

[Apuntes DSD filtrado de altavoces](#)

[La importancia de la medida de la FR de altavoz](#)

[Obtención de la FR del altavoz](#)

[Herramientas para la medida de la FR del altavoz](#)

[Uso de DSD](#)

[Cortes de las vías o en los extremos: CF Crossover Filter](#)

[Parámetros](#)

[Pasaaltos en cascada](#)

[EQ del altavoz: TW Transition Window](#)

[Estrategias de ecualización](#)

[Extensión del grave de una caja cerrada](#)

[Extensión del grave de una caja bass reflex](#)

[Control de woofers pequeños usando fase lineal](#)

[Corrección del GD del grave](#)

[CFLowAsMP = true](#)

[El retardo audible en LP](#)

[Latencia](#)

[DSD para DRC digital room correction](#)

[Funciones auxiliares](#)

[wavwrite \(Octave\)](#)

[trimwav2pcm \(script de DSD\)](#)

[Algunas referencias](#)

Apuntes DSD filtrado de altavoces

DSD resuelve el filtrado de corte xover y la EQ del altavoz con filtros FIR.

Aquí intentamos introducir el uso del programa DSD. El manual del usuario detalla en profundidad sus facilidades.

La importancia de la medida de la FR de altavoz

La correcta medida de la FR (respuesta en frecuencia) del altavoz es de suma importancia para la validez del filtro calculado.

DSD procesa la magnitud de la FR para dejarla plana, con ciertos parámetros de limitación de las correcciones para adaptarlas a las capacidades físicas del altavoz, a criterio del diseñador.

Un altavoz en una caja dentro de la banda óptima de trabajo tiene comportamiento de fase mínima: las variaciones de fase se corresponden con las variaciones de magnitud. En la práctica hay que contar con fenómenos break-ups en el cono, la difracción por los bordes, sobreexcursión, etc que no son de fase mínima.

Consecuentemente, podemos afirmar que si aplicamos correcciones MP (fase mínima) para dejar la magnitud plana, la fase quedará corregida solidariamente.

Con mediciones complementarias como las curvas de impedancia o de distorsión se pueden descubrir irregularidades que pueden pasar desapercibidas en medidas FR derivadas de la respuesta a impulso.

Obtención de la FR del altavoz

El diseñador de la caja puede elegir el procedimiento que considere para la medida de la FR **respuesta en frecuencia anecoica del altavoz**, que es la base fundamental para poder aplicar una ecualización válida.

Se aconseja como referencia el siguiente documento “Loudspeaker Free-Field Response” de ARTA Labs http://www.artalabs.hr/AppNotes/AP4_FreeField-Rev03eng.pdf

Cabe considerar también que la FR obtenida se corresponda con el eje de escucha previsto en el uso de la caja. El diseñador puede considerar el promediado de respuestas en frecuencia para proporcionar la FR que servirá de base para el filtrado.

Herramientas para la medida de la FR del altavoz

La técnica de medición y de estimación de la FR normalmente consiste en medir una IR (respuesta a impulso) en condiciones semianecoicas, de manera que de esa IR se pueda aproximar la FR **respuesta en frecuencia anecoica del altavoz**.

En entornos de medida limitados no anecoicos, se debe recurrir a técnicas de medida semianecoicas, normalmente combinando una medida IR en campo lejano* de la que se elimina las primeras reflexiones a costa de perder información de graves en la FR derivada. Entonces se complementa con una medida IR en campo muy cercano (micro en el guardapolvos del woofer) que será ensamblada con la FR de la medida anterior.

* “Campo lejano” se denomina a una distancia al altavoz suficientemente grande respecto de su diámetro como para que no ocurran irregularidades en su respuesta en frecuencia. En la práctica se suele medir a 1 m.

Existen diversas herramientas para la medida y análisis, por ejemplo:

- ARTA <http://www.artalabs.hr>
- REW <https://www.roomeqwizard.com/> más orientado a DRC (room correction)
- Holm <http://www.holmacoustics.com/holmimpulse.php>
- Praxis
- Sweepscope

Uso de DSD

DSD necesita del language de programación científico Octave.

En realidad, DSD es un conjunto de scripts escritos en Octave, lo que significa que para "ejecutar" DSD, en definitiva deberemos entrar en el entorno de Octave y usar alguno de los scripts de DSD.

En DSD el diseño de la EQ del altavoz y de los cortes se definen en un archivo de texto de parametrización, nombrado por ejemplo **wo.xof** para un woofer, veamos su aspecto inicial:

```
FSInputFile="wo.frd";
GSLExp=15;    % Longitud final del FIR = 32768 taps
GSFs=44100;
...
... ..
```

Para generar el filtro FIR de, por ejemplo, un woofer cuya respuesta en frecuencia está en el archivo wo.frd (*) y cuyos parámetros de filtrado están en wo.xof entramos en Octave y ejecutamos el script RRxof de la siguiente manera:

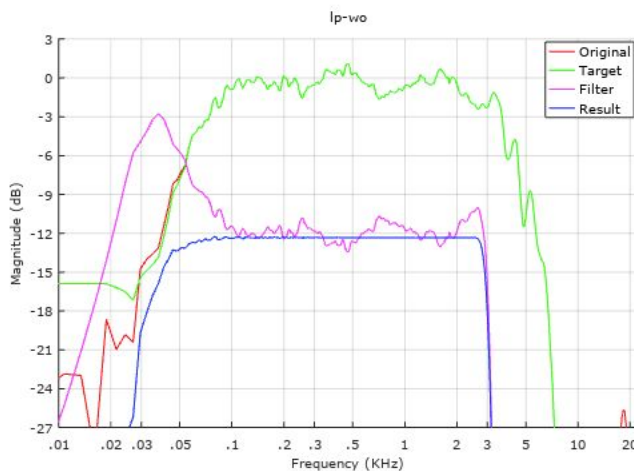
```
$ octave
GNU Octave, version 4.2.1

octave:1> RRxof tw 44100 lp

RRxof
RR Loudspeaker Crossover and Equalizer designer
(C) 2012 Roberto Ripio
Running from /Users/rafaelsanchez/DSD
GSFs = 44100
FSOutPrefix = mp-tw
Calculado en:1.4445 s

octave:2> exit

$ _
```



Nótese que se omite la extensión .xof. Se puede invocar el comando añadiendo la Fs y el tipo de filtro:

```
octave:> RRxof wo 44100 lp
```

Entonces se sobreescriben las indicadas en el archivo .xof. Así podemos invocar al programa para generar juegos de filtros mp y lp fácilmente recuperando el comando anterior.

(*) NOTA: algunos programas de medición de respuesta en frecuencia, como por ejemplo ARTA, proporcionan archivos de texto con algunas líneas de cabecera con información adicional antes de las columnas de frecuencia/magnitud. Debemos comentar dichas líneas añadiendo '%' al principio para que DSD procese correctamente estos archivos.

Cortes de las vías o en los extremos: CF Crossover Filter

DSD permite elegir '**lp**' **linear phase** o '**mp**' **minumum phase** para los cortes, bien sean los cortes de cruce de las vías o sean cortes en los extremos.

Usando **LP** se aplican pendientes muy altas que evitan el solape acústico, y sin distorsiones de fase en la banda de transición.

Los filtros **MP** son equivalentes a los tradicionales, podemos elegirlos de tipo Butterworth de cualquier orden o Linkwitz-Riley de cualquier orden par. Habrá un solape acústico entre las vías, como ocurre con los filtros tradicionales (pasivos, activos, o digitales IIR).

Parámetros

El filtrado se lleva a cabo indicando un valor distinto de cero en CFLowF o CFHighF.

Nótese que los parámetros CFLowType/Order, CFHighType/Order solo aplican a las variantes CFCClass='mp' (minimum phase).

Debe tenerse en cuenta que, si bien los woofers o los tweeters no tienen tradicionalmente corte en el extremo de la banda, el filtrado digital permite o aconseja poner cortes en esos extremos, a criterio del diseñador, para algunas estrategias de ecualización de fase en graves, o en el extremo agudo para mejorar los filtros de reconstrucción de los convertidores DAC (*apodizing filters*).

Ejemplo para un woofer:

```
% SIN CORTE POR ABAJO
CFLowF = [0];

% CORTE DE CRUCE
CFHighF = 3000;
CFHighType = 'LinkwitzRiley';
CFHighOrder = 4;
```

Pasaaltos en cascada

Conviene reproducir el efecto de un filtro activo tradicional (de fase mínima) en cascada que es la topología adecuada para alinear los retardos acústicos consecuencia de todos los pasaaltos del sistema.

Con el fin de incorporar a una vía los retardos MP de los pasaaltos de las vías inferiores, es posible añadir las frecuencias, tipo de filtro y orden de los pasaaltos precedentes en la sección CFLow del cruce pasaaltos actual.

Veamos un ejemplo para un sistema de tres vías con “protección subsónica” en el woofer y cortes xover MP de tipo LR4.

Nota: solo se muestran las secciones relevantes CFxxx de los cortes.

--- wo.xof ---

```
% CORTE SUBSÓNICO DE PROTECCIÓN
CFLowF = [10];
CFLowType = {'Butterworth'};
CFLowOrder = [2];
CFLowAsMP

% CORTE DE CRUCE PASABAJOS a 1500 Hz
CFHighF = 1500;
CFHighType = 'LinkwitzRiley';
CFHighOrder = 4;
```

--- mid.xof ---

```
% CORTE DE CRUCE PASAALTOS a 1500 Hz, le añadimos el precedente:
CFLowF = [1500, 10];
CFLowType = {'LinkwitzRiley', 'Butterworth'};
CFLowOrder = [4, 2];

% CORTE DE CRUCE PASABAJOS a 3500 Hz
CFHighF = 3500;
CFHighType = 'Butterworth';
CFHighOrder = 4 ;
```

--- tw.xof ---

```
% CORTE DE CRUCE PASAALTOS a 3500 Hz, y añadimos los precedentes:
CFLowF = [3500, 1500, 10];
CFLowType = {'LinkwitzRiley', 'LinkwitzRiley', 'Butterworth'};
CFLowOrder = [4, 4, 2];

% CORTE SUPERSONICO (RECOMENDABLE)
CFHighF = 20000;
CFHighType = 'Butterworth';
CFHighOrder = 4;
```

Nota: DSD requiere que los valores de los pasabajos CFLowF/Type/Order sean arrays, por lo que siempre usaremos corchetes (para valores numéricos) o llaves (para los tipos de filtro) como en el ejemplo.

EQ del altavoz: TW Transition Window

El conjunto de parámetros TWxxx de DSD define los límites de la EQ del altavoz.

DSD ecualiza el altavoz con MP minimum phase, como se ha discutido al inicio.

La estrategia general de ecualización que lleva a cabo el programa puede implicar ecualizaciones con ganancias muy elevadas si se intenta ecualizar fuera de la banda pasante. Para poner límites a esta ecualización está el grupo de parámetros TWxx.

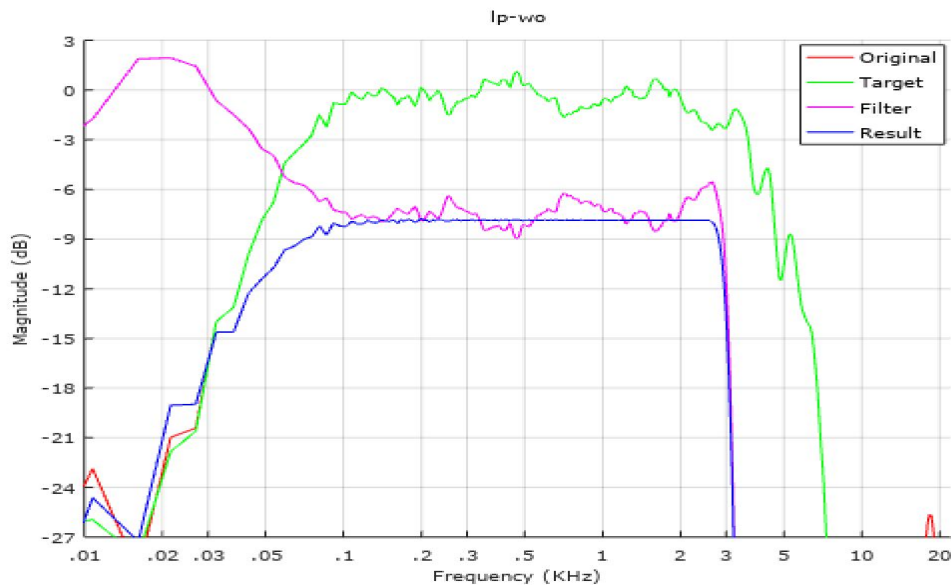
Ejemplo:

```
% ECUALIZACION: alcances genericos
TWFlatInterval=2;          % oct después del xo hasta que se ecualiza plano
TWTransitionInterval=2;    % oct adicionales para ir dejando de ecualizar

% ECUALIZACION: límites sobre los alcances genericos %
TWLimitLowF1 = 10;         % Lím. absoluto de la ecualización (Low)
TWLimitLowF2 = 40;         % Lím. absoluto a ecualizar plano (Low)
TWLimitHighF1 = 15000;     % Lím. absoluto a ecualizar plano (High)
TWLimitHighF2 = 20000;     % Lím. absoluto de la ecualización (High), se
                           % recomienda menor que la frec. max del .frd
```

NOTA: Puede ser conveniente que TWLimitHighF2 sea inferior a la mayor frecuencia del archivo de datos .frd, en el caso de que se generen filtros para una frecuencia de muestreo superior a la usada al hacer la medida.

Un ejemplo de ecualización de un mid-woofer que se muestra al ejecutar DSD:



Estrategias de ecualización

Extensión del grave de una caja cerrada

El rolloff acústico de los graves de una caja cerrada es suave de 2º orden. Podemos extender el grave ajustando en DSD los límites de ecualización `TWLimitLow` al estilo de la “transformada de Linkwitz”, a base de excursion del driver, con cuidado de no exceder los límites del altavoz y del amplificador, ejem...

Extensión del grave de una caja bass reflex

En una caja réflex es difícil extender el grave aplicando ganancia por debajo de la F_b frecuencia de sintonía del reflex. El motivo es la pendiente rolloff acústico de 4º orden que existe por debajo de la F_3 de la caja, cercana a la F_b . Lo más probable es que excedamos la excursion del cono y aparezca mucha distorsión.

Control de woofers pequeños usando fase lineal

`CFLowAsMP` permite dar un target butterworth de fase lineal a un pasaaaltos de ‘protección’ de un woofer. Se emula la respuesta de magnitud del target MP especificado en `CFLowType/Order`.

Así tendremos la misma respuesta acústica con ambos juegos de filtros ‘mp’ y ‘lp’.

Esto es de ayuda para aplicar una corrección de graves DRC válida para ambos juegos de filtros.

Corrección del GD del grave

El pasaaaltos natural de una caja de graves introduce distorsión de fase que implica un GD variante a medida que descendemos en graves. Esa distorsión es audible: afecta a coherencia temporal percibida en el grave.

Una caja cerrada introduce un cambio de GD relativamente progresivo y suave.

Una caja bass reflex presenta una rotación de fase más irregular.

Si corregimos esto, el resultado será un grave percibido muy coherente.

Se necesita relativamente poca resolución es decir filtros FIR no demasiado largos, pues se ecualiza una curva suave sobre todo en el caso de una caja cerrada.

El proceso de EQ ecualización en fase mínima, de la magnitud decreciente hacia los graves que presenta la FR, introduce una compensación de fase que consigue lo que queremos, siempre y cuando indiquemos al algoritmo que ecualice hasta muy abajo por ejemplo 5 Hz.

La desventaja de esto es que como indicamos al algoritmo que ecualice hasta muy abajo, resultan entonces unas ganancias muy altas, posiblemente serán físicamente insoportables.

Para proteger al amplificador y evitar sobreexcursiones del woofer, podemos añadir un corte pasaaltos Butterworth a modo de filtro subsónico limitador de ganancias, por ejemplo a 20Hz si estimamos que nuestro altavoz lo soportará. Usaremos Butterworth por introducir menos distorsión de fase lo que es de interés en el extremo inferior de la banda, como se ha discutido arriba.

Dicho pasaaltos en el woofer es opcional y tiene como objetivo limitar los efectos secundarios de ecualizar hasta muy abajo. Esto es fácilmente observable en las gráficas que nos muestra DSD al computar un filtro.

NOTA: es muy importante que la FR estimada del archivo *wo.frd* no esté contaminada con resonancias de la sala. Es crítico disponer de una buena FR de estimación semianecoica en graves.

CFLowAsMP = true

Este parámetro es de aplicación en el woofer y tiene efecto al calcular el filtro LP de la sección opcional CFLow que hemos comentado arriba para el woofer. Se precisa que dicha sección se defina con CFLowType={ 'Butterwothrh' }

Este parámetro indica si la variante LP de la sección CFLow debe seguir la misma curva de magnitud de la variante MP, de modo que la respuesta acústica sea idéntica para MP y LP.

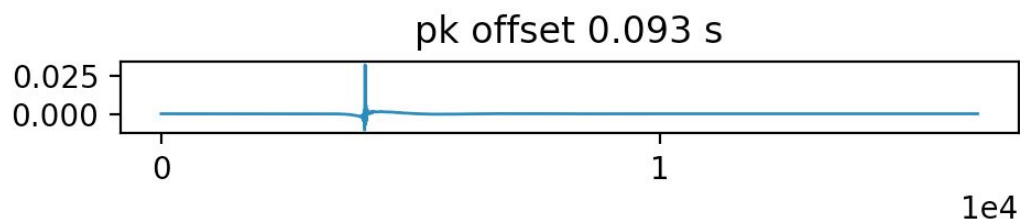
El filtro LP resultante tiene la ventaja de no alterar la fase del conjunto al ser éste un corte pasaaltos de tipo subsónico de fase lineal, como se ha comentado arriba.

El retardo audible en LP

En la modalidad de fase lineal, el FIR calculado es resultado de combinar una ecualización del altavoz en MP con la aplicación de cortes xover o cortes en los extremos de tipo LP.

Esto se traduce en un impulso con el pico NO centrado en la mitad de la longitud del FIR como ocurriría con un FIR de fase lineal pura. Ejemplo:

```
GSLExp = 14; % longitud final 16K taps
GSFs = 44100;
CFClass = 'lp'
```

El pico está desplazado 4K taps y el retardo audible será de 93 ms + la latencia de procesado.

Latencia

Como cualquier proceso DSP, habrá una latencia directamente relacionada con la potencia de la CPU y los buffers de I/O de la tarjeta de sonido. Las cifras típicas van de pocos milisegundos a alguna decena.

Si hay potencia de CPU suficiente puede ajustarse convolución particionada en el convolver Brutefir para reducir la latencia de procesamiento de la convolución.

DSD para DRC digital room correction

Se dispone del script RRreq para generar un FIR de DRC a partir de una medida IR (impulso) tomada en el punto de escucha previsto.

El formato del archivo del IR puede ser .wav, .pir (ARTA), .pcm (raw float) o .txt (raw text values).

Al igual que hicimos con script de corte y eq de vías, debemos elaborar un archivo de texto con los parámetros con que se calculará la ecualización.

Ejemplo de archivo de parametrización .req para generar el FIR de DRC para un canal:

```
--- L.req ---
```

```
%% User Data for RR_req.m octave script
```

```
% FSInputFile Impulse filename, whether .wav, ARTA .pir,
%             raw float .pcm or raw text .txt.
%             These extensions are mandatory for each file type.
FSInputFile="L.wav";
```

```
% FSFs Sampling frequency of input impulse (not *final* filters)
% Only applies for raw, not used if FSInputFile is a .wav or .pir file.
FSFs=48000;
```

```
% GSFs *final* filters
GSFs=44100;
GSLExp=15;           % usually 2^15 = 32K taps
```

```
% EF Equalization Filter: Reference dB level for magnitude inversion
%                         (it depends on measured level)
EFRef=-2.5;
```

```

% EF Equalization Filter: Avoid equalization of upper range
EFBassOnly=false;

% TW Transition window: Schroeder range
TWSchroeder=200;          % End of modal range (Hz) (Schroeder freq.?)
TWIntervalSchroeder=1;    % octaves of progressive elimination of eq.

% TR Target Response: Bass rolloff
TRLowXoF=0;               % Target HP Butterworth xover freq
%                           (use 0 for no crossover)
TRLowXoOrder=2;           % Target HP Butterworth xover order (usually 2)
TWFlatInterval=2;         % oct después del xover hasta que se ecualiza plano

% TR Target Response: Room curves (House_curve, Room_curve, Linkwitz_DSS)
TRHouseCurve=false;       % Target House Curve.
TRH_f_corner=500;         % Frecuencia en la que empieza a bajar la curva.
TRH_atten=6;              % Atenuación a 20kHz.
TRRoomGain=false;         % Target Room Gain.
TRR_gain=6;               % Ganancia en DC sobre la respuesta plana.
TRDSS=false;              % Target Linkwitz DSS curve.

% TW Transition window: EQ scope and limits
TWLimitLowF1=20;          % Lím. absoluto de la eq (Low) < TWLimitLowF2
TWLimitLowF2=40;          % Lím. absoluto eq plano (Low)
TWLimitHighF1=15000;      % Lím. absoluto eq plano (High)
TWLimitHighF2=20000;      % Lím. absoluto de la eq (High) > TWLimitHighF1

```

... PONER AQUÍ UNA GRAFICA

Funciones auxiliares

wavwrite (Octave)

NOTA: esta función está deprecated en Octave 4.x se propone usar **audiowrite**

La función `wavwrite` vuelca a un archivo wav el array de valores float que componen los taps de un filtro FIR.

En `FIRtro` tiene interés para convertir a wav el archivo de rendered que proporciona el módulo 'eq' de `Brutefir`. Es el módulo usado para las ecualizaciones suaves de tonos y loudness. `Brutefir` proporciona un archivo de texto que contiene los valores numéricos de los taps que internamente hacen el proceso del módulo 'eq', cada tap en una fila.

Primero lo leemos en octave, cargando el archivo con la orden "load" en una variable array. Luego la variable se exporta con "wavwrite". Por ejemplo:

```
octave:1> load "rendered-0"
```

(carga el archivo de texto en la variable "rendered_0". Ojo que el - se cambia a _ para no confundirlo con un operador de resta)

```
octave:2> wavwrite (rendered_0, 44100, "rendered_0.wav");
```

(Salva la variable a un wav 16 bits con fs=44100)

Si se quiere otra profundidad de bits, añadir la tercera opción BPS de wavwrite(Y, FS, BPS, FILENAME)

trimwav2pcm (script de DSD)

```
octave:1> trimwav2pcm <nombrearchivo> (sin extensión)
```

Es útil para el caso de REW Room Equalizer Wizard que proporciona el filtro de DRC en forma de impulso WAV largo, de 128K. Entonces con esta función podemos recortarlo con ventana, para adaptarlo a la longitud de nuestro convolvedor, por defecto **2^15 (32K)**. OjO editar el propio script **trimwav2pcm.m** para otras longitudes.

Algunas referencias

<http://dspguru.com/dsp/faqs/fir>

buscar más

