

INSTITUTO TECNOLÓGICO AUTÓNOMO DE MÉXICO  
LABORATORIO: Procesamiento Digital De Señales

## Proyecto 1

### Ecualizador de 7 Bandas

---

Integrantes:

Mauricio Martinez Riede – 176762

Humberto Martinez Barron y Robles – 166056

Instructor:

Marco Hernandez

Fecha de elaboración:

26/03/19

## 1 Introducción

El procesamiento de audio es una extensión importante y divertida del procesamiento de señales. Una aplicación particularmente interesante es el control del sonido de una canción por medio de un ecualizador que es capaz de amplificar o atenuar bandas de la señal original (la canción). Antes de que existieran programas poderosos que permitieran el diseño fácil y rápido de filtros como MatLab y Simulink, los filtros se debían diseñar a mano y se debían obtener los componentes necesarios para lograr que los filtros tuvieran las frecuencias de corte deseadas, sin mencionar que el alambrado del circuito representaba en sí una limitación más para lograr armar ecualizadores propios. Sin embargo, gracias a estas tecnologías, los ingenieros que diseñan equipos de audio ahora lo pueden hacer fácilmente con herramientas como filterDesigner y filterBuilder, o, si lo prefieren, pueden programarlas directamente en código de MatLab gracias a las bibliotecas que el programa tiene incluidas para lograr un procesamiento rápido y simple.

Lo interesante de proyectos como el que se presenta a continuación es que permite crear un ecualizador con gran sonido y buena atenuación o ganancia sin perder tiempo en alambrear circuitos para prototipar. Utilizar estas herramientas es tan fácil como dar unos cuantos clicks y especificar unos cuantos datos. Como una ventaja adicional, el diseño de diagramas de bloques en Simulink permite que la traducción a código de tipo VHDL sea suficientemente fluida para utilizar Simulink directamente en la impresión y en el diseño de circuitos integrados. Además, resulta bastante divertido visualizar los cambios en la señal filtrada cuando las bandas se atenúan o se amplifican.

Para este proyecto del laboratorio de Procesamiento Digital de Señales se nos dio la tarea de crear un equalizador para archivos de audio de 7 bandas, que van desde los 90Hz hasta los 14000Hz. Los filtros pueden ser diseñados en Simulink con la ayuda de FilterBuilder, una funcionalidad en Matlab que permite al usuario diseñar sus filtros simplemente sabiendo los parámetros a utilizar para cada uno. FilterBuilder además tiene una función para implementar estos filtros en Simulink. Además, los alumnos deberán diseñar un GUI (Graphic User Interface) para poder controlar cada uno de los parámetros del filtro mientras un archivo de audio se corre en el programa. La suma de todos los filtros deberá dar la señal original de audio.

## 2 Marco Teórico

Los tipos de filtros más comunes utilizados en el procesamiento de audio son: Chebyshev tipos I y II, Butterworth y elíptico. Las diferencias principales entre ellos son las bandas de paso, bandas de rechazo, rizo de la banda de paso y de rechazo y atenuación de la banda de rechazo.

## 2.1 Filtros Butterworth

En este filtro, la respuesta del paso de banda es “máximamente plana” porque este filtro no induce un rizo en el paso de banda pero da una atenuación relativamente aguda de la banda de rechazo (*roll-off*). Sin embargo, el *roll-off* de las bandas de paso es más suave que el resto de los filtros, y la falta de rizo da una nitidez especial a la señal. Los filtros Butterworth son una buena aproximación de los filtros ideales cuando se usan como pasabandas.

## 2.2 Filtros Chebyshev

Los filtros de Chebyshev ofrecen un rizo en la banda de paso o de rechazo, dependiendo de si se escoge el tipo I o el tipo II. Sin embargo, a cambio de este rizo, puede lograr un *roll-off* más agudo a partir de la frecuencia de corte que los filtros Butterworth. Por eso se utilizan estos filtros cuando se requiere que la caída sea aguda. Cabe señalar que, entre más aguda sea la caída, más grande será el rizo generado.

## 2.3 Filtros elípticos

Este diseño de filtro tiene el *roll-off* más agudo que existe. Sin embargo, presenta rizados tanto en la banda de paso como en la banda de rechazo. Por lo tanto, la señal original es distorsionada en todo su espectro. Además, se puede diseñar el filtro elíptico para lograr atenuaciones importantes en algunas frecuencias de la banda de rechazo.

## 3 Desarrollo

Ahora, se expone el procedimiento que se siguió para escoger el tipo de filtros que se utilizarían, así como la implementación en Simulink y el diseño de la interfaz gráfica. También se menciona rápidamente la forma de utilizar archivos de audio (canciones) para probar los filtros.

### 3.1 Diseño de los filtros

La primer pregunta que se debía responder era: ¿FIR o IIR? Vale la pena considerar que los filtros IIR tienen fase no-lineal. Esto significa algo de distorsión de la señal. Por otro lado, los filtros FIR son de mayor orden que los de IIR con las mismas especificaciones (porque son filtros de mayor orden). Esto significa que la computadora tendrá que hacer

más cálculos, lo que significa que el procedimiento será más lento. Esto podría limitar la cantidad de pruebas que se pueden hacer. Por otro lado, la fase no lineal de los filtros IIR podría representar un inconveniente para el ecualizador, además de que sería necesario garantizar que el filtro es estable.

Sin embargo, gracias a la implementación de MatLab, se puede abstraer la mayor parte del diseño de los filtros analógicos y simplemente dar los parámetros requeridos. Se escogieron los filtros IIR para utilizar el ecualizador por la facilidad con la que se pueden diseñar y porque son computacionalmente más “baratos”.

Ahora bien, para escoger si se utilizaría un filtro Butterworth, Chebyshev o elíptico, se comenzó por diseñar los filtros de forma tal que fueran visibles las bandas de paso en el analizador de espectros. Así, con las frecuencias sugeridas en las especificaciones del proyecto para un ecualizador de siete bandas, se escogieron los filtros de las bandas 1 y 2 como elípticos (pues las frecuencias estaban muy cerca, y era difícil notar las siete bandas con otros tipos de filtros porque tendían a intersectarse y parecer una sola banda). Para las bandas 3, 4 y 5 se utilizó el tipo Chebyshev con anchos de banda grandes, pues a pesar de que se consideró importante un filtro con *roll-off* agudo, no se necesitaba exagerarlo. Además, un ancho de banda mayor puede ayudar a corregir la fase no lineal del filtro Chebyshev. Para las bandas 5, 6 y 7 se utilizaron filtros Butterworth, pues no era necesario atenuar con demasiado *roll-off*. Una vez diseñado este circuito, se probó en Simulink con una canción como entrada (ver las secciones 3.2 y 3.3) y, desafortunadamente, ¡resultó que se escuchaba horrible! La razón por esto es que se había dado prioridad a la atenuación de las bandas de rechazo porque se estaba tratando de lograr visualizar los filtros. Así, fueron necesarios varios cambios.

Después de pensar que, inicialmente (es decir, con una ganancia de uno para todas las bandas), la señal filtrada debería de escucharse casi idéntica a la señal original, se llegó a la conclusión de que sería mejor utilizar un tipo de filtro que distorsione menos la señal de entrada y que no tenga un *roll-off* tan abrupto. Con esto en mente, se cambiaron los tipos de filtro a Butterworth. Sin embargo, aún se tenían bandas que rechazaban frecuencias importantes para el audio. Fue así que se tomó la decisión de cambiar las frecuencias centrales y de corte sugeridas para lograr una aproximación más suave a la señal original con todas las ganancias puestas con valor 1.

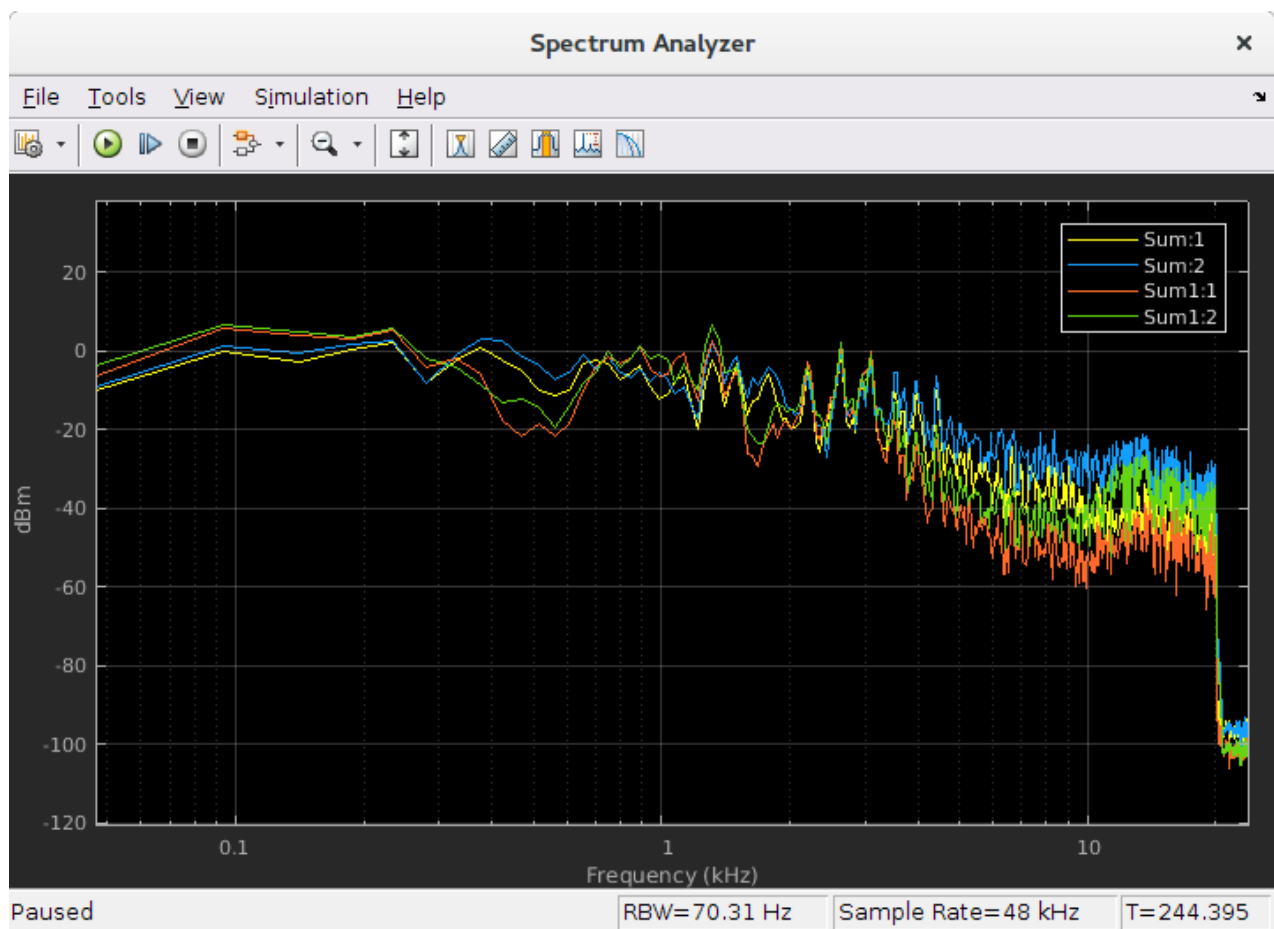
Así, se escogió utilizar las siguientes frecuencias centrales con sus respectivos anchos de banda mostrados en la tabla 1.

**Tabla 1.** Frecuencias centrales y anchos de banda utilizados para cada banda del ecualizador.

	<b>Frecuencia central (Hz)</b>	<b>Ancho de banda (Hz)</b>
<b>Banda 1</b>	130	N/A
<b>Banda 2</b>	240	180-300

<b>Banda 3</b>	500	350-750
<b>Banda 4</b>	1200	800-1600
<b>Banda 5</b>	2400	1800-3000
<b>Banda 6</b>	7000	4000-10000
<b>Banda 7</b>	10000	N/A

El resultado obtenido de esto fue más que satisfactorio. Como se puede observar en la figura 1, la señal de slaida es prácticamente idéntica a la señal original. Una vez que se logró esto, el resto fue cuestión de conexiones con la interfaz gráfica y la construcción de estos filtros en Simulink.



**Figura 1.** Análisis del espectro resultante del ecualizador (verde, rojo) vs el espectro de la señal original (azul, amarillo).

### 3.2 Implementación de los filtros en Simulink

Una vez terminado el diseño, fue necesario implementar cada filtro. Para esto, se utilizó la herramienta filterBuilder en MatLab (con parámetros 'lp' para diseño de pasabajas, 'bp' para diseño de pasabandas y 'hp' para diseño de pasaaltas). Se proporcionaron los

parámetros mostrados en la tabla 1 con especificaciones de frecuencias de media potencia (3dB) y tipo de filtro Butterworth con un muestreo de 44100 Hz. Después de oprimir el botón *Apply*, en la pestaña de código se generó el bloque de Simulink, y lo único necesario fue repetir este procedimiento con todos los parámetros y conectar todos los bloques con un sumador al analizador de espectros. Para comparar con la señal original, se conectó ésta también al analizador (ver figura 2).

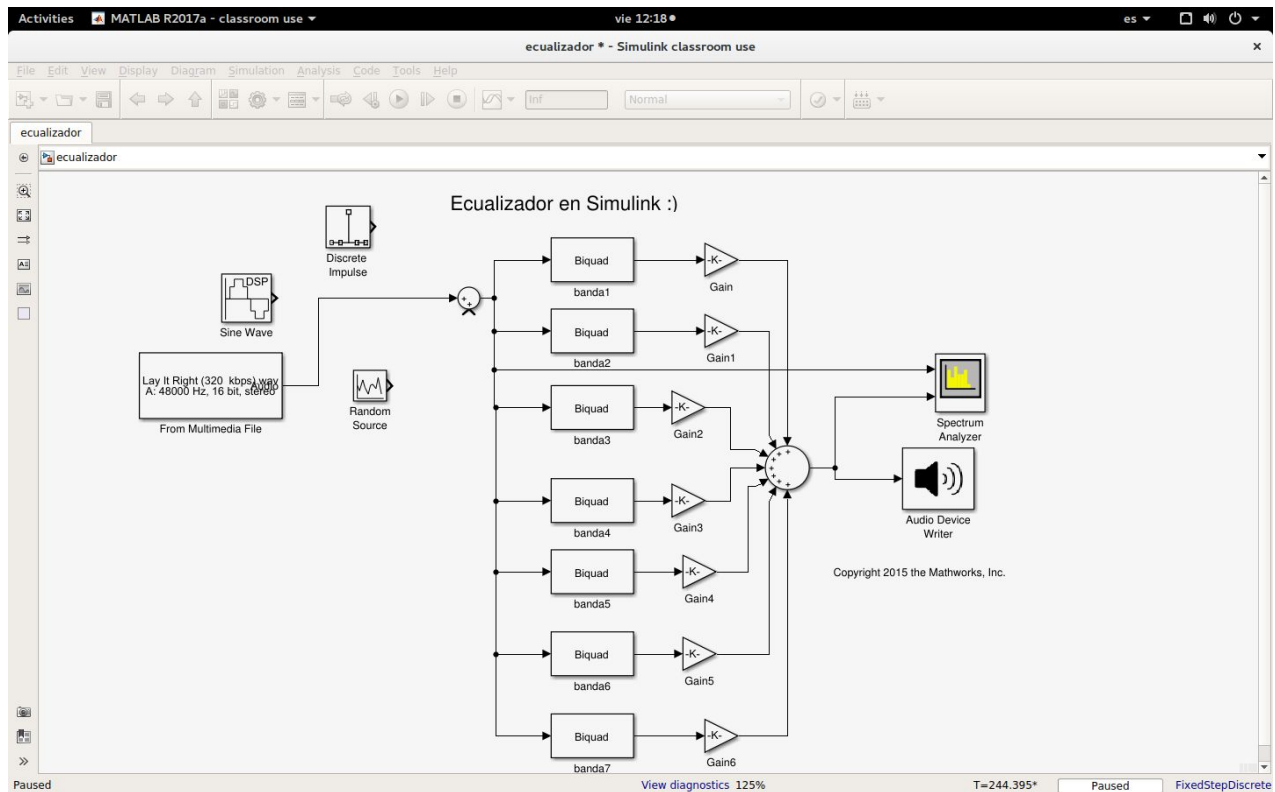
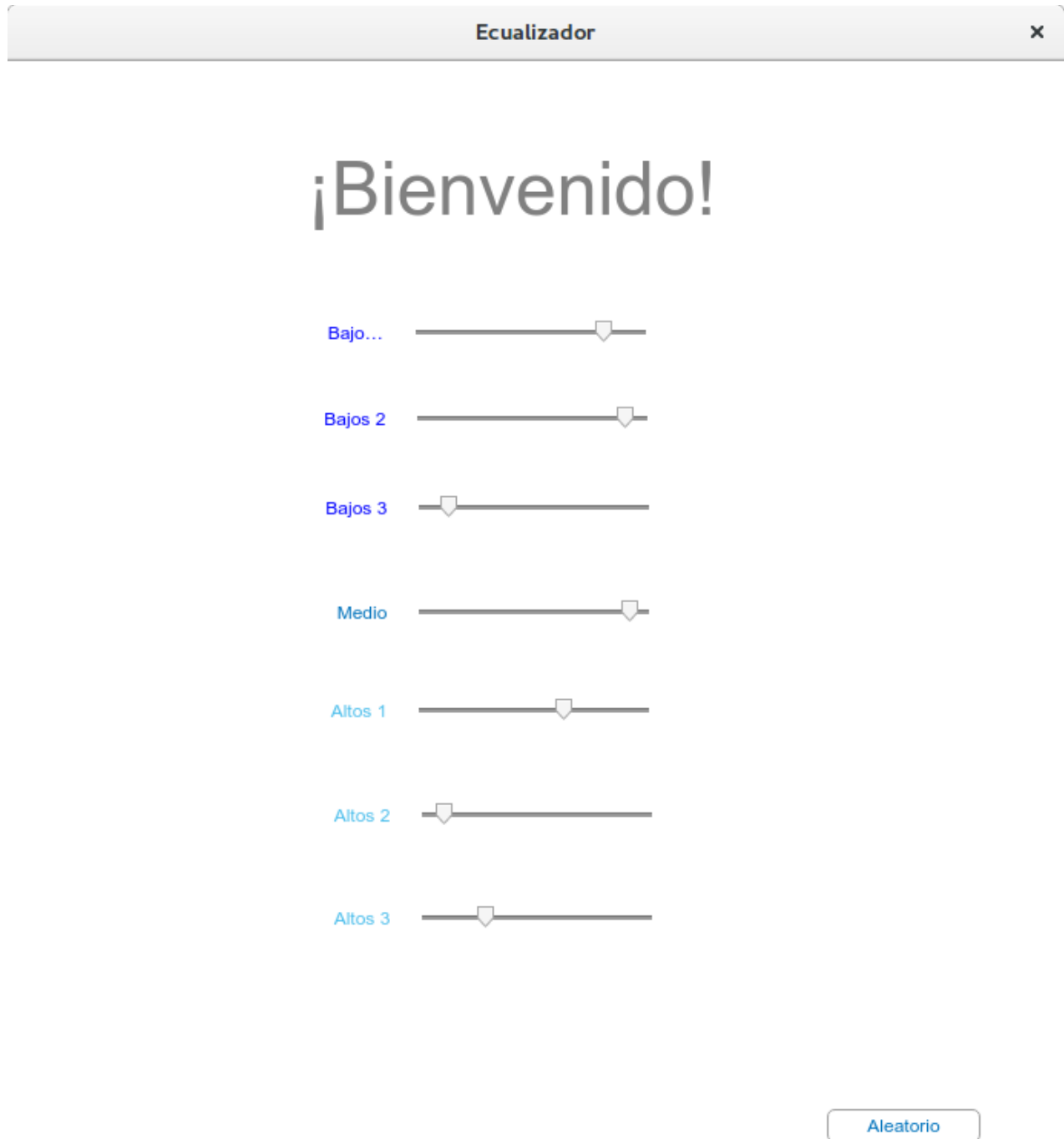


Figura 2. Ecualizador armado con ganancias aleatorias y una entrada de audio.

### 3.3 Diseño de la interfaz gráfica

Para la interfaz gráfica de usuario, lo único necesario fue crear una nueva aplicación, arrastrar los elementos visuales deseados, darle el estilo que se buscaba y programar la funcionalidad de los sliders. Para esto, se crearon funciones de *callback* con las que se especificaba el valor de las ganancias del archivo *ecualizador.slx*. La función *set\_param* de MatLab ya implementa esto cuando el archivo adecuado está corriendo y se proporciona el nombre del componente y del parámetro a modificar, así como su valor en formato *string*.

Además, se buscó que la interfaz tenga un diseño simplista y, aunque sea inútil como aplicación de ecualizadores industriales, se agregó un botón de aleatorio para probar el ecualizador, poniendo todas las ganancias en un valor arbitrario para escuchar el cambio en la señal filtrada (ver figura 3).



**Figura 3.** Diseño de la interfaz gráfica de usuario para controlar las ganancias cada banda del ecualizador.

### 3.4 Uso del archivo de audio

En Simulink se puede encontrar un convertidor de archivos de audio a un archivo que el simulink puede correr. En el laboratorio MAC puede correr archivos de wav hasta mp3. En Linux, por otro lado, no es posible utilizar mp3, por lo que la conversión a wav es necesaria. La diferencia entre wav y mp3 es que wav puede cubrir todo el rango de frecuencias que el oído humano puede escuchar, mientras que el mp3 es un archivo comprimido y hay pérdida de calidad. Aun así, si un mp3 corre a un bit rate de 320 Kbps (kilo bytes por segundo) el oído humano no debe poder escuchar la diferencia.

## **4 Conclusiones**

La creación de los filtros tomó varios intentos, ya que probamos diferentes tipos de filtros y diferentes parámetros para que el ecualizador funcionara de manera correcta. Aunque se nos dieron valores de frecuencia para cada filtro, modificamos un poco estos valores de manera que la suma de los filtros diera la señal original del audio. Afortunadamente nuestros filtros y nuestra interfaz gráfica funcionan de manera correcta y el ecualizador quedó completo. El proyecto además dejó aprendizajes acerca del tipo de filtros que conviene utilizar, las diferencias entre los órdenes de los mismos y las posibilidades de tipos de respuestas al impulso que se pueden seleccionar. Lo más interesante fue ver una aplicación del procesamiento digital de señales que tiene que ver con la forma en la que logra que escuchemos música. Esta aplicación práctica hace que la materia se vuelva aún más interesante, pues da una idea de la variedad de cosas que se pueden lograr con conocimientos como los que se están adquiriendo en el curso.