



LIBRERÍA DE EFECTOS DIGITALES DE AUDIO

Índice

Introducción	3
Descripción	3
Esquema de la aplicación	4
Documentación	5
Requisitos del sistema	5
Características de la aplicación	6
Importación de archivos	6
Frecuencia de muestreo	6
Espectro de señal	6
Ejecución de efectos	7
Paso de salida a variables	7
Interfaz gráfica	8
Index	8
Efecto	8
LFO	9
Generador de señal	9
Automatización de parámetros	10
Modulador de señal	10
Librería de archivos de la interfaz	11
Plantilla	12
Parámetros del efecto	12

Oscilador de baja frecuencia	13
Generador de señal	13
Automatización de parámetros	13
Modulador	13
Tipos de osciladores	13
Señal sinusoidal	14
Señal triangular	14
Señal diente de sierra ascendente	14
Señal diente de sierra descendente	15
Señal cuadrada	15
Ruido blanco gaussiano	15
Señal externa	15
Actualización de parámetros en el efecto	16
Gráficos	17
Gráficos de entrada y salida	17
Señales temporales	17
Espectros	17
Gráfico del efecto	17
Comparación entre espectros de entrada y salida	17
Efectos disponibles	18
Clasificación de los efectos	18
Autowah	19
Filtro	19
Gráfico del efecto	20
Delay	21
Modulación de duración	22
Flanger	22
Chorus	22
Vibrato	22
Distortion	23
Overdrive	24
Panning	25

Introducción

En la era actual, donde la digitalización se ha adueñado a nivel global y sin disputa del espacio que anteriormente ocupaba los sistemas analógicos, quizás sea en la música y el tratamiento de audio el sector donde aún se mantiene el uso de algunos sistemas analógicos, probablemente debido a la subjetividad del oyente que se ha acostumbrado con nostalgia al característico sonido de algunos de estos sistemas. Sin embargo, es probable que con el paso de tiempo este campo quede sucumbido casi en su totalidad a la digitalización, como se puede observar con el auge de la música electrónica y el uso incluso en tiempo real de las *Digital Audio Workstations* para conseguir efectos que salvando colores del sonido, son objetivamente más baratos, versátiles y eficientes.

Además, el rápido crecimiento del potencial computacional hace que crezca a la par el potencial de procesamiento de audio y la complejidad de este tipo de efectos. La gran variedad exige al gremio conceptos cada vez más avanzados de señales y filtros digitales, y por tanto es un campo laboral interesante para ingenieros afines al campo de las señales digitales, puesto que se demandan mayores conocimientos técnicos.

Aún así, no existen diferencias teóricas en los conceptos de los efectos, aunque sí son notables las diferencias en su implementación respecto a su desarrollo analógico o digital. Incluso existen algunos cuya implementación analógica es en la práctica imposible, pero que el procesado digital lo hace realizable.

Descripción

El proyecto descrito en esta documentación es el desarrollo de una librería de efectos digitales de audio en lenguaje nativo de Matlab con procesamiento a tiempo no real. Incluye una interfaz de usuario sencilla y autoexplicativa, y ofrece un control libre de los parámetros del efecto, elección y visualización del audio de entrada, reproducción y visualización del audio de salida, y representación característica del procesamiento que se está realizando.

El objetivo principal de la librería es permitir la experimentación con diversos parámetros y entradas de audio facilitando, de esta forma, la comprensión de los diferentes procesamientos que se están realizando. La documentación del proyecto incluye unas instrucciones y una plantilla con el objetivo de que se puedan añadir en más programas de efectos.

Debido a lo anterior, se ha pretendido disponer también de una interfaz sencilla, clara y estandarizada para todos los efectos, con un grado de libertad de los parámetros total, teniendo en cuenta de que el perfil del usuario final incluye un nivel avanzado de conocimientos relacionados. Aunque ésta documentación dispone de algunas guías de configuración de los parámetros, es el usuario el responsable de la elección acertada de los valores.

Con motivo de agilizar los procesamientos, y con esto aumentar la cantidad de experimentación que el usuario puede hacer, se recomienda habilitar archivos de audio de corta duración (en torno a 10 segundos) y con una frecuencia de muestreo estándar de 44100Hz .

Finalmente, la librería y la documentación se ofrecerá libremente a disposición de cualquier usuario en forma de software libre, a través de la siguiente dirección:

<https://github.com/andresgonzalezfornell/TFG/archive/master.zip>

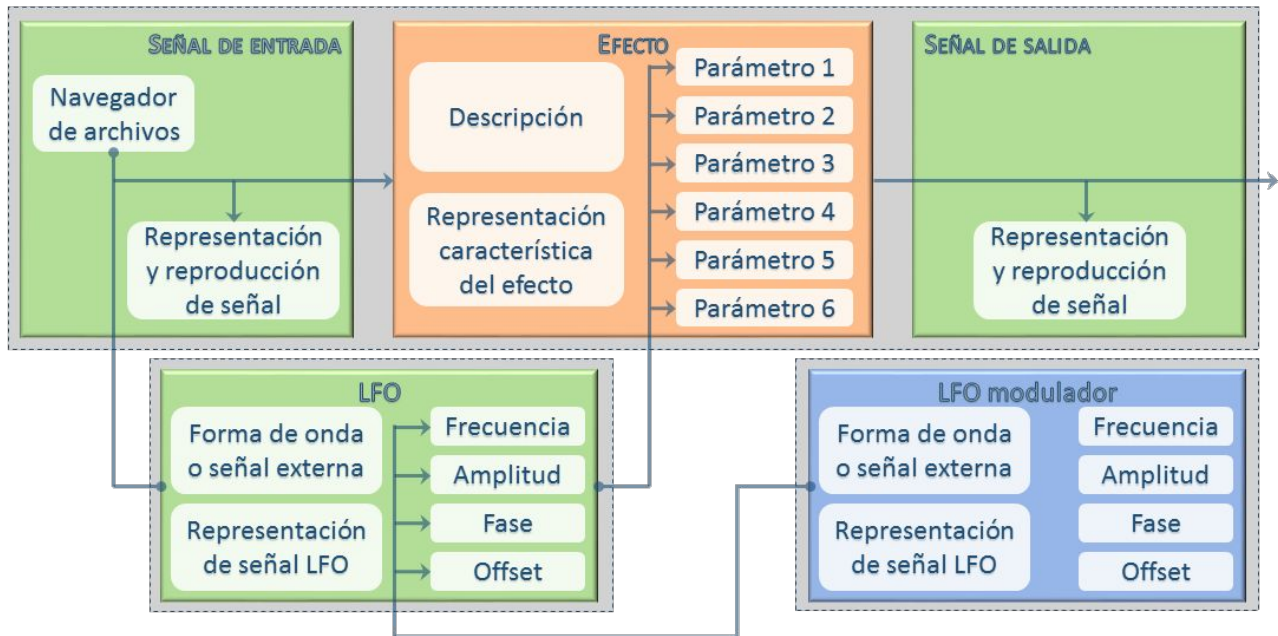
Los derechos de autoría quedan reservados, pero se permite el uso total y el desarrollo de mejoras u otras versiones a partir del software ofrecido por parte de terceros, aunque no se permitirá la explotación lucrativa de la aplicación.



Librería de efectos digitales de audio
CC by-nc-sa 2.5 es Andrés González Fornell

Esquema de la aplicación

El procesamiento de las señales sigue el esquema que se muestra a continuación. En los apartados siguientes de la documentación se profundizarán en cada uno de los módulos.



Los tres módulos principales se corresponden con las tres ventanas de control de cada efecto. Dos de ellos (*LFO* y *LFO modulador*) son comunes y compatibles para todos los efectos, y además se pueden aplicar de forma independiente para configurar el módulo principal, de tal forma que cada uno de los parámetros además de la misma señal de entrada pueden ser controlados por un LFO diferente que a su vez esté modulado en frecuencia, amplitud y fase por un otro oscilador LFO diferente para cada LFO y cada parámetro modulado de éste. En total se puede dar el caso máximo en que se hayan configurado

$$(6 \text{ parámetros} + 1 \text{ generador de señal}) \cdot (1 \text{ LFO principal} + 3 \text{ LFO moduladores}) = 28 \text{ LFOs}$$

todo ello a partir del potencial que ofrece la implementación de tan sólo dos módulos de osciladores diferentes (el LFO principal y el LFO modulador) que se dan juego entre ellos mismos.

Documentación

En esta documentación se tratarán el conjunto de efectos desarrollados en lenguaje Matlab, propuestos con un nivel de detalle suficiente para entender la idea principal de éstos e incluso ser capaz de realizar pequeñas modificaciones al comportamiento de los efectos. Sin embargo, no se pretenderá hacer entender al detalle el funcionamiento de la interfaz gráfica ni de los módulos de control de los efectos, aunque en la sección [Características de la aplicación](#) se dará una visión suficiente para el desarrollo de nuevos efectos de la librería. Todo el código se ha probado y desarrollado en la versión de Matlab R2015a, por lo que no se asegura el correcto funcionamiento bajo una versión anterior, aunque todas las dependencias usadas se encuentran a partir de la versión R2012b.

Debido a la continua referencia de los conceptos con el libro *DAFX: Digital Audio Effects*¹ durante el desarrollo de la librería, se ha indicado en la documentación de cada efecto la referencia específica de cada uno de los efectos en el libro. Por ello se aconseja al usuario que utilice el libro como documentación adicional para obtener una experiencia de aprendizaje máxima.

Algunos apartados de la documentación aparecen sombreados de colores. A continuación se describe el significado de cada color.

Resumen del efecto. Este apartado incluye:

- Descripción Breve descripción del efecto.
- Tipo Ubicación dentro de la clasificación de los efectos.
- Parámetros Parámetros de entrada que pedirá el programa y aparecerá en la interfaz.
- Referencia Capítulo del libro *DAFX: Digital Audio Effects* donde se aborda el efecto.

Código. Descripción en lenguaje lógico de Matlab, no necesariamente ejecutable ni parte del código real del programa.

Comando. Escritura en la ventana de comandos de Matlab.

Preset. Configuración de parámetros para obtener un determinado sonido o efecto.

Requisitos del sistema

Sistema operativo	Linux Ubuntu 14.04 LTS o superior Mac OS X 10.9.5 (Mavericks) o superior Windows XP o superior
Mathworks Matlab	Desarrollado en R2015a (aunque funciona a partir de la versión R2012b no se garantiza para versiones anteriores)
Plug-ins de Matlab	Signal Processing Toolbox
Archivos de entrada	Formato <i>.mp3</i> o <i>.wav</i> , a cualquier frecuencia de muestreo ²

La librería está garantizada y probada bajo los requerimientos indicados. Aunque es probable que funcione bajo circunstancias diferentes, es posible que su funcionamiento no sea completo y por tanto no es recomendable.

¹ ISBN: 978-0-470-97967-9, varios autores, libro editado por Udo Zolzer.

² La librería ha sido desarrollada para ser óptima a una frecuencia de muestreo de 44100Hz.

Características de la aplicación

A continuación se describirán a grandes rasgos el funcionamiento de la aplicación para el uso avanzado de ésta, su mantenimiento y la implementación de mejoras y nuevos efectos.

Importación de archivos

Las funciones `importdata`, para la importación de archivos `.wav`, y `audioread`, para archivos `.mp3`, devuelven una variable array de $n \text{ muestras} \times 2 \text{ canales}$ en el caso de estéreo, o $n \text{ muestras} \times 1 \text{ canal}$ para archivos de audio mono. Para la librería, la señal siempre será estérea, por tanto, en caso de que solo se disponga un sólo canal se duplicará para su procesamiento.

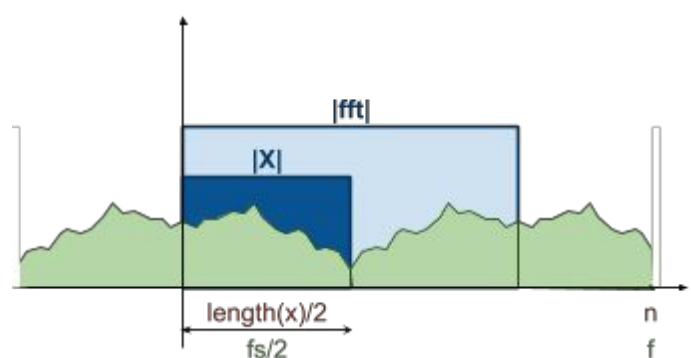
```
if strcmp(format,'wav')           % Si el formato es wav
    file = importdata(strcat(path,'/',filename));
elseif strcmp(format,'mp3')       % Si el formato es mp3
    [file.data, file.fs] = audioread(strcat(path,'/',filename));
end
x = file.data;
fs = file.fs;
if length(x(1,:)) == 1           % Duplicación del canal si el archivo es mono
    x(:,2) = x(1,:);
end
```

Frecuencia de muestreo

Además de los dos canales de señal de audio se obtiene la frecuencia de muestreo del archivo. Ésta es necesaria tanto para la reproducción de las señales como para el cálculo de su espectro y su procesamiento, por ejemplo, en la estimación del número de muestras necesarios para realizar el retardo en un *delay*. Siguiendo el teorema de Nyquist, el ancho de banda máximo de la señal, y a partir de esto su calidad, coincidirá con la mitad de la frecuencia de muestreo. Puesto que el oído humano abarca a lo sumo un rango de frecuencias desde 20 Hz hasta 20000 Hz , la frecuencia de muestreo (muy utilizada) de 44100 Hz es más que suficiente para ofrecer la mejor calidad, y la librería ha sido optimizada para trabajar a esta frecuencia.

Espectro de señal

Para una señal dada, al obtener su transformada FFT (Fast Fourier Transform) mediante la función `fft(x)` de Matlab, el resultado tiene de longitud igual al número de muestras de la señal. Puesto que en frecuencias, el espectro se encuentra duplicado simétricamente (la señal siempre será real) y en el eje positivo en el caso de la función `fft(x)` (tal como se muestra en la figura), habrá que tomar como espectro la primera mitad de las muestras obtenidas de esta función. Finalmente, es necesario escalar el eje de frecuencias de tal forma que el número de muestras total de `fft(x)`, que como se ha indicado es igual al de muestras de `x`, se corresponda con `fs`.



```
n = 1 : 1 : length(x);
t = n/fs;
f = 0+fs/length(n) : fs/length(n) : file.fs/2;
X = fft(x);
X = X(1:length(n)/2); % Recorte de la fft a la mitad
plot(f,X);
```

Ejecución de efectos

Para una ejecución simple teclee en la ventana de comandos de Matlab el nombre del efecto que desea lanzar. Ejemplo:

```
>> delay
```

Se ofrece también la posibilidad de iniciar la librería mediante una interfaz índice donde aparecen todos los efectos disponibles.

```
>> index
```

Paso de salida a variables

Es posible guardar como variables la señal de salida del procesamiento que se va a realizar. Para ello, a la hora de ejecutar el efecto deseado habría que igualar una variable nueva al efecto que se ejecutará. Esto se debe a que los programas de cada efecto están definidos como funciones de Matlab. Ejemplo:

```
>> y = delay;
```

La variable igualada se corresponderá con un array de longitud $N \times 5$, siendo N el número de muestras de la señal de salida. Las señales devueltas correspondientes se describen a continuación.

```
>> y(:,1)  Señal de salida del canal L
>> y(:,2)  Señal de salida del canal R
>> y(:,3)  Espectro de la señal de salida del canal L
>> y(:,4)  Espectro de la señal de salida del canal R
>> y(:,5)  Espectro de la señal media aritmética de ambos canales
```

Para obtener ayuda de como tratar el paso de salida a variables teclee en matlab el comando `help` seguido del programa. Ejemplo:

```
>> help delay
```

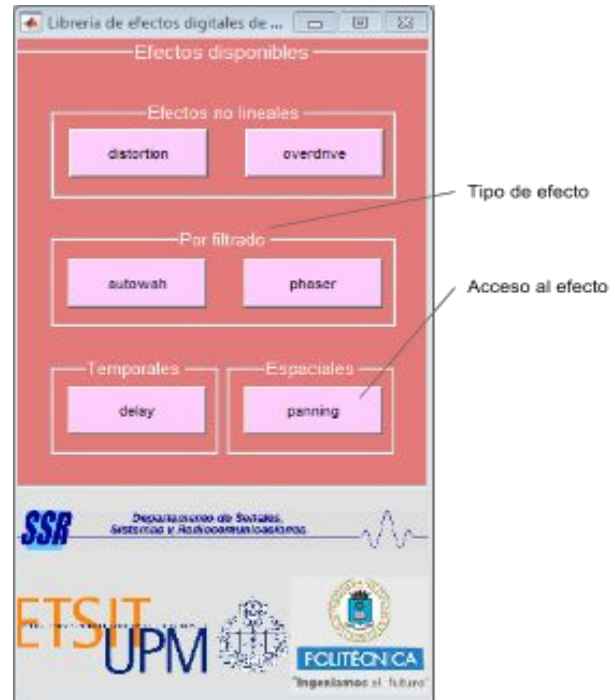

Interfaz gráfica

Index

Si se accede a la librería mediante el comando `index` se muestra una interfaz similar a la figura de la derecha. En ella se encuentran todos los efectos disponibles.

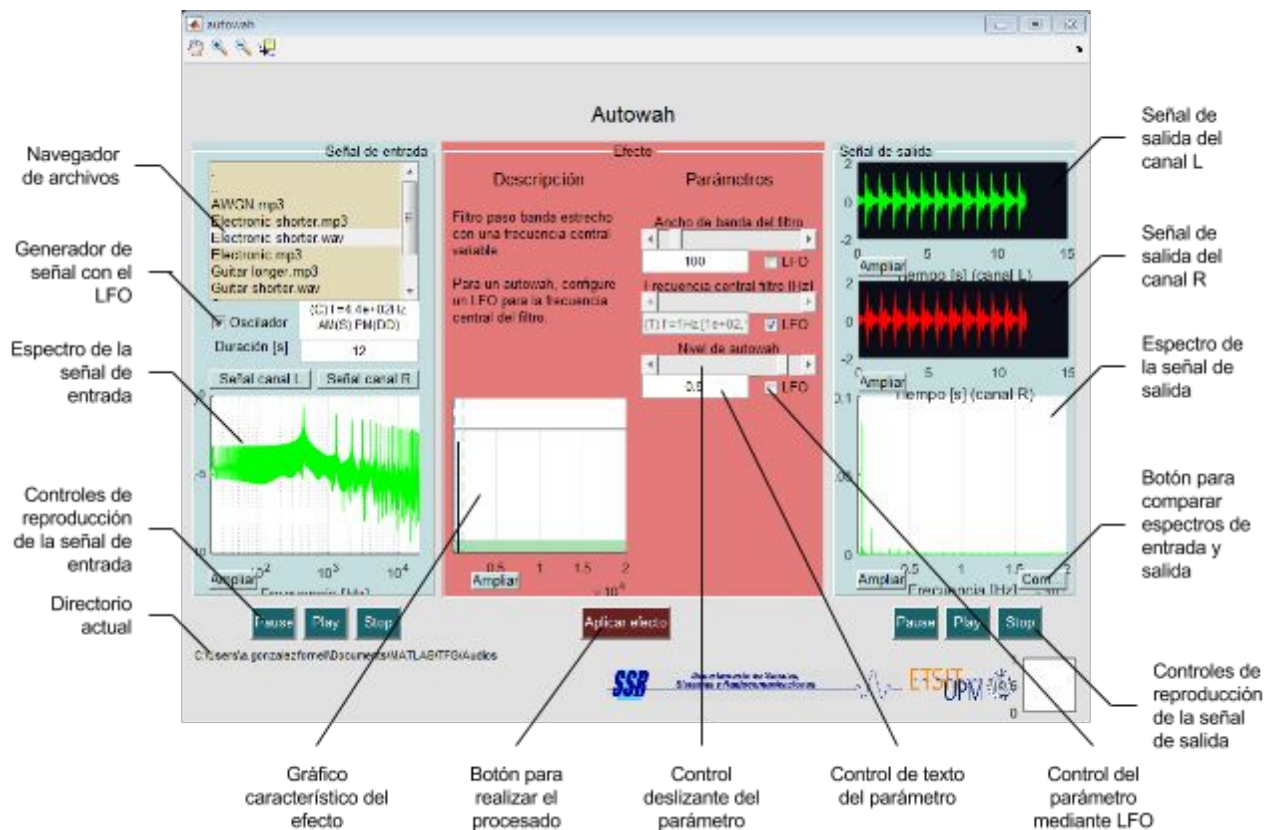
Para acceder a cualquiera de ellos simplemente se deberá pulsar en el efecto deseado. Los efectos se encuentran catalogados en grupos según el tipo de proceso que realizan.

La aplicación está diseñada para poder usar más de un efecto al mismo tiempo. Es por ello por lo que la ventana de índice no se cierra una vez seleccionado un efecto, permitiendo de este modo la apertura de varios efectos al mismo tiempo, lo que no se podría hacer mediante la ejecución de los efectos a través de la ventana de comandos.



Efecto

Todas las interfaces de los efectos siguen la misma estructura. Los controles se agrupan en tres paneles principales, correspondientes a los controles de señal de entrada, efecto y señal de salida.



La navegación de archivos toma como ruta por inicial *Audios*, situada en el directorio dónde se encuentra la aplicación. En caso de que no exista este directorio usará el mismo de la aplicación.

LFO

La interfaz del LFO es común para todos los efectos e independiente de para qué parámetros o generador de señal se aplique. En la práctica es una subaplicación que devuelve a la aplicación principal una señal que determina los valores que toma el parámetro configurado a lo largo del tiempo.

En los cuadros de información de parámetro/señal se mostrará la siguiente notación, en el caso de que esté manipulado por un LFO.

	Forma de onda	Modulación	Frecuencia	Valor máximo	Valor mínimo
(S)	Sinusoidal	AM	$f =$	[,]
(T)	Triangular	FM			
(DA)	Diente de sierra ascendente	PM			
(DD)	Diente de sierra descendente				
(C)	Cuadrada				
(N)	Ruido blanco gaussiano				
Externa	Señal externa				

valor máximo = offset + amplitud

valor mínimo = offset - amplitud

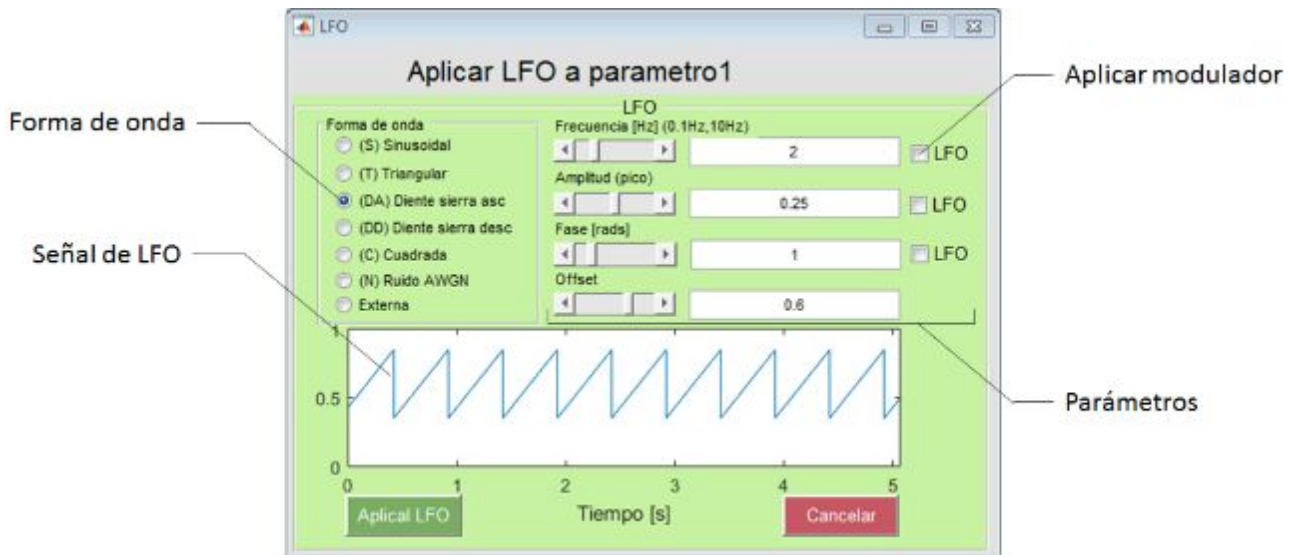
Generador de señal

Esta interfaz aparece al seleccionar el *checkbox* de “Oscilador” en el panel “Señal de entrada”. En este caso la duración de la señal es variable y está seleccionada por el usuario en el panel de entrada. Además, no se habilita el *offset* de la señal, puesto que la componente continua de señal no aporta absolutamente nada en audio y puede dañar los altavoces. Tampoco se habilita la selección de una señal externa porque resulta redundante a seleccionar directamente un archivo de audio en el panel de entrada.



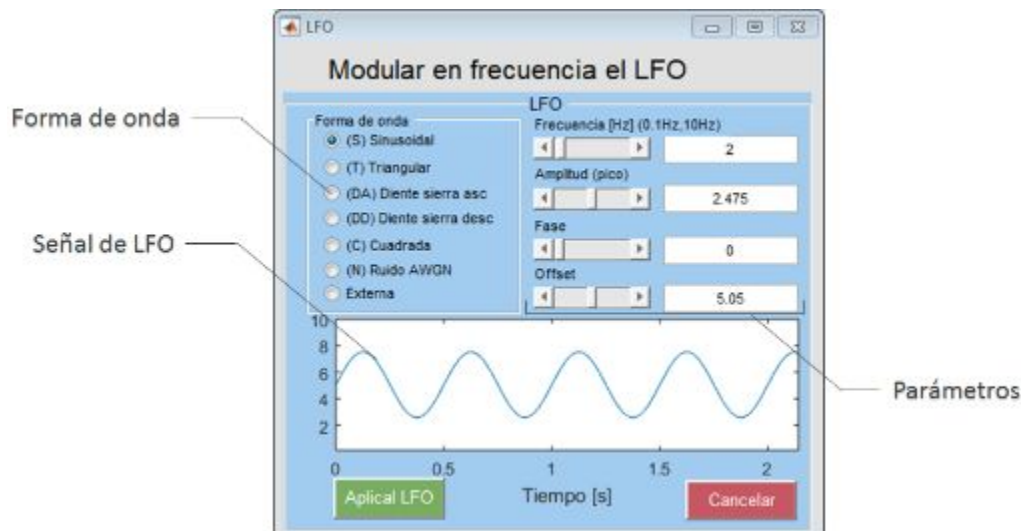
Automatización de parámetros

Esta interfaz aparece al seleccionar el *checkbox* de LFO de un parámetro. La duración de la señal es fija y coincide con la duración de la señal de entrada seleccionada. Es por esto por lo que el LFO no se habilita hasta que no se haya elegido un fichero de audio de entrada.



Modulador de señal

Esta interfaz aparece al seleccionar el *checkbox* de LFO de un parámetro dentro del mismo LFO de un parámetro o generador de señal. En este caso la interfaz es ligeramente diferente de las dos anteriores debido a que Matlab no permite abrir una interfaz desde ella misma. Además no se ha deseado ofrecer otra modulación para la misma señal moduladora para no complicar excesivamente la librería, aunque siempre se tiene la posibilidad de que el control sea totalmente libre a través de una señal externa.



En cualquiera de las interfaces de LFO descritas se aplican la configuración cuando se pulsa en “Aplicar LFO”, mientras que se cancela en “Cancelar”, como es de suponer. Sin embargo, si se cierra la ventana, el programa se detendrá en su plenitud y toda la configuración anterior se perderá.

Librería de archivos de la interfaz

Con motivo de optimizar los cambios en la interfaz gráfica, se han externalizado en la medida de lo posible algunas partes de códigos comunes para todos los efectos en archivos independientes. Todos estos archivos comienzan por el prefijo “z_”. A continuación se describen la funcionalidad de cada uno de ellos.

Archivo	Descripción
z_interfaz_OpeningFcn.m	Realiza todas las funciones al inicio del programa, tales como limpieza de variables y carga de elementos de la interfaz comunes a todos los efectos.
z_interfaz_OutputFcn.m	Devuelve, si se ha requerido, las salidas descritas anteriormente.
z_interfaz_entrada_lista_Callback.m	Controla la navegación entre los directorios así como la importación de los archivos como señal de entrada y su correspondiente representación.
z_entrada.m	Gestiona la graficación de la señal de entrada.
z_interfaz_salida.m	Representa la señal de salida en la interfaz además de habilitar su reproducción.
z_interfaz_limpieza.m	Realiza la limpieza de las variables de salida para aplicar de nuevo el efecto.
z_LFO.m z_LFO_GUI.m z_LFO_graf.m	Llevar a cabo el uso del LFO para el parámetro. <ul style="list-style-type: none"> • z_LFO.m se encarga del paso de parámetros y la gestión de <i>checkboxs</i>. • z_LFO_GUI.m se encarga de la interfaz gráfica de usuario de LFO. • z_LFO_graf.m se encarga de crear la señal deseada.
z_modulador.m z_modulador_GUI.m z_LFO_graf.m	Llevar a cabo el uso del modulador del LFO. <ul style="list-style-type: none"> • z_modulador.m se encarga del paso de parámetros y la gestión de <i>checkboxs</i>. • z_modulador_GUI.m se encarga de la interfaz gráfica de usuario del LFO que modula al LFO principal. • z_LFO_graf.m se encarga de crear la señal moduladora deseada.
z_entrada_espectro_open.m	Despliega en una nueva ventana el gráfico del espectrograma de la entrada.
z_entrada_L_open.m	Despliega en una nueva ventana el gráfico de entrada del canal L.
z_entrada_R_open.m	Despliega en una nueva ventana el gráfico de entrada del canal R.
z_salida_espectro_open.m	Despliega en una nueva ventana el gráfico del espectrograma de la salida.
z_salida_L_open.m	Despliega en una nueva ventana el gráfico de salida del canal L.
z_salida_R_open.m	Despliega en una nueva ventana el gráfico de salida del canal R.
z_comparar.m	Despliega en una nueva ventana el gráfico de los espectros de las señales de entrada y salida para su comparación.
z_graf_open.m	Despliega en una nueva ventana el gráfico característico del efecto.

Plantilla

Con motivo de facilitar el desarrollo de efectos y su entendimiento, la interfaz gráfica se ha estandarizado, la cual puede obtenerse a partir de una plantilla en el archivo *template.m* y *template.fig*. Esta plantilla incluye tres paneles principales.

Señal de entrada	Efecto	Señal de salida
Navegación y selección de archivo de entrada así como su representación temporal y espectral.	Breve descripción, parámetros y representación característica del efecto.	Representación temporal y espectral de la señal de salida y su comparativa espectral con la señal de entrada.

En el archivo *template.m* es necesario modificar el nombre de la ventana en el inspector de propiedades del elemento de fondo y todas las funciones *Callback* de los elementos de la GUI, renombrando *template* por el nombre del efecto a crear. Para facilitar la búsqueda se muestra a continuación una lista de los elementos que se deben modificar.

entrada_L_open entrada_R_open entrada_espectro_open pause_entrada play_entrada stop_entrada entrada_lista entrada_oscilador entrada_length	graf_open par_1 par_1_value par_2 par_2_value par_3 par_3_value par_4 par_4_value par_5 par_5_value par_6 par_6_value	salida_L_open salida_R_open salida_espectro_open comparar pause_salida play_salida stop_salida
--	---	--

En el archivo *template.m* es necesario buscar y reemplazar todas las coincidencias de las palabras *<efecto>*, *<Efecto>* y *<EFECTO>* por el correspondiente nombre del efecto (siguiendo la capitalización de las letras), así como renombrar el archivo por el mismo nombre. La palabra *<Descripcion>* se tiene que reemplazar por la descripción del efecto, la cual aparecerá en el panel central.

El efecto se implementa en la función *aplicar_Callback*, mientras que el control de los parámetros se realiza en las funciones *par_<#>_Callback* y *par_<#>_value_Callback* (siendo *<#>* el número del parámetro). Además, se deberán inicializar todos estos parámetros en la función *<efecto>_OpeningFcn*, donde también se seleccionarán los parámetros deseados estableciéndolos como visibles. Además, mediante las variables *handles.limite(<#>).Min* y *handles.limite(<#>).Max* se establecen los límites del parámetro para su tratamiento en el control del mismo así como para cuando se habilite el LFO. Finalmente, se ofrece la posibilidad de representar en el gráfico *handles.graf_open* cualquier tipo de representación característica del efecto.

Parámetros del efecto

La plantilla ofrece hasta seis parámetros, que se encuentran ocultos. El control de parámetros se realiza dentro de la función *<efecto>_OpeningFcn*. Para hacer visible un parámetro se usa la sentencia:

```
set(handles.par_1,'Visible','on','Min',handles.limite(1).Min,'Max',handles.limite(1).Max)
set(handles.par_1_value,'Visible','on')
set(handles.par_1_title,'Visible','on')
set(handles.par_1_LFO,'Visible','on')
```

El elemento *handles.par_1* es el desplazador, *handles.par_1_value* hace referencia al cuadro de texto para introducir el valor directamente, mientras que *handles.par_1_title* es el título del parámetro. La numeración de los parámetros abarca desde 1 hasta 6. En el caso de que se quiera dar la posibilidad de usar un LFO para el parámetro, se deberá también hacer visible *handles.par_1_LFO*, cuya documentación se encuentra en el siguiente punto.

Oscilador de baja frecuencia

El uso del LFO está disponible para la generación de una señal de entrada sintética, para el control de parámetros del efecto y para la modulación en frecuencia (FM), amplitud (AM) y fase (PM) de los LFOs aplicados a los dos uso anteriores. Cada LFO es aplicado independientemente de los demás, además de estar adaptado a los umbrales marcados por la función a la que se va a aplicar, séase valores máximos y mínimos de señal y parámetros.

Generador de señal

Se ofrece la posibilidad de crear la propia señal de entrada mediante el oscilador. En este caso la posibilidad de usar una señal externa está deshabilitada, ya que resulta redundante a obtener la señal de entrada por el panel de navegación de archivos. Para evitar la componente continua, también se ha deshabilitado el parámetro *offset* para esta aplicación.

Los valores de frecuencia del oscilador abarcan desde 20Hz hasta 20000Hz . Las variables de este tipo de oscilador se pasan bajo el nombre de: `LFO_0`.

Automatización de parámetros

Todos los parámetros incluyen la posibilidad de ser controlados mediante un oscilador de baja frecuencia. Por ello, la interfaz incluye un botón para configurar el LFO en el parámetro seleccionado. Tanto la frecuencia de muestreo del LFO aplicado a parámetros como su duración coincide con la de la señal de entrada. Por tanto, la configuración del LFO en la interfaz del usuario no se habilita hasta que no se haya importado la señal de entrada.

Los valores de frecuencia del oscilador abarcan desde $0,1\text{Hz}$ hasta 10Hz . Las variables de este tipo de oscilador se pasan bajo el nombre de: `LFO_#`, siendo # desde 1 hasta 6 (coincidente con el parámetro a aplicar el LFO).

Modulador

Para los dos casos anteriores se habilita la opción de modular a su vez el parámetros de amplitud del mismo LFO mediante un LFO secundario. El oscilador secundario sin embargo, para evitar problemas de complejidad, ya no ofrece esta opción anidada. Matlab no permite que una interfaz de usuario se llame de nuevo a sí misma, por lo que se ha copiado en `z_modulador_GUI.m` la interfaz `z_LFO_GUI.m`, pero ambas llaman a `z_LFO_graf.m` para crear la señal del oscilador.

Los valores de frecuencia del oscilador abarcan desde $0,1\text{Hz}$ hasta 100Hz . Las variables de este tipo de oscilador se pasan bajo el nombre de: `LFO.frecuencia`, `LFO.amplitud` y `LFO.fase`.

Para realizar la modulación en frecuencia para algunos tipos de onda, se usará una variable que indique el número de muestras de un periodo y que se actualice en cada pasada del bucle que recorre la señal moduladora. Con la ayuda de un contador de muestras, cuando éste sea mayor que la variable anterior, se produce el cambio característico de la forma de onda (por ejemplo, el cambio de pendiente en la señal triangular). El contador además aumenta dependiendo de la diferencia entre la fase instantánea y la anterior.

Tipos de osciladores

Para el control de los osciladores el usuario establece las siguientes magnitudes:

Frecuencia	$f[\text{Hz}]$	Frecuencia real del oscilador.
Amplitud	A	Valor pico del oscilador, es decir, variación máxima respecto al offset.
Fase	ϕ	Fase real del oscilador, en radianes.
Offset	y_0	Valor inicial y al mismo tiempo valor medio de la señal.

Señal sinusoidal

La señal obtenida es la siguiente:

$$y = A(i) \cdot \sin(2 \cdot \pi \cdot f(i) \cdot t + \phi(i)) + y_0$$

Señal triangular

Se divide el periodo de la señal en dos partes y se inicializa el primero con una cierta fase. Las parte, que son alternadas mediante un contador dependiente de la variación de frecuencia y fase, se caracterizan por la pendiente de la recta igual a A y $-A$ cuyo signo varía en cada alternancia. Debido a los errores de truncamiento de cada mitad de señal del LFO, se acumulará un error que quedará reflejado en la frecuencia del LFO, aunque éste resulta aceptable para el rango permitido en la interfaz. La fase además se inicializa en $\pi/2$ para asemejarse a la señal anterior.

```
m = 1;      % Signo de la pendiente
n = 0;      % Variable de muestras
j = mod(floor(L)*(fase(1)+pi/2)/2/pi,floor(L/2)); % Contador
if fase(1) < pi/2 || fase(1) >= 3*pi/2
    m = 1;
else
    m = -1;
end
n = m*(mod(floor(L)*(fase(1)+pi/2)/2/pi,floor(L/2)) - L/4);
for i = 1:MuestrasTotales
    L = 1/(Ts*frecuencia(i)); % Número de muestras de un periodo
    if j > floor(L/2) % Cambio de mitad de periodo
        j = 1;
        m = -m;
    else
        j = j+1+L*(fase(i)-faseanterior)/2/pi;
    end
    x(i) = 4*amplitud(i)/L*n + offset;
    n = n+m*(1+L*(fase(i)-faseanterior)/2/pi);
    faseanterior = fase(i);
end
```

Señal diente de sierra ascendente

Se divide el periodo de la señal en dos partes, correspondientes a las partes positivas y negativas. Primero se inicializa la mitad positiva, para luego entrar en el bucle en ambas partes como una recta que se reinicia con un contador dependiente de la variación de frecuencia y fase. Debido a los errores de truncamiento de cada cuarto del periodo de señal de LFO, se acumulará un error que quedará reflejado en la frecuencia del LFO, aunque éste resulta aceptable para el rango permitido en la interfaz.

```
j = mod(floor(L)*fase(1)/2/pi,floor(L)) + 1; % Contador de cambio de periodo
n = mod(floor(L)*fase(1)/2/pi,floor(L)) - floor(L/2); % Variable de muestras
for i = 1:MuestrasTotales
    L = 1/(Ts*frecuencia(i)); % Número de muestras para el cambio
    if j > floor(L) % Cambio de cuarto de periodo
        n = -floor(L/2); % Reset de la variable de muestras
        j = 1;
    else
        j = j+1+L*(fase(i)-faseanterior)/2/pi;
    end
    x(i) = 2*amplitud(i)/L*n + offset;
    n = n+1+L*(fase(i)-faseanterior)/2/pi;
    faseanterior = fase(i);
end
```

Señal diente de sierra descendente

Misma implementación que en el tipo de señal anterior, salvo un signo negativo en la pendiente de la recta.

Señal cuadrada

Se divide el periodo de la señal en dos partes, igualando cada una de ellas a la amplitud positiva y negativa respectivamente, y se reitera con un contador dependiente de la variación de frecuencia y fase. Debido a los errores de truncamiento de cada cuarto del periodo de señal de LFO, se acumulará un error que quedará reflejado en la frecuencia del LFO, aunque éste resulta aceptable para el rango permitido en la interfaz. La variable actualizada para cada cambio de frecuencia es la mitad del periodo.

```
if fase(1) < pi                                % Signo de la pendiente
    m = 1;
else
    m = -1;
end
j = mod(floor(L)*fase(1)/2/pi, floor(L/2)) + 1;    % Contador
for i = 1:MuestrasTotales
    L = 1/(Ts*frecuencia(i));                    % Número de muestras para el cambio
    if j > floor(L/2)                             % Cambio de cuarto de periodo
        j = 1;
        m = -m;                                   % Cambio del signo de la pendiente
    else
        j = j+1+L*(fase(i)-faseanterior)/2/pi;
    end
    x(i) = m*amplitud(i) + offset;
    faseanterior = fase(i);
end
```

Ruido blanco gaussiano

Se genera una señal ruidosa con la función `randn()`. Finalmente se recorta cada una de las muestras de la señal para que no sobrepase los umbrales requeridos en amplitud. En este caso los parámetros frecuencia y fase evidentemente son indiferentes.

```
x = amplitud.*randn(MuestrasTotales,1)/2 + offset;
for n = 1:MuestrasTotales                    % Recorte de señal
    if x(n) < limites.Min
        x(n) = limites.Min;
    elseif x(n) > limites.Max
        x(n) = limites.Max;
    end
end
```

Señal externa

La señal proviene de un archivo externo. Ésta se recortará en caso de ser de mayor duración que la señal de entrada, y se repetirá en caso de durar menos. No permite ningún tipo de modulación.

```
if length(file.data(:,1)) < length(LFO.n)        % Repetición de señal
    for i = 1:floor(length(LFO.n)/length(file.data(:,1)))
        LFO.x((i-1)*length(file.data)+1:i*length(file.data)) = file.data(:,1);
    end
    LFO.x(i*length(file.data)+1:length(LFO.n)) =
file.data(1:length(LFO.n)-i*length(file.data),1);
else                                              % Recorte de señal
    LFO.x = file.data(1:length(LFO.n),1);
end
LFO.x = (LFO.x+1)*(limites.Max-limites.Min)/2+limites.Min;
```


Actualización de parámetros en el efecto

Los programas del LFO exportan al programa del efecto el oscilador deseado en la variable `handles.LFO_#` (para # desde 1 hasta 6). Esta variable de tipo *structure* contiene la señal del oscilador en `handles.LFO_#.x` de tipo *array* y de longitud igual a la longitud de la señal de entrada. La actualización de los parámetros en función de ésta señal del LFO se debe realizar en la función `<efecto>_Aplicar_Callback`, que es donde se procesa la señal. En ocasiones, algunos procesamiento resultan tan costosos cuando varían ciertos parámetros que se opta por no actualizar el parámetro en cada muestra, sino cada un número entero de ellas.

En el siguiente ejemplo de actualización de los parámetros los parámetros se actualizan cada `LFO_res` muestras de la señal.

```
if handles.LFO_1.checkbox || handles.LFO_2.checkbox % Con LFO
    par1(1:length(handles.x(:,1)),1) = handles.par1;
    par2(1:length(handles.x(:,1)),2) = handles.par2;
    LFO_res = 100; % Resolución de actualización del LFO en muestras de señal
    wb = waitbar(0, 'Processing...', 'Name', 'Delay'); % Diálogo de espera
    for n = 0:LFO_res:length(handles.x(:,1))-LFO_res
        if handles.LFO_1.checkbox
            par1(n+1:n+LFO_res,:) = original(n+1-M:n+LFO_res-M,:);
        end
        if handles.LFO_2.checkbox
            par2(n+1:n+LFO_res,:) = handles.LFO_2.x(n+1);
        end
        waitbar(n/L,wb, 'Processing...'); % Diálogo de espera
    end
    if handles.LFO_1.checkbox
        par1(n+LFO_res+1:length(handles.x(:,1)),:) = handles.LFO_1.x(n+1);
    end
    if handles.LFO_2.checkbox
        par2(n+LFO_res+1:length(handles.x(:,1)),:) = handles.LFO_2.x(n+1);
    end
    handles.y = (1-d).*original + d.*delay;
end
```

Gráficos

Gráficos de entrada y salida

Señales temporales

Grafica cada uno de los dos canales de las señales de entrada y salida. Para el caso de la señal de salida se muestra directamente en la interfaz, mientras que en el caso de la señal de entrada es necesario pulsar el botón correspondiente.

Espectros

La interfaz de usuario muestra el espectro de las señales de entrada y salida. En la ampliación se ofrece el espectrograma de cada canal de las mismas, para tener una visión temporal. Tanto la escala de frecuencias como de nivel de señal están en formato logarítmico (no es el caso del eje temporal en el caso del espectrograma).

Gráfico del efecto

La interfaz gráfica de plantilla incluye unos ejes, referenciados como `handles.graf`, para representar cualquier propiedad característica de cada efecto, que se encuentra inicialmente oculto, junto con un botón de ampliación del gráfico. Para habilitarlos, deberá añadir las siguientes líneas a la función `<efecto>_OpeningFcn`:

```
set(handles.graf,'Visible','on')
set(handles.graf_open,'Visible','on')
```

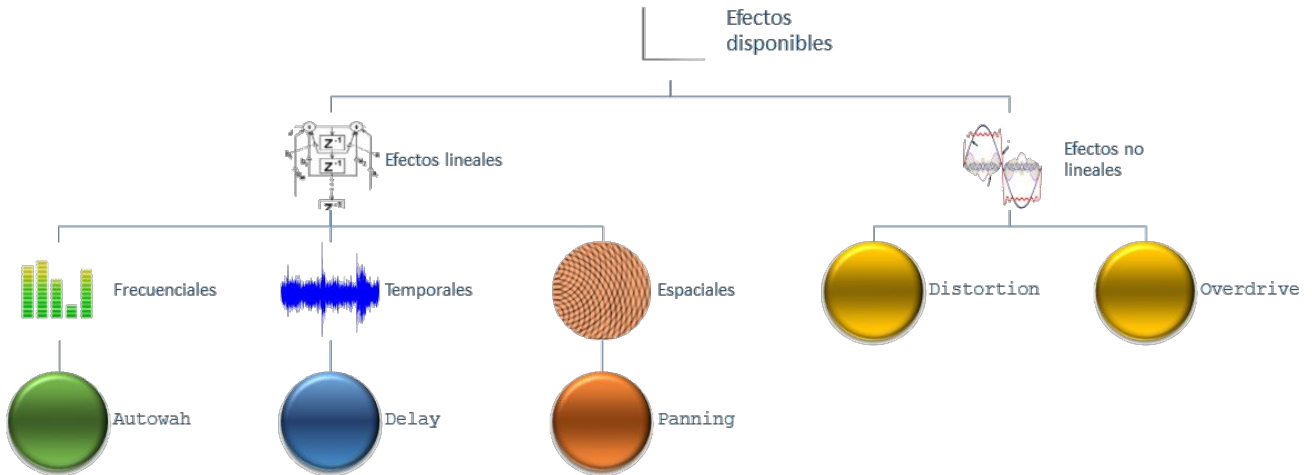
En la ampliación del gráfico, se copiarán todas las propiedades añadidas al gráfico.

Comparación entre espectros de entrada y salida

Se puede desplegar un gráfico comparativo entre los espectros de entrada y salida pulsando el botón *Comparar*. Tanto la escala de frecuencias como de nivel de señal están en formato logarítmico. El espectro de las señales del canal L es verde mientras que el del canal R es rojo.³

³ Debido a que en la mayoría de los casos coinciden prácticamente ambas señales, la señal L se superpondrá y probablemente se aprecie el gráfico únicamente de color verde.

Efectos disponibles



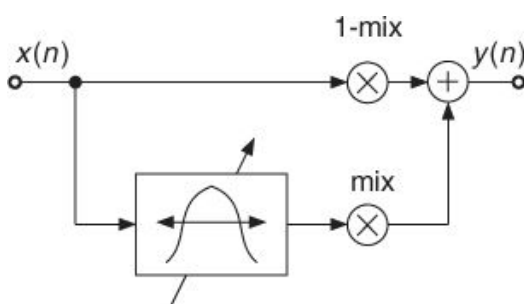
Clasificación de los efectos

Los efectos actualmente disponibles pueden ser clasificados de la siguiente manera.

- Efectos lineales. Conjunto de efectos en los que se realiza un procesamiento lineal, en otras palabras, un filtrado. Estos efectos se caracterizan por su respuesta al impulso y pueden ser tratados matemáticamente con mayor facilidad.
 - Frecuenciales. Agrupa todos los efectos centrados en un filtrado variable en frecuencia.
 - Autowah
 - Temporales. Agrupa todos los efectos realizados mediante suma de retardos temporales de la señal original.
 - Delay
 - Espaciales. Efectos cuyo objetivo es producir en el oyente una sensación de situación espacial concreta.
 - Panning
- Efectos no lineales. Conjunto de todos aquellos efectos con procesos no lineales. No pueden ser caracterizados mediante una respuesta al impulso.
 - Distortion
 - Overdrive

Autowah

Descripción	Filtro paso banda estrecho con una frecuencia central variable, que se controla mediante un LFO. El nivel de autowah marca la proporción de las señales original y filtrada.	
Tipo	Lineal frecuencial	
Parámetros	<i>BW</i>	Ancho de banda del filtro
	<i>f_c</i>	Frecuencia central del filtro
	<i>mix</i>	Nivel de autowah
Referencia	2.4.1 Wah-wah Filter	



El efecto implementado únicamente realiza un filtrado paso banda con los parámetros especificados. Para que el efecto se corresponda con un auténtico *autowah* es necesario aplicar un LFO a la frecuencia central del filtro.

No está de más mencionar que la diferencia entre un *Wah-Wah Filter* y un *Autowah Filter* reside en el sujeto que realiza la variación de la frecuencia central. En el primer caso la realiza el usuario manualmente (típicamente mediante un pedal), mientras que en el segundo caso, ésta es automatizada por el

oscilador de baja frecuencia. En ésta aplicación, dado que el procesamiento no es tiempo real, el control sólo puede ser realizado por el oscilador.

Filtro

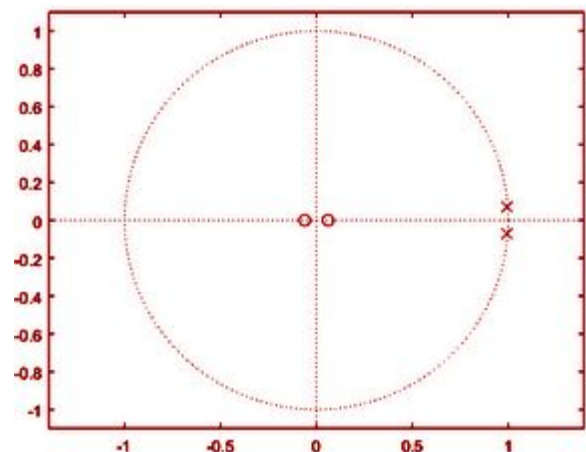
El filtro que se ha implementado se corresponde con un filtro FIR digital de segundo orden paso banda. Aunque Matlab incluye funciones de diseño y procesamiento de filtro, con motivo de optimizar la variación de las variables del filtro cuando se usa el LFO, el filtrado de la señal se hace manualmente.

$$c = \frac{\tan(\pi BW/f_s) - 1}{\tan(\pi BW/f_s) + 1}$$

$$d = -\cos(2\pi \frac{f_c}{f_s})$$

$$H(z) = \frac{1}{2} \left(1 - \frac{-c + d(1-c)z^{-1} + z^{-2}}{1 + d(1-c)z^{-1} - c \cdot z^{-2}} \right) = \frac{1}{2} \cdot \frac{1 - (1+c)z^{-2}}{1 + d(1-c)z^{-1} - c \cdot z^{-2}}$$

En la figura se muestra el diagrama de polos y ceros del sistema para un ancho de banda de 50Hz, una frecuencia central de 1000Hz y una frecuencia de muestreo de 44100Hz.



```

y = zeros(length(x(:,1)),2);           % Filtro banda eliminada
xh = zeros(2,2);                       % Inicialización de memorias
c = (tan(pi*BW/fs)-1) / (tan(pi*BW/fs)+1); % Coeficientes
d = -cos(2*pi*f_c/fs);
for n = 1:length(x)
    xh_new = x(n,:) - d.*(1-c).*xh(1,:) + c.*xh(2,:);
    ap_y = -c .* xh_new + d.*(1-c).*xh(1,:) + xh(2,:);
    xh = [xh_new; xh(1,:)];
    y(n,:) = x(n,:) - ap_y;
end

```

Gráfico del efecto

En el gráfico del efecto se representa lo siguiente:



Filtro de autowah.



Señal limpia.



Máximo o mínimo del ancho de banda del filtro de autowah (si LFO 1 está activo).



Frecuencia de corte superior máxima o inferior mínima del filtro de autowah (si LFO 2 está activo).
No tiene en cuenta la variación del ancho de banda si LFO 1 está activo simultáneamente.



Rango máximo de variación del nivel de autowah (si LFO 3 está activo).

Una configuración óptima de autowah típicamente usada para guitarra es la siguiente:

<i>Si LFO</i>	<i>Forma de onda</i>	<i>Frecuencia</i>	<i>Amplitud</i>	<i>Offset</i>
Parámetro	Configuración			
Ancho de banda		100Hz		
Frecuencia central	Sinusoidal	0.5Hz - 2Hz	1000	1050
Nivel de autowah		0.9		

Delay

Descripción	Suma a la señal original de entrada la misma señal con un retardo. El nivel de delay marca la proporción de las señales original y retardada.	
Tipo	Lineal temporal	
Parámetros	<i>delay</i>	Tiempo de delay en segundos
	<i>d</i>	Nivel de delay
Referencia	2.5 Basic Delay Structures	

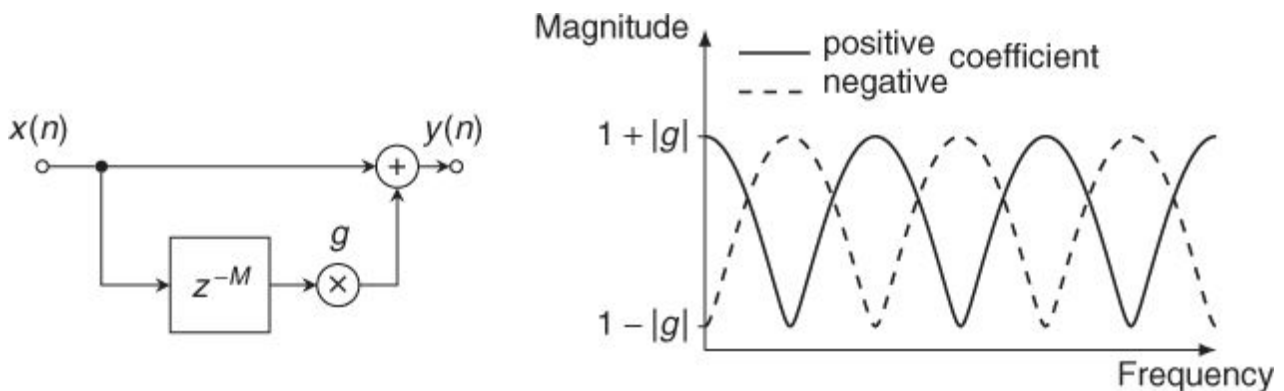
Es necesario saber de antemano la frecuencia de muestreo de la señal, puesto que el tiempo de retardo se define en segundos. El concepto a seguir es el siguiente:

$$y[n] = (1 - d) \cdot x[n] + d \cdot x[n - M] \quad M \quad \text{número de muestras de delay}$$

$$H(z) = 1 + d \cdot (-1 + z^{-M})$$

$$M = \text{delay} \cdot f_s$$

Para valores positivos de d , el filtro amplifica todas las frecuencias múltiples de $1/t$ y atenúa las que se sitúan entre las anteriores.



Aunque es razonable que la señal de salida dure suficientemente para que la señal retardada termine, surgen problemas de delimitación de *arrays* cuando se usa el LFO del parámetro de tiempo de delay. Debido a la complejidad de determinar la duración final que tendría el audio de salida, se ha optado por limitar la duración de la señal de salida a la misma que la entrada, por lo que la señal retardada no acaba (salvo en el caso $M = 0$).

Cuando se aplica un oscilador para el parámetro de tiempo de delay, surgen efectos interesantes de variación de la frecuencia percibida por el oyente. Esto se debe a que la variación del tiempo de retardo produce una compresión o expansión de la señal al adelantar o atrasar en el instante la señal retardada de forma progresiva. El aumento de la frecuencia percibida depende de la derivada de la variación de este tiempo de retardo. Este efecto sigue el mismo principio que el efecto Doppler.

Modulación de duración

Si el tiempo de un filtro delay es mayor que 50 ms , el sonido percibido por el cerebro humano es un eco. Si éste se sitúa entre 15 ms y 50 ms , se percibe una rápida repetición llamada *slapback* o *doubling*. En cambio, si es menor que 15 ms , el efecto producido es *flanging*. A partir de *delays* se pueden obtener los siguientes efectos comúnmente conocidos.

	Si LFO	Forma de onda	Frecuencia	Amplitud	Offset
Efecto	Parámetro	Configuración			
Flanger	Tiempo de delay	Sinusoidal	$0.1\text{Hz} - 10\text{Hz}$	$0.001s$	$0.005s - 0.010s$
	Nivel de delay		0.5		
Chorus	Tiempo de delay	Ruido AWGN	Indiferente	$0.002s$	$0.010s - 0.025s$
	Nivel de delay		0.5		
Vibrato	Tiempo de delay	Sinusoidal	$1\text{Hz} - 2\text{Hz}$	$0.004s$	$0.004s$
	Nivel de delay		1		

Flanger

Descripción	Filtro delay con un tiempo de retardo menor que 15ms , que varía sinusoidalmente mediante un LFO con una frecuencia entre 0.1Hz y 1Hz .
Tipo	Lineal temporal
Referencia	2.6.2 Flanger, Chorus, Slapback, Echo

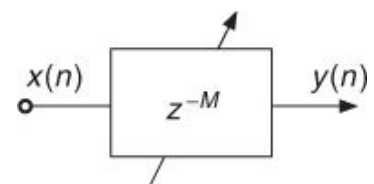
Chorus

Descripción	Filtro delay cuyo tiempo de retardo se encuentra entre 10ms y 25ms , y varía ligeramente de forma aleatoria.
Tipo	Lineal temporal
Referencia	2.6.2 Flanger, Chorus, Slapback, Echo

Vibrato

Descripción	Variación frecuencial mediante un LFO.
Tipo	Lineal frecuencial
Referencia	2.6.1 Vibrato

Su implementación se realiza mediante variaciones del tiempo de retardo, usando un LFO, de forma que se produzcan leves variaciones de frecuencia en torno a la frecuencia central del sonido original. Hay que ser sutil a la hora de elegir los parámetros del LFO ya que la variación de frecuencia dependerá de la frecuencia del tono original, por lo que una configuración del LFO no es válida para dos tonos distantes en frecuencia.



Distortion

Descripción	Amplificador no lineal. El nivel de distortion marca la proporción de las señales original y distorsionada.		
Tipo	No lineal		
Parámetros	k	Nivel de distortion	
Referencia	4.3.2 Overdrive, Distortion and Fuzz		

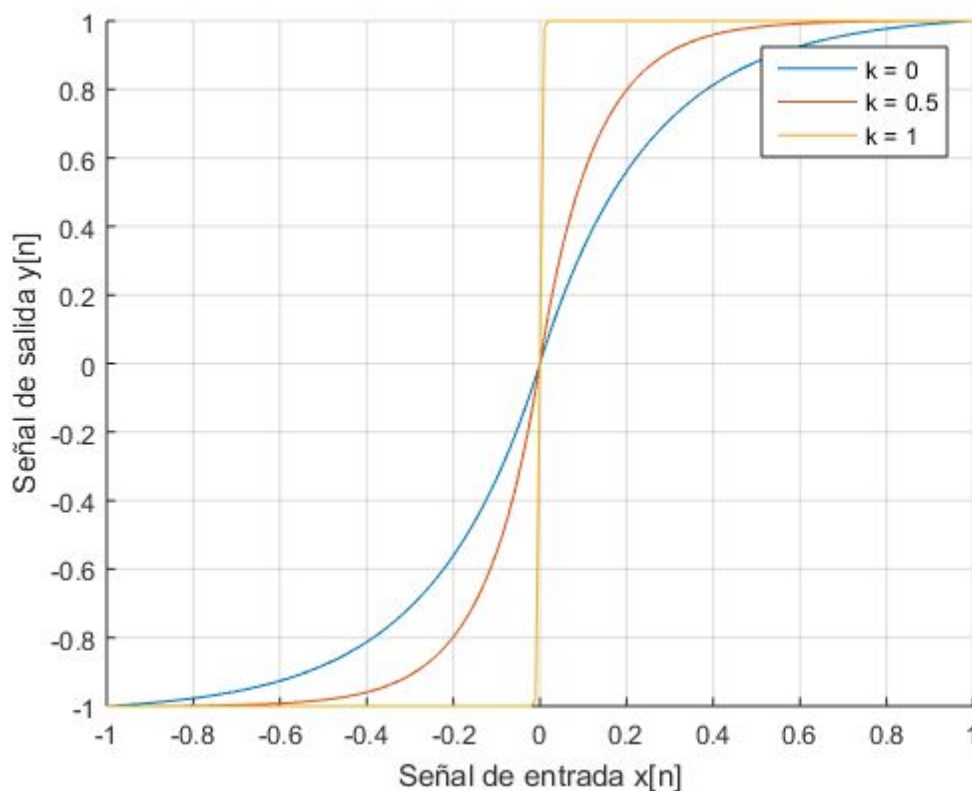
Aunque su implementación es muy variada, la que se ha usado es:⁴

$$y[n] = \frac{x[n]}{|x[n]|} \cdot \frac{e^l}{e^l - 1} \cdot (1 - e^{-l \cdot |x[n]|})$$

$x[n], y[n]$ entrada y salida normalizadas

$$l = 4 \cdot \frac{1}{1-k}$$

La función de la variable l se ha ajustado empíricamente de forma que la variación del nivel de overdrive se ajuste a un rango de variación de la linealidad cómodo para el usuario. Además, en su implementación hay que evitar el caso extremo $k = 1$, donde $l = \infty$, por lo que se adjudicará el valor $l = 100$ para $k > 0.99$.



En el caso de la *distortion*, el nivel cero no se aproxima tanto como con el *overdrive*, además de que es notable que la distorsión de la señal y la producción de armónicos es mucho más notable, lo que ocurre a menudo en la práctica.

⁴ La componente $x[n]/|x[n]|$ equivale a la función signo.

Overdrive

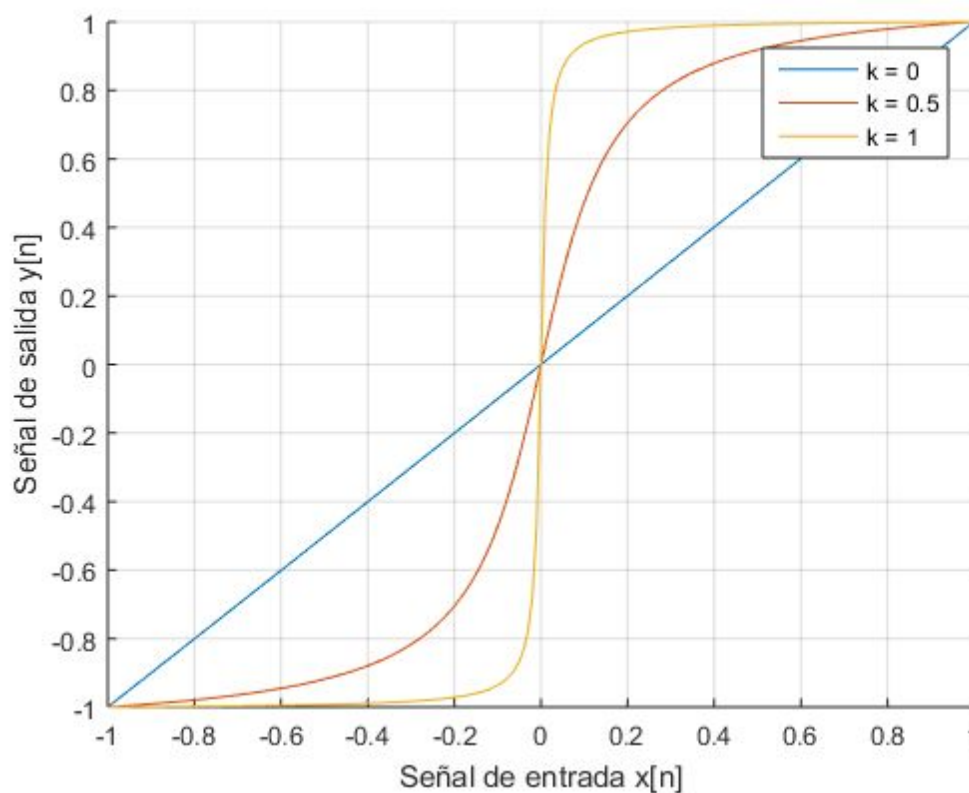
Descripción	Amplificador no lineal, aunque con zona lineal mayoritaria. El nivel de delay marca la proporción de las señales original y distorsionada.		
Tipo	Quasi-lineal		
Parámetros	k	Nivel de overdrive	
Referencia	4.3.2 Overdrive, Distortion and Fuzz		

Aunque su implementación es muy variada, la que se ha usado es:

$$y[n] = \frac{\text{atan}(l \cdot x[n])}{\text{atan}(l)} \quad x[n], y[n] \quad \text{entrada y salida normalizadas}$$

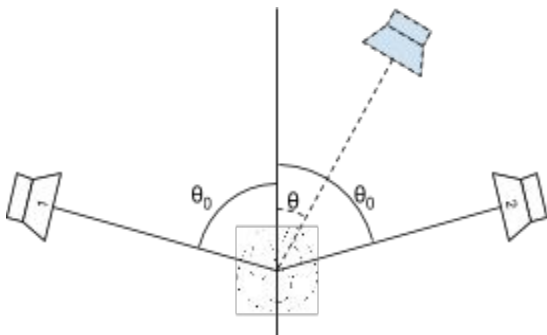
$$l = 10 \cdot \left(\frac{1}{1 - 0.9 \cdot k} - 1 \right)$$

La función de la variable l se ha ajustado empíricamente de forma que la variación del nivel de overdrive se ajuste a un rango de variación de la linealidad cómodo para el usuario. Además, en su implementación hay que evitar el caso extremo $k = 0$, donde se encuentra la asíntota de la arcotangente, por lo que se adjudicará el valor $l = 0.136$ para $k < 0.01$.



Panning

Descripción	Produce la ilusión al oyente de una dirección determinada de origen del sonido, siempre y cuando se cuente con al menos dos canales estéreo y dos fuentes separadas suficientemente rodeando al oyente.	
Tipo	Lineal espacial	
Parámetros	θ	Acimut virtual
	θ_i	Acimut de altavoz i
Referencia	5.3 Basic Spatial Effects for Stereophonic Loudspeaker and Headphone playback	



Para el caso de dos altavoces, separados equidistantemente del oyente, se puede virtualizar la dirección de origen del sonido siguiendo la ley de Bennett. Para ello se desequilibran los canales estéreo modificando sus amplitudes.

En la implementación se ha limitado θ_0 a 89° para evitar la asíntota de la tangente. Además es obvio que el ángulo virtual, en módulo, es siempre menor que el de las fuentes, por lo que se va a limitar a como máximo el ángulo de las fuentes seleccionado en ese instante.

$$x_i[n] = g_i \cdot x[n]$$

$$\frac{g_1}{g_2} = \frac{\tan(\theta_0) - \tan(\theta)}{\tan(\theta_0) + \tan(\theta)}$$

$$x_i[n]$$

canal i

$$g_i$$

ganancia del canal i

$$\theta$$

ángulo virtual

$$\theta_0$$

ángulo de las fuentes

Para no saturar la señal, siempre se aplicará una atenuación en vez de una ganancia al canal correspondiente, es decir, si uno de los canales debe ser un número de veces mayor que el otro, se dividirá la señal del otro canal este número de veces.

```
if k>1
    g(2) = 1/k;
else
    g(1) = k;
end
```

Glosario

Diagrama de polos y ceros	Representación gráfica de una función de transferencia racional en el plano complejo, donde se pueden apreciar algunas propiedades del sistema.
Digital Audio Workstation	Aplicación software destinada a la grabación, edición y producción de archivos de audio.
Filtro FIR (Finite Impulse Response)	Filtro cuya respuesta al impulso es de duración finita, debido a que no existe una realimentación en el filtro.
Filtro IIR (Infinite Impulse Response)	Filtro cuya respuesta al impulso es de duración infinita, debido a que existe una realimentación en el filtro.
LFO (Low Frequency Oscillator)	Oscilador de baja frecuencia.
Rango dinámico de una señal	Margen de valores en que aparece la señal, es decir, diferencia entre el pico máximo y el mínimo de una señal.

Bibliografía

Wikipedia. Wikimedia Foundation. Web. 11 Feb. 2015. <<http://www.wikipedia.org/>>.

"Browse the Auralization Database | The Open Acoustic Impulse Response Library." *Browse the Auralization Database | The Open Acoustic Impulse Response Library*. Web. 16 Feb. 2015. <<http://www.openairlib.net/auralizationdb>>.

"Control Systems/State-Space Equations." - Wikibooks, *Open Books for an Open World*. Web. 18 Mar. 2015. <http://en.wikibooks.org/wiki/Control_Systems/State-Space_Equations>.

"Documentation." MATLAB. Web. 26 Mar. 2015. <<http://es.mathworks.com/help/?refresh=true>>.

Zölzer, Udo. *DAFX: Digital Audio Effects*. Chichester: Wiley, 2011. Print.