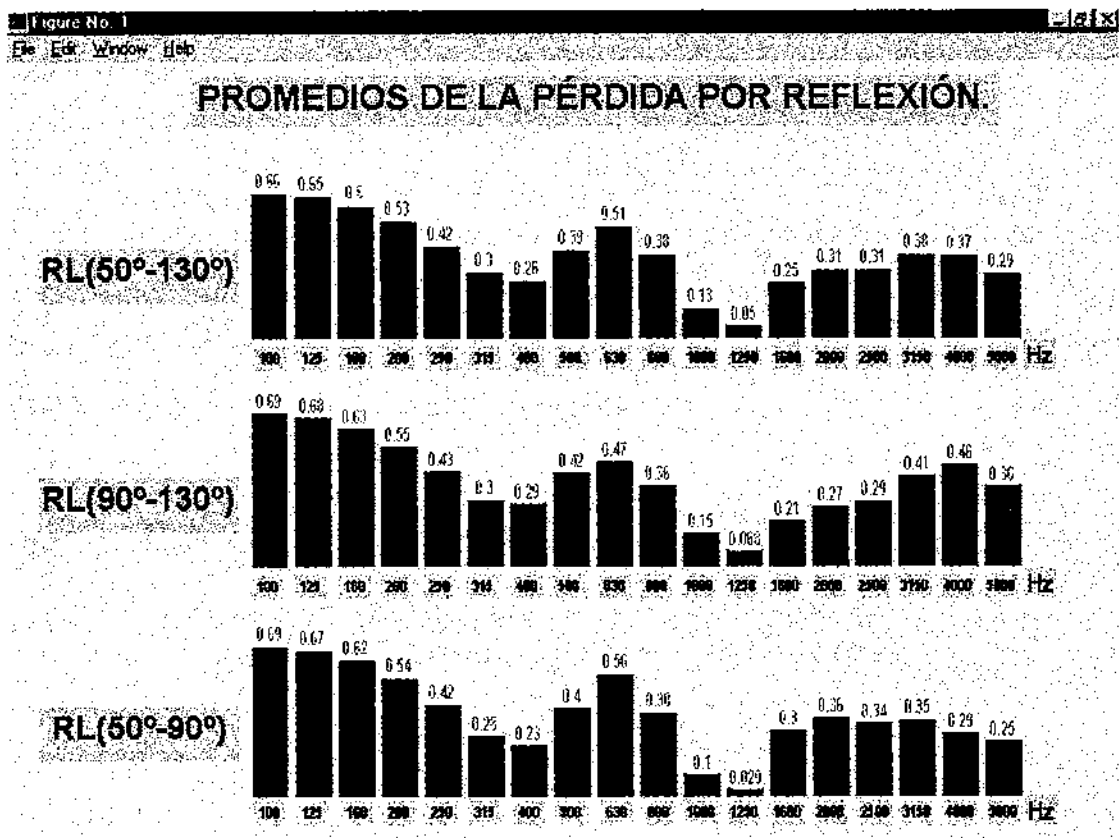
Esta pantalla es la que realmente es el resultado final del programa ya que el objetivo del proyecto es determinar los valores de la pérdida por reflexión (RL) que posteriormente serán utilizados para determinar las propiedades absorbentes de la barrera antirruído.

En ella aparecen representadas las pérdidas por reflexión para los tres rangos angulares ( $50^\circ$ - $130^\circ$ ,  $90^\circ$ - $130^\circ$  y  $50^\circ$ - $90^\circ$ ) en gráficos de barras. Cada barra representa la pérdida por reflexión en cada tercio de octava. Encima de cada barra aparece el valor de la pérdida por reflexión (RL) para el valor de la frecuencia indicado más abajo.

Al contrario que en la pantalla anterior, aquí las RL para los distintos promedios angulares aparecen separadas, ya que interesa conocer el valor de la misma para cada tercio de octava de cada promedio angular.

Por tanto, en esta pantalla, se puede obtener una **medida cuantitativa** de la pérdida por reflexión (**RL**) para una frecuencia y rango angular determinados.

Esta pantalla, para los datos del ejemplo particular que se ha ido desarrollando, aparece aquí abajo.

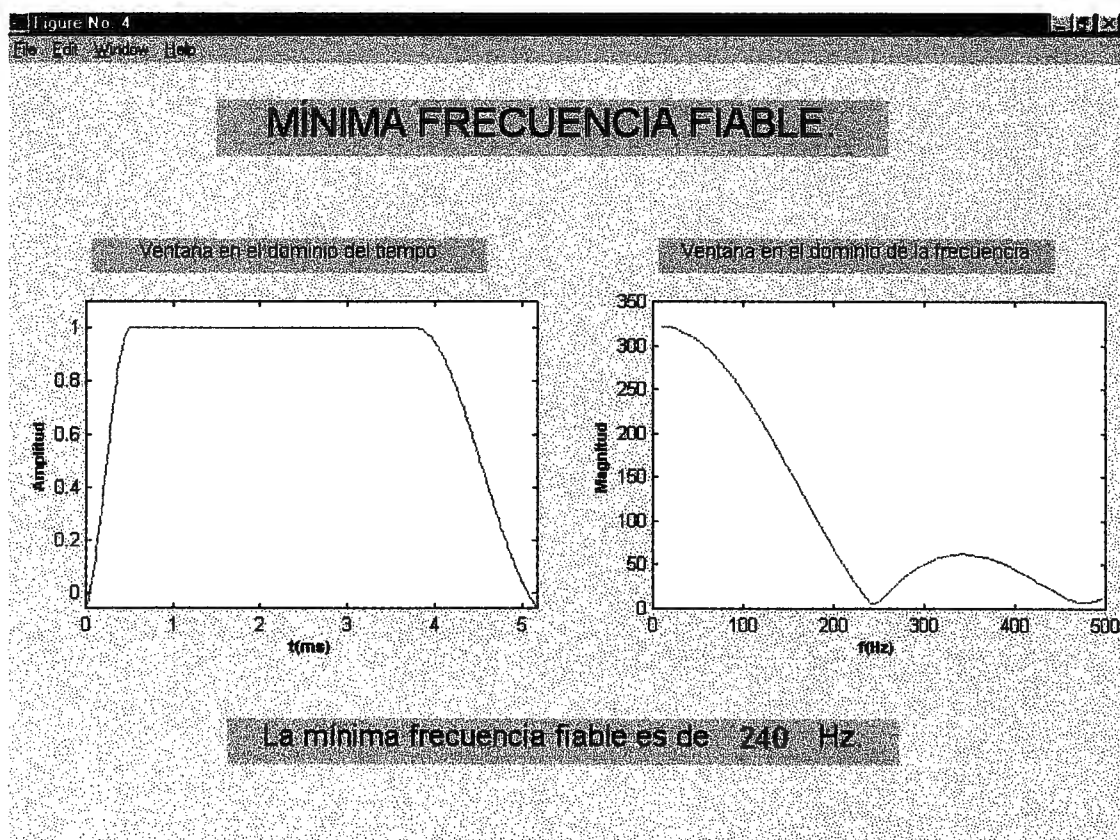


### 3.2.1.12. MÍNIMA FRECUENCIA FIABLE.

El proceso de ventaneo descrito en el apartado "1.1.3.2.2. VENTANA ADRIENNE" da como resultado un efecto de **truncamiento** en la frecuencia, de modo que, por debajo de un valor mínimo de la frecuencia, los valores de la RL no son fiables.

El valor de la mínima frecuencia fiable coincide con el primer "mínimo local", el mínimo a la frecuencia más baja, de la transformada de Fourier de la ventana Adrienne.

Para que el usuario pueda identificar claramente este mínimo, se representa la ventana Adrienne en el dominio del tiempo y en el dominio de la frecuencia. Viendo el mínimo de la ventana, en el dominio de la frecuencia, que está a la frecuencia más baja se obtiene el valor de la mínima frecuencia fiable.



Este valor también aparece expresado, en número, en la pantalla.

Este es uno de los mencionados problemas que tiene el equipo investigador del equipo Adrienne ya que no se ha conseguido llegar al objetivo inicial de dar los valores de la RL desde 100Hz a 5kHz, al tenerse problemas para alcanzar el límite inferior debido al truncamiento en frecuencia provocado por el ventaneo.



# MEDICIONES Y PRESUPUESTOS

## **4.0. ÍNDICE.**

4.0. ÍNDICE.....1

4.1. INTRODUCCIÓN.....2

4.2. COSTES DIRECTOS.....3

4.3. COSTES INDIRECTOS.....4

4.4. PRESUPUESTO GENERAL.....5

## **4.1. INTRODUCCIÓN.**

Este proyecto consiste en el diseño e implementación, mediante un programa informático, de un sistema para la obtención de la pérdida acústica por reflexión de barreras antirruido.

Se distinguirán dos clases de costes:

- **Directos**: Son aquellos en los que se contabiliza el trabajo realizado en horas según cada concepto y su valoración económica.
  
- **Indirectos**: Son los debidos a la amortización de los sistemas usados en la realización del proyecto. Se distinguirá entre costes debidos a hardware y a software.

## **4.2. COSTES DIRECTOS.**

<u>Concepto</u>	<u>horas realizadas</u>	<u>ptas / hora</u>	<u>ptas / concepto</u>
INGENIERÍA	50	8.000	400.000
ANÁLISIS	75	11.500	862.500
PROGRAMACIÓN	200	6.000	1.200.000

**TOTAL COSTES DIRECTOS ..... 2.462.500 pesetas**

### **4.3. COSTES INDIRECTOS.**



El periodo de amortización del hardware es de 4 años. Como el periodo de realización del proyecto es de 1 año, repercutirá en el coste el  $100/4 = 25\%$  del valor de los equipos.

Ordenador PC Pentium	200.000
Impresora Epson Stylus	50.000
	<hr/>
	250.000 ptas

**TOTAL HARDWARE..... $0.25 * 250.000 = 62.500$  pesetas**

El periodo de amortización del software es de 2 años. Como el periodo de realización del proyecto es de 1 año, repercutirá en el coste el  $100/2 = 50\%$  de su valor.

Matlab 5.1	450.000
Microsoft Word 7.0	25.000
Ulead iPhoto Express 1.1	40.000
	<hr/>
	515.000 ptas

**TOTAL SOFTWARE..... $0.50 * 515.000 = 257.500$  pesetas**

**TOTAL COSTES INDIRECTOS..... 320.000 pesetas**

## **4.4. PRESUPUESTO GENERAL.**

TOTAL COSTES DIRECTOS .....	2.462.500 pesetas
TOTAL COSTES INDIRECTOS .....	320.000 pesetas
	<hr/>
	<b>2.782.500 pesetas</b>

Subtotal	2.782.500 pesetas
I.V.A. (16%)	445.200 pesetas
	<hr/>
<b>Total</b>	<b>3.227.700 pesetas</b>

Se estima como coste total de este proyecto la cantidad de **tres millones doscientas veintisiete mil setecientas pesetas.**

