

TRABAJO FINAL DE TDSÑ: **ECUALIZADOR PARAMÉTRICO**

Jorge Calatayud Maeso

Ángela Burgaleta Ledesma

Grupo 33.

Indice

1.- Descripción del problema	2
2.- El ecualizador paramétrico	2
3.- Desarrollo del trabajo	3
4.- Comprobaciones	12
5.- Simulación del Sistema usando Señales de Audio	15
6.-Anexos	17

1.- Descripción del problema

Nos encontramos ante una sala, que debido a sus características supone una distorsión a la hora de querer grabar cualquier tipo de audio. La **distorsión** es la alteración de la forma de una señal cuando pasa a través de un sistema. La alteración ocurre cuando el sistema actúa de diferente manera sobre las componentes de la señal, cambiando amplitud, fase o frecuencia en desigual proporción.

En lo que se refiere a audio podemos tener distorsiones debidas al amplificador o fuera del amplificador (este es el tipo que contemplamos en el trabajo). Dentro de estas últimas tenemos el efecto doppler (los sonidos graves requieren más potencia que los agudos), la distorsión de fase del recinto (al oyente le llegan antes los sonidos agudos que los graves) y la frecuencia mínima de la habitación (la nota más grave que es capaz de reproducir una habitación depende de sus dimensiones).

En resumen, para conservar la calidad del audio es necesario compensar esta distorsión y para ello la herramienta que vamos a utilizar es el ecualizador.

2.- El ecualizador paramétrico

Un ecualizador es un dispositivo que nos permite aumentar o reducir la potencia o amplitud de determinadas frecuencias de manera individualizada, con el objetivo de corregir esas desviaciones de la respuesta frecuencial que se producen en nuestro sistema de sonido.

Es decir, si medimos la respuesta de nuestro equipo y se observa una resonancia de 4 dBs en torno a los 100 Hz y un valle de 6 dBs en 1 KHz, el ecualizador nos permitirá corregirlo reduciendo 4 dBs a 100 Hz y aumentando 6 dBs a 1 KHz para así obtener una respuesta más plana y fiel de todo lo que reproduzcamos.

Hemos de tener claro que cualquier equipo de sonido junto a la acústica de la sala en la que suena, van a alterar en mayor o menor medida el espectro frecuencial de la señal original.

Dependiendo de la acústica de la sala y la calidad del equipo, nos encontraremos que se refuerzan o atenúan indeseadamente ciertas frecuencias, dando un toque más frío, o más cálido, o más irregular, provocando una mayor fatiga auditiva, una menor claridad, una ausencia de ciertos detalles, etc y por lo tanto *coloreando* la grabación original.

Una prueba muy ilustrativa (y algo que suele hacerse en la realidad) para comprobar cuál es la respuesta en frecuencia de un equipo de sonido, consiste en reproducir una grabación de un ruido blanco (ruido que contiene la misma potencia a

todas las frecuencias y por tanto la gráfica de su espectro es plana) y medir con un micrófono y un analizador el espectro del sonido que realmente se está recibiendo en el punto de escucha. Como el equipo y la sala ideal no existen, siempre se detectarán en mayor o menor medida irregularidades en la respuesta.

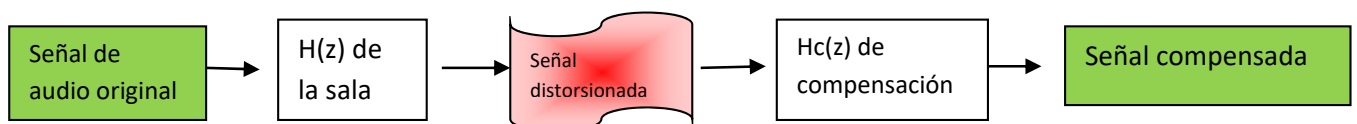
Los tipos de ecualizadoras según su funcionalidad se dividen en: ecualizador shelving, ecualizador semiparamétrico, ecualizador paramétrico y ecualizador software, siendo el ecualizador paramétrico el más completo de todos, ya que permite elegir tanto la frecuencia de actuación de las ganancias, como su ancho de banda o su factor Q, que permite hacer que la ganancia afecte a más o menos frecuencias en torno a la frecuencia central seleccionada en cada banda, facilitando una ecualización con gran detalle y precisión.

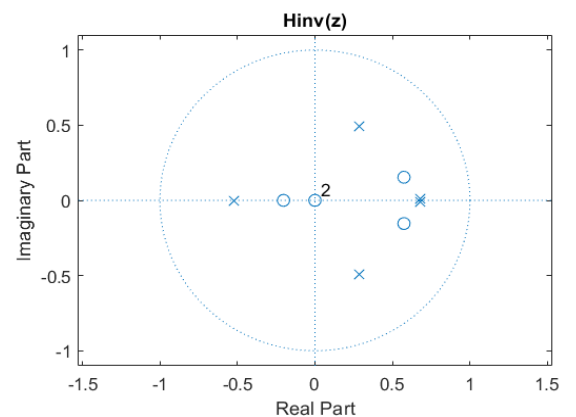
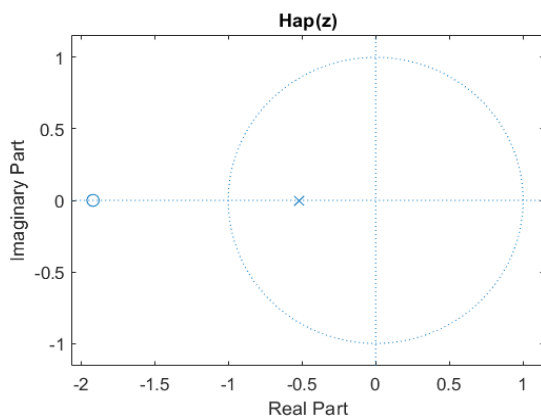
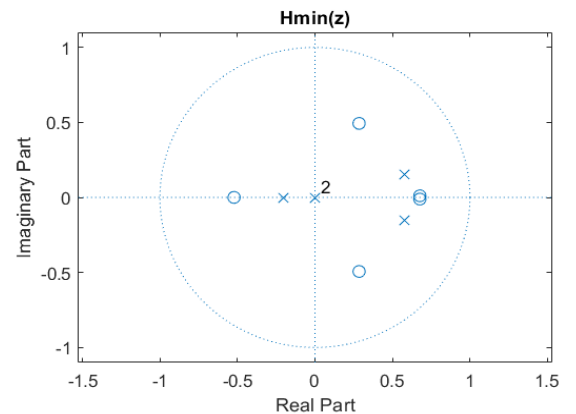
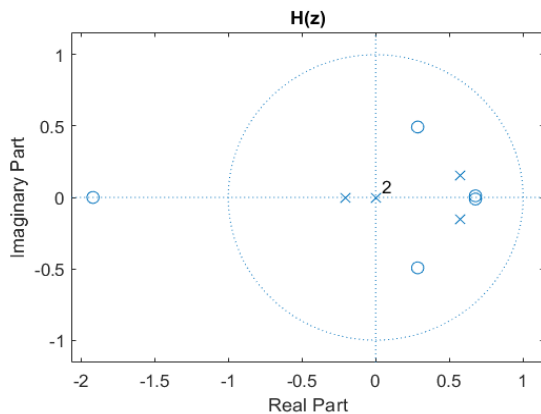
Aparte del uso para corregir la respuesta de un equipo de sonido, los ecualizadores también son usados para otros objetivos; como filtrado, por ejemplo para reducir un ruido a una frecuencia muy localizada, o como efecto creativo en la producción musical para manipular el espectro de ciertas pistas musicales o instrumentos, esto último ahora mismo no nos concierne. La funcionalidad del filtrado es la que vamos a considerar en nuestro trabajo.

3.- Desarrollo del trabajo

En primer lugar, Matlab nos va a proporcionar la función de distorsión asociada a la sala, vamos a considerar que representa cualquier tipo de distorsión indeseada. Esta función será la que necesitemos compensar, para ello pondremos en cascada la función de la sala con la función de compensación, esto es posible porque ambos sistemas son LTI.

La función de compensación ideal es la función inversa a la de la sala, no obstante debemos tener en cuenta un par de cosas: Que la función de la sala sea causal y estable no significa que su inversa lo sea, en general no existe una compensación perfecta y normalmente solo se puede compensar la distorsión de amplitud pero no la de fase. En definitiva, el objetivo es intercambiar los polos por los ceros.





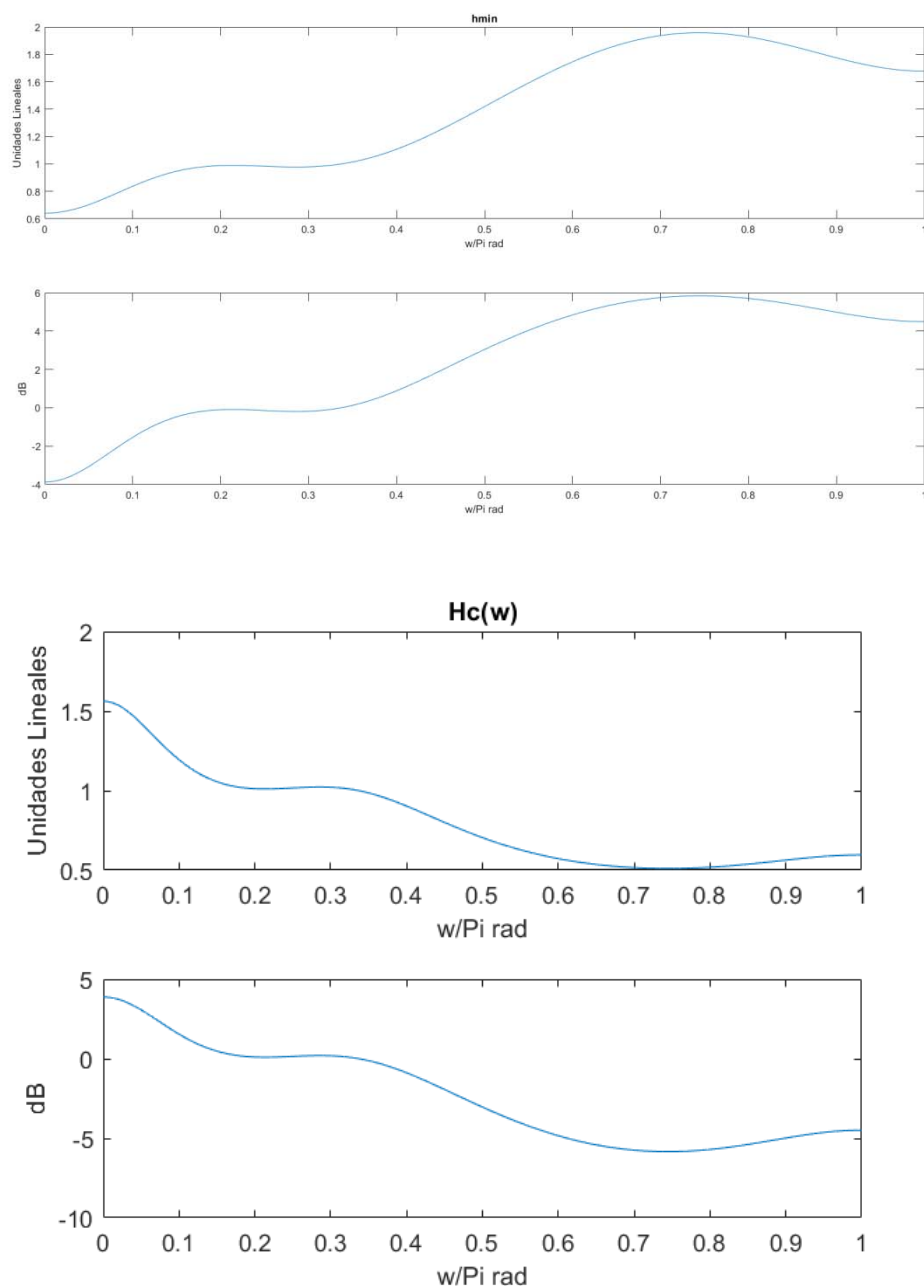
Una vez tengamos la función inversa teórica vamos a intentar replicarla mediante una cadena de cinco filtros. Un filtro paso bajo, tres filtros paso banda y finalmente un paso alto. Para digitalizar los prototipos analógicos se utiliza la Transformada Bilinial, esta herramienta matemática suele usarse para transformar la representación en tiempo continuo de las señales en tiempo discreto y viceversa.

La utilizaremos con el objetivo de convertir una función de transferencia de un filtro lineal e invariante definido en el dominio continuo del tiempo, en una definida en el dominio discreto del tiempo (también llamados filtros digitales). Para implementarlo hay que tener en cuenta la deformación de frecuencias y la compresión que experimenta el ancho de banda al asignarlo de analógico a digital.

La implementación más directa es la forma directa 1. Los coeficientes pueden obtenerse mediante el ancho de banda en octavas (BW) o el factor de calidad (Q). Nosotros hemos escogido la técnica del ancho de banda por parecernos más sencilla. Además también se debe fijar la frecuencia central de cada filtro, no debemos olvidar que al fin y al cabo lo que tratamos de hacer es concatenar cinco filtros, de forma que saquemos una función de transferencia lo más parecía posible a la teórica. Una vez teniendo estos conceptos claros, apoyándonos en los valores de los parámetros de

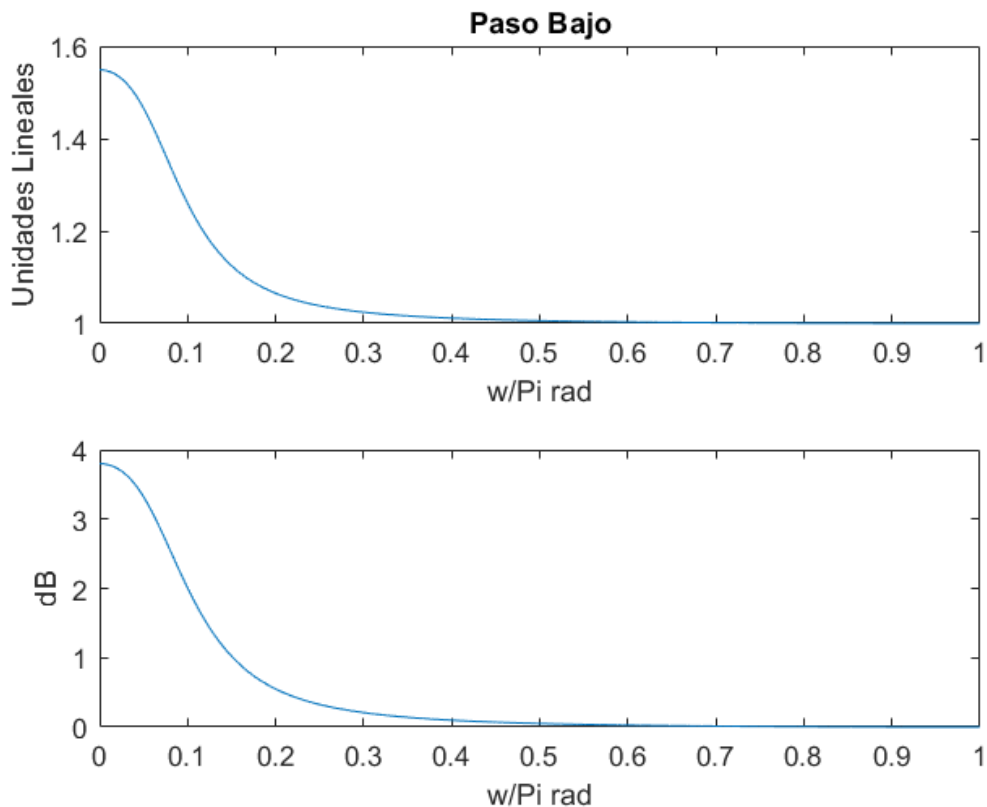
diseño admitidos y sabiendo que la frecuencia de muestreo del ecualizador es 48 kHz, todo consiste en ir probando hasta obtener un resultado satisfactorio.

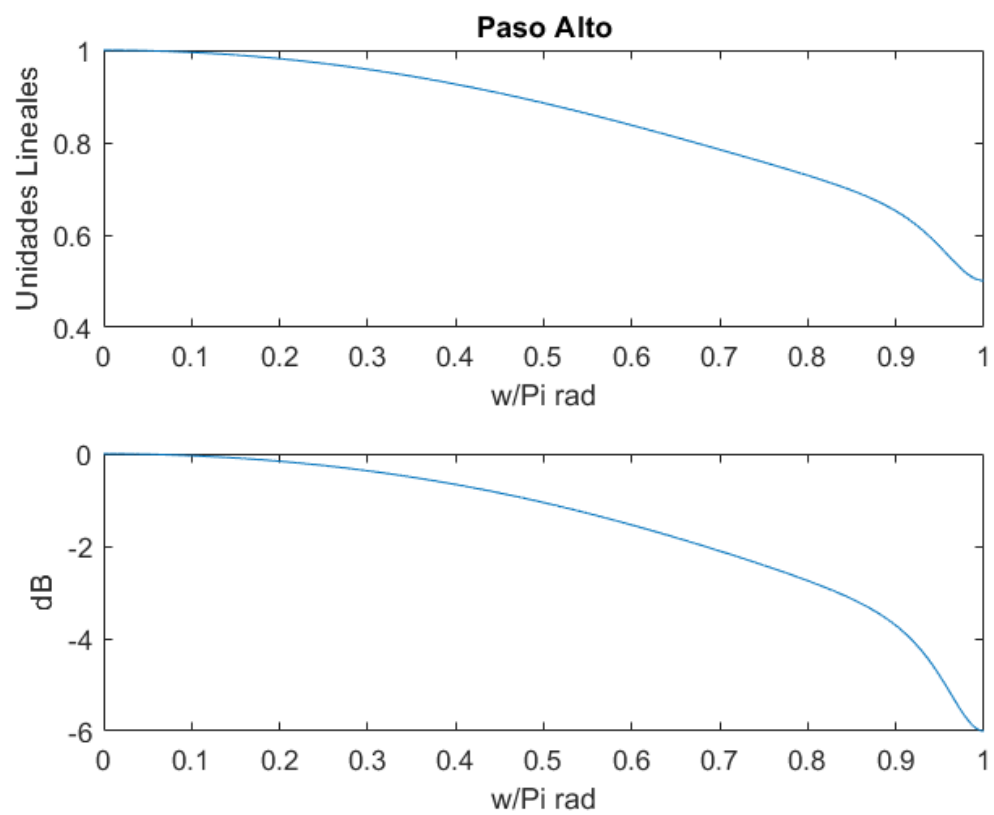
Para ajustar los diferentes filtros, nos hemos basado en la función inversa a H_{\min} , H_c , para intentar imitarla lo mejor posible con nuestro conjunto de 5 filtros. A la hora de colocar los filtros y fijar su ancho de banda, nos hemos fijado en las caídas de la función inversa perfecta, y hemos intentado imitarlas, fijando las frecuencias de corte de los filtros en aquellos sitios donde la caída es a 3 Db. Esto ha sido de gran ayuda a la hora de colocar los filtros paso bajo y paso alto, sin embargo, a la hora de compensar las frecuencias intermedias, ha sido un proceso de ensayo y error.

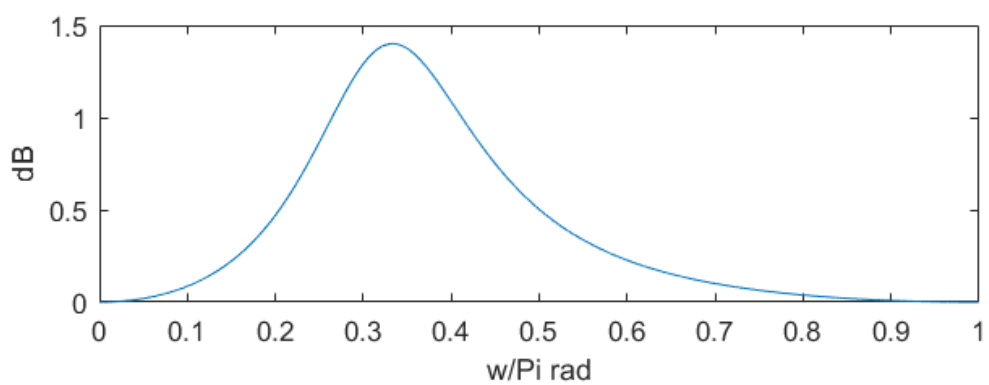
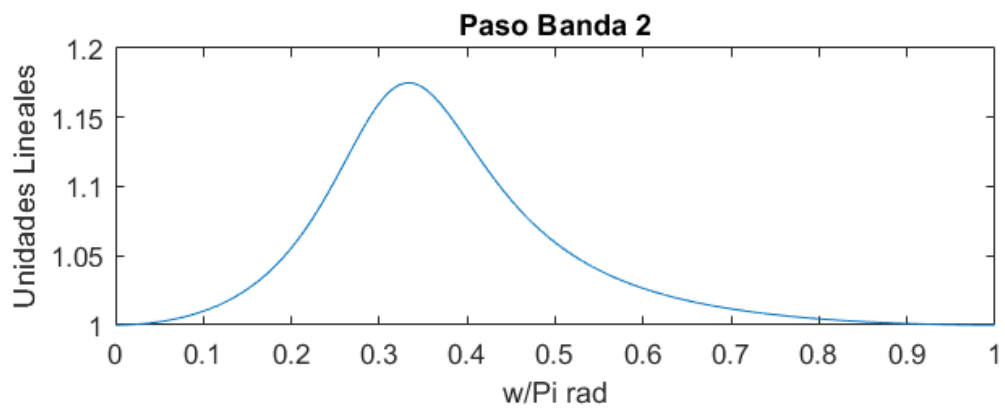
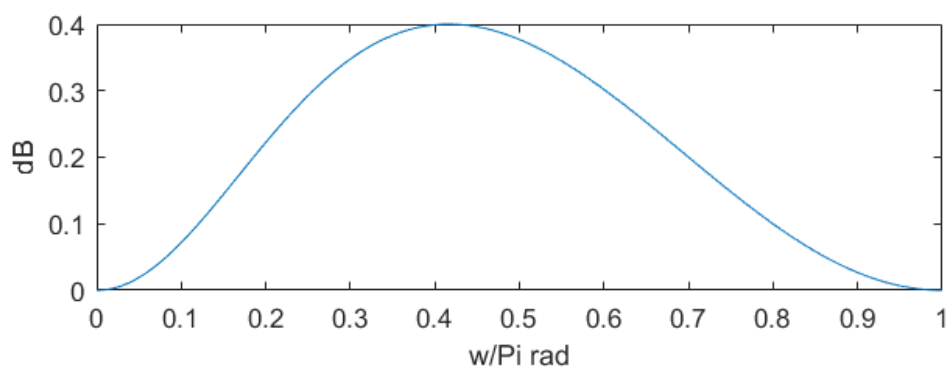
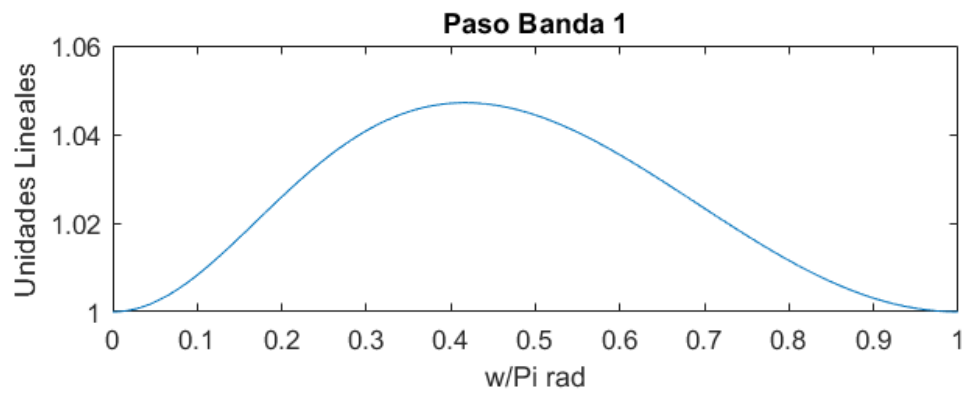


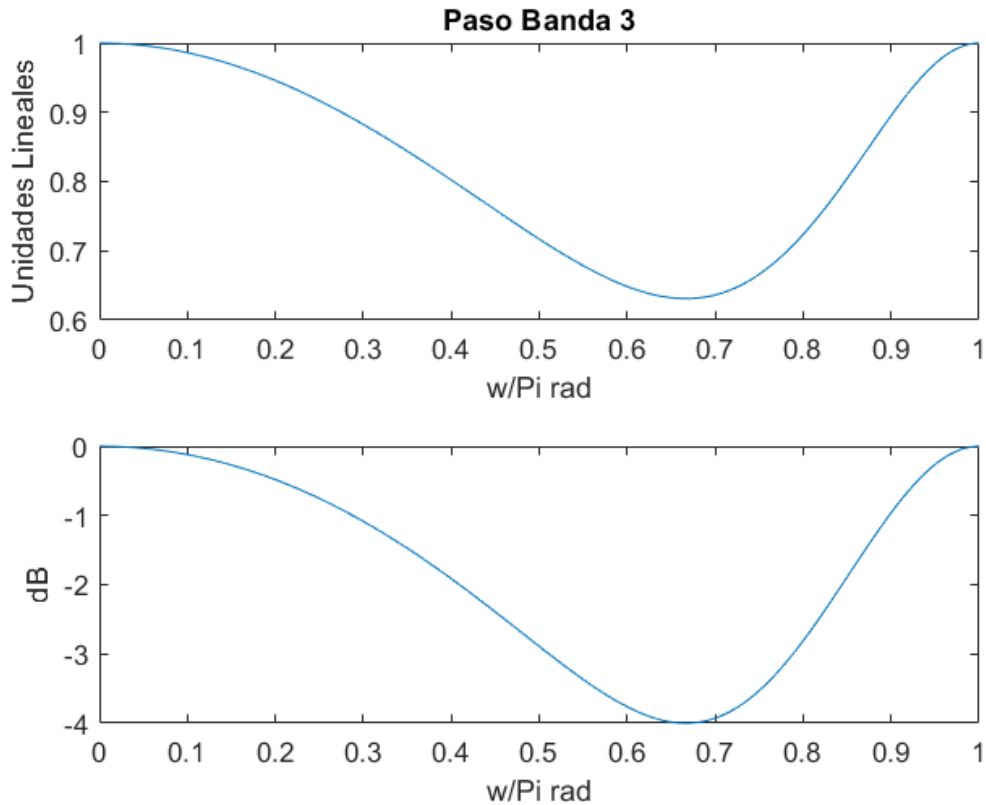
Para modularizar el código, hemos programado, cada tipo de filtro en una función separada, lo que nos permite además evitar redundancias innecesarias del código. Dichas funciones calculan los coeficientes a_n y b_n , que son los parámetros de retorno de las mismas. Como ya hemos explicado antes, dichos coeficientes han sido calculados a partir de los parámetros especificados en la guía de diseño, siguiendo el criterio del ancho de banda de los filtros.

Bw (Octavas)	Frecuencia central/Frecuencia de corte (kHz)	Fm(kHz)	Ganancia(dB)	Tipo de filtro
2.2	2.5	48	3.8	Paso Bajo
2/3	20	48	-6	Paso Alto
2	10	48	0.4	Paso Banda
1	8	48	1.4	Paso Banda
1	16	48	-4	Paso Banda





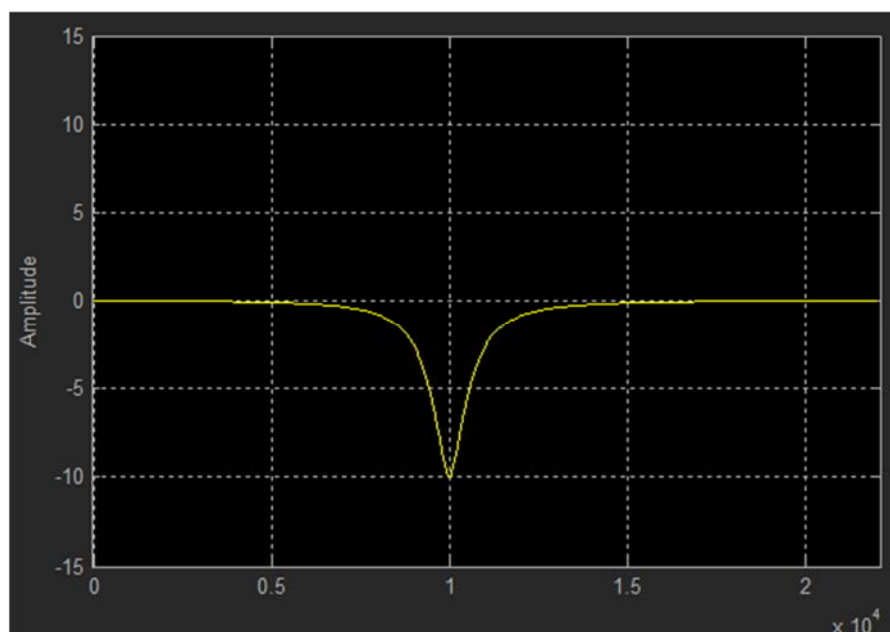
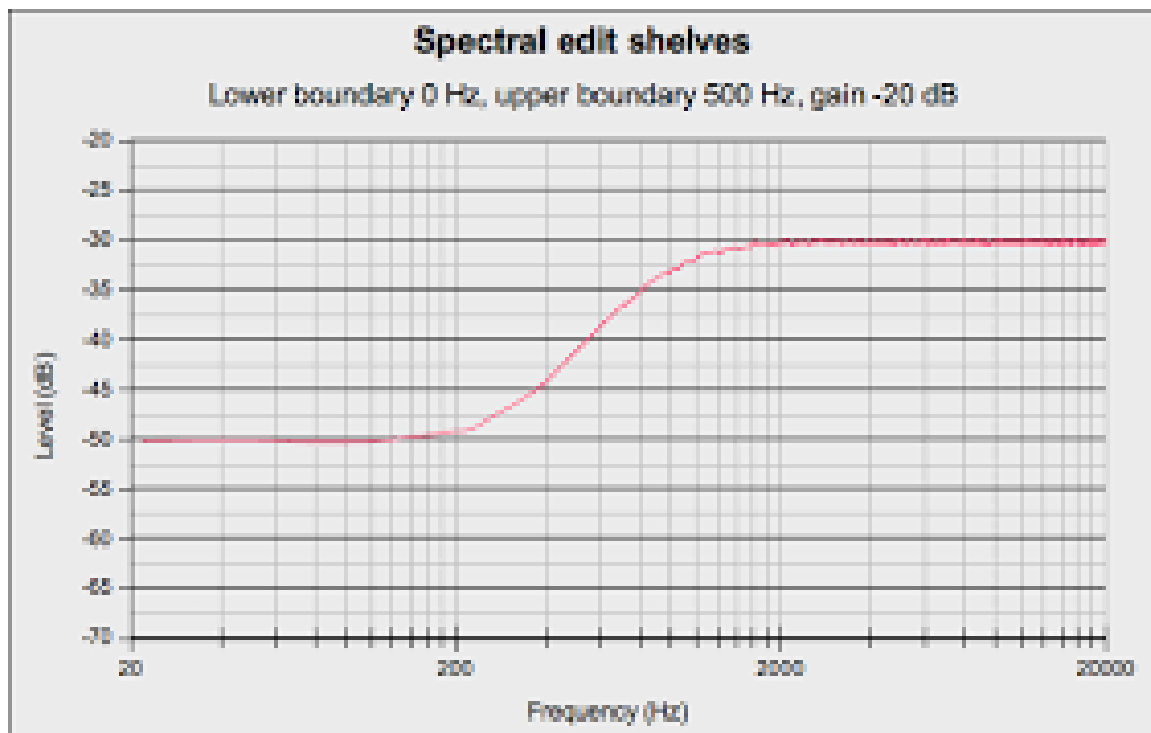




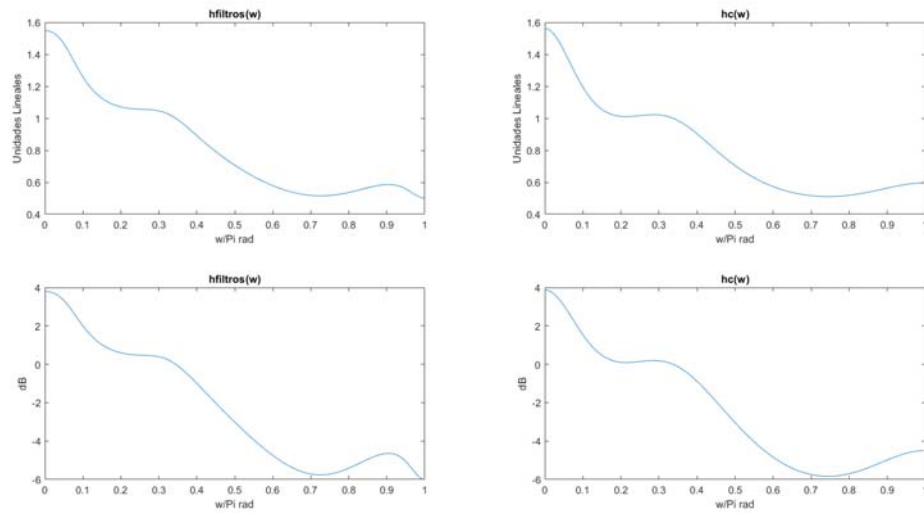
Para facilitar la representación del conjunto de filtros, hemos creado la función `concatfilters`. Esta sencilla función recibe de parámetros, dos matrices con los coeficientes A y B de cada filtro, como filas. Iterando sobre esta matriz, y calculando el equivalente en frecuencia de los coeficientes, gracias a `freqz`, juntamos todos los filtros en una única $H(w)$. Esto se logra multiplicando todos los filtros entre sí:

$$H_{filtros}(w) = \prod_0^4 Hf(w)_i$$

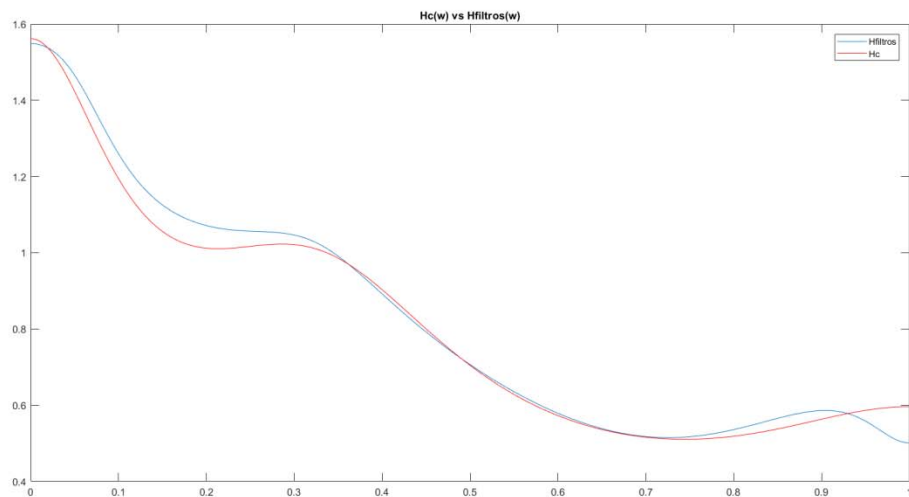
Esto es posible debido a que usamos filtros Shelf y Peak. Estos filtros se caracterizan por tener una ganancia unidad en la zona que no queremos filtrar, por lo que son neutros a la multiplicación fuera de su zona de intervención.

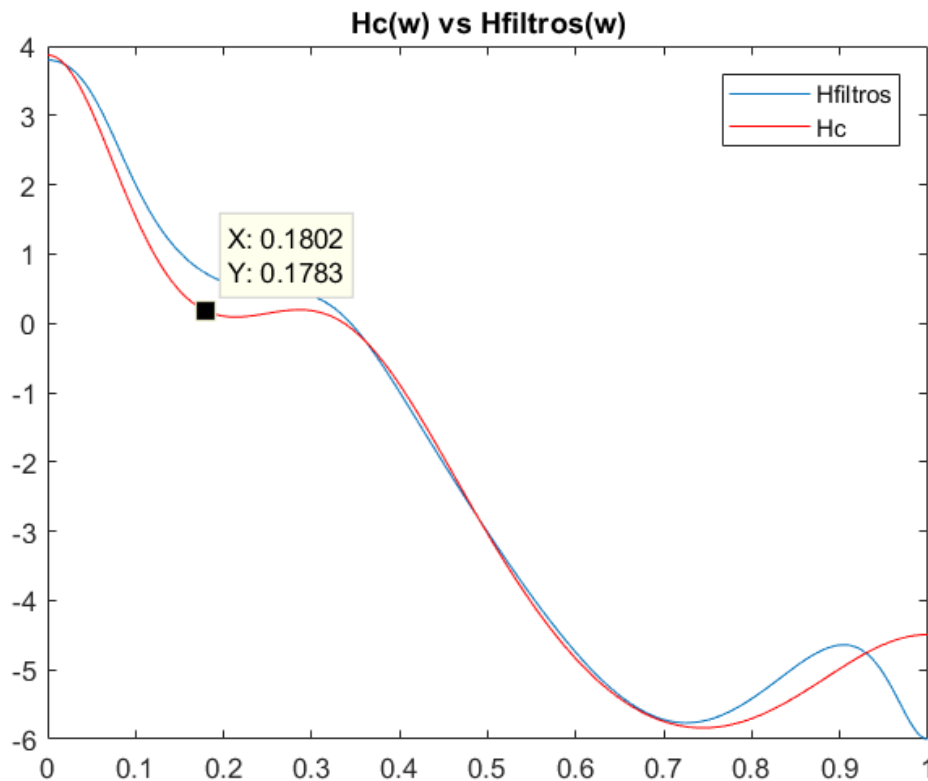


Finalmente el resultado de $H_{\text{filtros}}(w)$ es el siguiente:



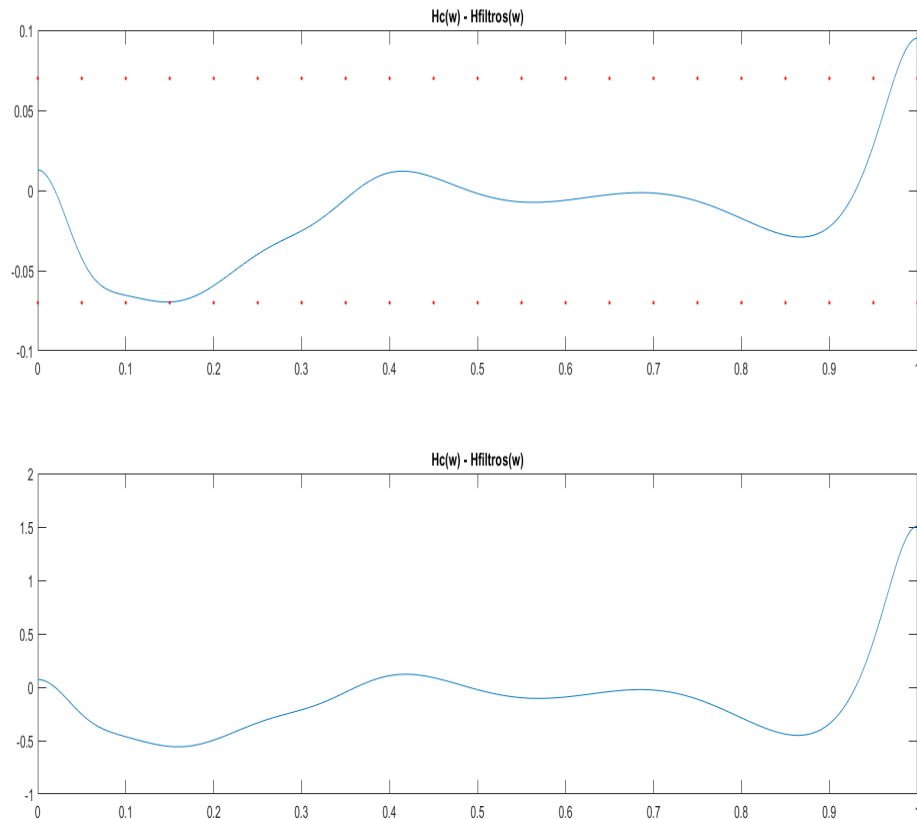
Una forma de comprobar, lo buena que es nuestra implementación, es sobreponer sobre una misma grafica el filtro ideal con el conseguido y evaluar si son muy diferentes.





En nuestro caso, hay una ligera diferencia en la zona acotada por $[0,1 \ 0,3] \left[\frac{\omega}{\pi}\right]$ lo que equivale al rango de frecuencias de $[2,4 \ 7,2]$ [kHz] que no hemos sido capaces de compensar de ninguna forma. Hemos intentado jugar con el ancho de banda de los filtros y la frecuencia de corte de los mismos, o introduciendo un filtro paso banda con ganancia negativa, pero finalmente esto ha sido lo más cerca que hemos conseguido llegar de la función. Sin embargo, el error que cometemos en esta zona está acotado entre ± 0.07 en unidades lineales. Para ver este error hemos representado en las siguientes graficas la diferencia entre el filtro ideal, y el que hemos conseguido.

En las frecuencias del final el error se dispara, pero no nos importa, puesto que este error se corresponde con frecuencias superiores a los 20 KHz, por lo que no son percibidas por la mayoría de las personas.



4.- Comprobaciones

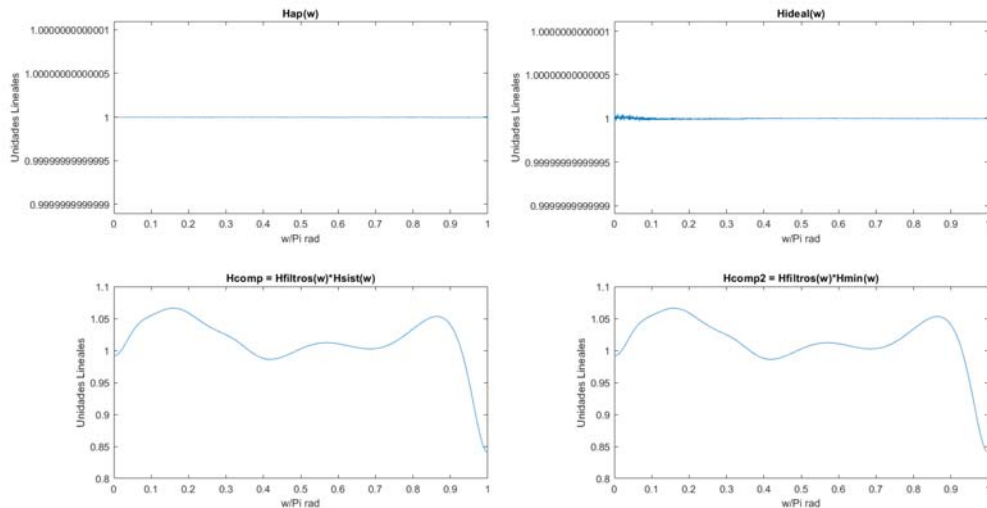
Antes de proceder a evaluar el funcionamiento del equalizador, queremos cerciorarnos de se han seguido las directrices de diseño adecuadas. Para ello comenzamos comprobando que se cumplen las ecuaciones de diseño a partir de los respuestas al impulso de los diferentes sistemas.

$$K * H(z) = K * AH_{min}(z) * BH_{ap}$$

$$H_c(z) = \frac{1}{H_{min}(z)}$$

Primero, procedemos a comprobar con las respuestas al impulso ideales que hemos sacado a partir de los diagramas de polos y ceros ya descritos. H_{ideal} es el resultado de concatenar, el sistema con la H_c ideal, dando lugar a H_{ap} , presenta una pequeña oscilación en torno a 1 al principio, que puede ser por falta de precisión de la máquina para ser 1 exacto, puesto que las sucesivas conversiones, van generando un error que se acumula. También comprobamos el funcionamiento de $H_{filtros}$, para ver funciona adecuadamente como inversa de H_{min} y al emplearla concatenada con el sistema. En ambos casos obtenemos la señal de error que habíamos calculado en el

apartado anterior, solo que esta vez sobre 1 que es el valor que debería tomar el filtro todo. Por lo que concluimos que el diseño es correcto y el sistema se comporta como debería, pese a los errores en amplitud que no hemos conseguido compensar.



Adicionalmente hemos usado el Error Cuadrático medio de las muestras de los espectros para intentar aproximar el error cometido que se puede ver en las gráficas de arriba. Para ello hemos implementado la función `calc_ecm` que devuelve la raíz cuadrada del error cuadrático medio cometido entre 2 señales.

ECM Hfiltros(w)	0.0344
-----------------	--------

5.- Simulación del Sistema usando Señales de Audio

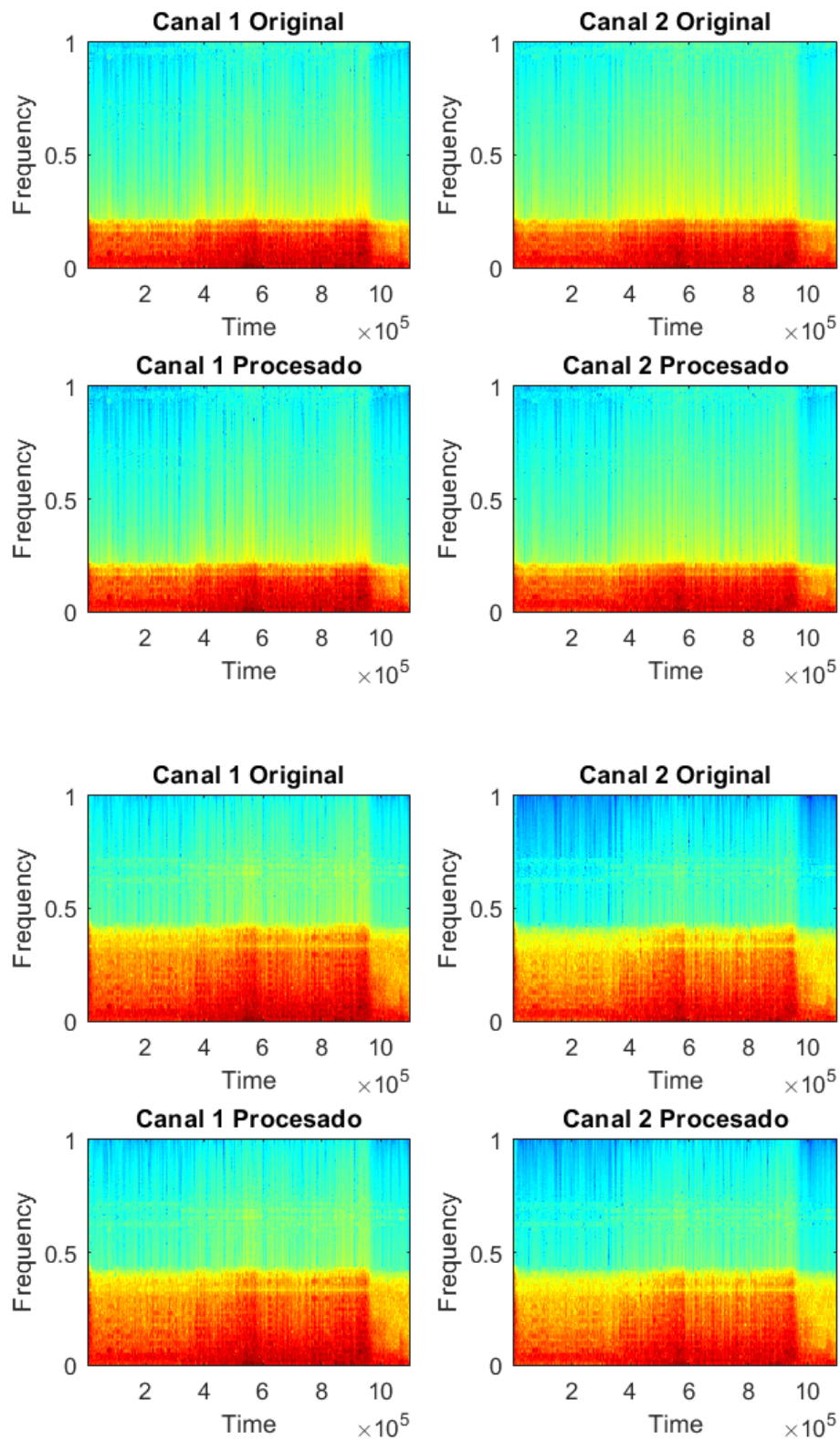
Vamos a ver si el sistema con una pista de audio real, para ello disponemos de 3 pistas de audio muestreadas a diferente frecuencia. Lo primero es cargarlas dentro de Matlab como vectores, con la función `audioread`. Como nuestros filtros están diseñados

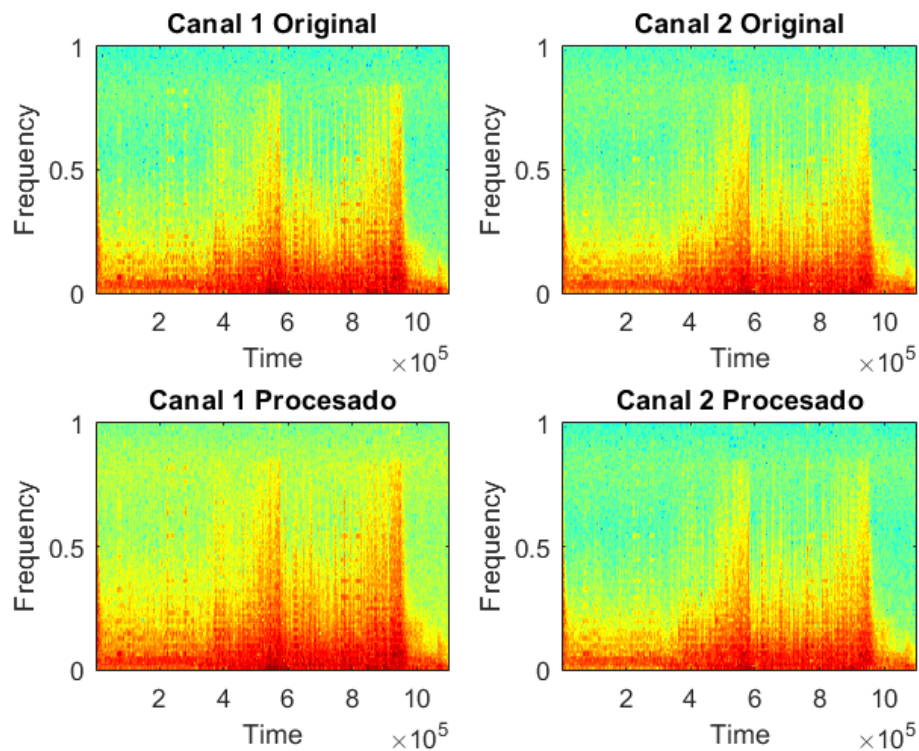
para una frecuencia de muestro de 48Khz, es necesario hacer un remuestreo de las distintas canciones. Para ello emplearemos las técnicas aprendidas en el tema 2 de la asignatura, haciendo un cambio de frecuencia de muestreo fraccionario. A la hora de calcular los coeficientes P y Q de esta técnica, usamos la función *rat* de Matlab, pasándole como argumento, el cociente entre la frecuencia de muestreo requerida y la frecuencia a la que está muestreada la canción. Para hacer el cambio de frecuencia de muestreo propiamente dicho, usaremos la función *resample* de Matlab. Los parámetros serán los P y Q calculados anteriormente así como un factor n proporcionado en el enunciado. Además requerimos del factor β de la ventana de Kaiser. Para procesar audio de calidad, es necesario que el filtro antisolapamiento presente una atenuación de 100 Db, para ese rango de atenuación α , el factor β se calcula con la siguiente expresión:

$$\beta = 0.1102 * (\alpha (dB) - 8.7)$$

Una vez remuestreadas nuestras señales de prueba, procedemos a filtrarlas por nuestro sistema, para ello, hacemos uso de la función de Matlab *filter*, usando como argumento los coeficientes a_n y b_n de los distintos filtros. Para modular el código, hemos creado la función *procesar_senal*, que se encarga de filtrar un vector X por el sistema distorsionador y el sistema compensador.

Finalmente generamos espectrogramas de cada uno de los canales de las 3 señales para comparar la señal original remuestreada con la obtenida después del procesado. Lo más significativo es ver como el sistema distorsionador, no introduce componentes espectrales fuera de las frecuencias de muestreo. En las señales de 8 y 16 khz, el sistema distorsionador, no introduce apenas componentes en frecuencias superiores a 4 y 8 Khz respectivamente, por lo que en ese sentido nuestro sistema funciona correctamente. En general, los espectrogramas son similares, por lo que podemos concluir que el ecualizador funciona





Adicionalmente al igual que con H_{filtros} , hemos calculado el error cuadrático medio de las distintas señales para intentar cuantificar el error cometido respecto a la señal original. El procedimiento es análogo al empleado con H_{filtros} .

Señal	Ecm Canal 1 (Sin Compensar)	Ecm Canal 1	Ecm Canal 2 (Sin Compensar)	Ecm Canal 2
Señal muestreada a 8khz	0,0238	0,0029	0,0254	0,0028
Señal muestreada a 16khz	0,0238	0,0034	0.0254	0,0031
Señal muestreada a <u>41khz</u>	0.0239	0,0036	0.0254	0,0031

Los Errores obtenidos, creemos que son lo suficientemente pequeños como ver que el ecualizador funciona adecuadamente.

ANEXOS

Bibliografía

- Discrete-Time Signal Processing – Alan V. Oppenheim
- https://es.wikipedia.org/wiki/Transformaci%C3%B3n_bilineal
- <https://es.wikipedia.org/wiki/Distorsi%C3%B3n>

Funciones matlab

- calculafrecuencias.m Función auxiliar para calcular equivalencias entre frecuencias continuas y discretas
- calc_ecm.m Función que calcula el error ECM dadas dos señales
- concatfilters.m Funcion que devuelve la respuesta al impulso en el dominio de la frecuencia de la concatenación de una serie de filtros
- muestralmitar.m Funcion auxiliary usada para representar Hfiltros durante el proceso de diseño
- muestraSistema.m
- pasoalto.m Función de calculo de los coeficientes del Paso Alto
- pasobajo.m Función de calculo de los coeficientes del Paso Bajo
- pasobanda.m Función de calculo de los coeficientes del Paso Banda
- print_espectograma.m Funcion auxiliar para imprimir los espectrogramas
- procesar_senal.m función para procesar una señal por el sistema distorsionador y el conjunto de filtros.