



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



**“Desarrollo de herramientas de
usuario para la generación de
efectos digitales sobre señales de
audio en tiempo real”**

TRABAJO FINAL DE GRADO

Autor:

Manuel Larrosa Ortuño

Tutor:

Miguel Ferrer Contreras

GANDIA, 2019

Resumen

En este proyecto se aborda el diseño y prototipado de una serie de herramientas de usuario para la generación de efectos sobre la señal de audio en tiempo real en forma de *plugin* gracias al uso de las herramientas proporcionadas por el paquete Audio Toolbox del entorno de desarrollo MATLAB.

En primer lugar, se implementa una emulación del filtro analógico musical conocido como *Moog Ladder* a partir de un modelo digital simplificado (y optimizado para su ejecución en tiempo real), cuyo comportamiento imita las características no lineales y la respuesta frecuencial del filtro analógico original.

En segundo lugar, se desarrolla una serie de efectos digitales incluidos bajo un mismo prototipo de *plugin* cuyo objetivo es el de emular el sonido de los equipos de efecto eco basado en cinta magnética. Este sonido se obtiene a partir de un eco digital con realimentación que puede ser complementado con el resto de efectos que emulan características analógicas como la saturación no lineal, el ruido de fondo o las variaciones de tono producidas por la oscilación irregular de la bobina de cinta o las variaciones de velocidad angular en el cabezal. Adicionalmente, en este paquete multi-efectos se incluye una implementación del efecto *flanger* cuyos orígenes se sitúan junto al uso de magnetófonos, y que junto al resto de efectos proporcionen al usuario una herramienta para la obtención de una estética sonora de estilo retro-futurista.

Palabras clave: efectos digitales de audio, procesado en tiempo real, efectos virtuales analógicos, MATLAB, plugin, emulación.

Abstract

This project addresses the design and prototyping of a series of user tools for the generation of real-time digital audio effects in the form of plugin, with the use of the MATLAB development tools included in the Audio Toolbox.

In the first place, an emulation of the Moog Ladder musical analog filter is implemented based on a simplified (and optimized for real-time execution) digital model which emulates the non-linear characteristics and the frequency response of the original analog filter.

In the second place, a prototype of multi-effect plugin is developed to emulate the sound of the magnetic tape echo effect. This sound is obtained from a digital echo with feedback, which can be complemented with a package of effects that emulates analog sound characteristics like the non-linear tape saturation, the background noise and the pitch variations produced by the irregular oscillations of the tape reels and the variations in the angular speed at the capstan. Additionally, in this multi-effect package an implementation of the flanger effect is included, whose origins are placed next to the use of magnetic tape recorders and along with the rest of effects, provides to the user a tool for obtaining a retro-futuristic style of sound aesthetics.

Keywords: digital audio effects, real-time processing, virtual analog effects, MATLAB, plugin, emulation.

Índice

1	Introducción	1
1.1	Una perspectiva histórica	1
1.2	Objetivos	3
1.3	Estructura del documento	4
2	Herramientas de trabajo	6
3	Diseño y prototipado de un filtro analógico virtual	7
3.1	El filtro <i>Moog Ladder</i>	7
3.2	Modelos digitales del filtro <i>Moog Ladder</i>	8
3.3	Adaptación del modelo no lineal simplificado para tiempo real	11
3.4	Prototipado de un <i>plugin</i> de filtrado en tiempo real	13
3.5	Análisis frecuencial del filtro resultante	13
4	Diseño y prototipado de un <i>plugin</i> multi-efectos	17
4.1	Introducción y aspectos sobre el diseño	17
4.2	Efecto de eco simple con retroalimentación	18
4.3	Emulación de saturación de cinta magnética	19
4.4	Emulación de ruido <i>hiss</i>	21
4.5	Efecto <i>Vibrato</i>	23
4.6	Efecto <i>flanger</i>	25
4.7	Prototipo de la interfaz obtenida con la herramienta Audio Testbench	26
5	Futuras líneas de trabajo	29
6	Conclusiones	31
7	Referencias	33
A	Glosario de Acrónimos	34
B	Glosario de Términos	35

Índice de figuras

1	Circuito Moog Ladder (Moog, 1965)	8
2	Respuesta en frecuencia del filtro Moog para diferentes valores de ganancia de retroalimentación $k = [0, 0.3, 0.6, 0.9, 0.99]$ (Stilson y col., 1996)	8
3	Modelo digital del filtro Moog (Stilson y col., 1996)	9
4	Modelo digital mejorado del filtro Moog (Välämäki y Huovilainen, 2006)	10
5	Señal dientes de sierra filtrada con los tres modelos digitales de filtro Moog: (a) el modelo lineal Stilson-Smith de la Figura 3, (b) el modelo no lineal simplificado de la Figura 4, y (c) el modelo completo no lineal propuesto en (Huovilainen, 2004)	10
6	Estructura simplificada de filtro Moog no lineal (Zölzer, 2011)	11
7	Interfaz gráfica del <i>plugin</i> de filtrado en escalera generada por <i>AudioTestBench</i>	14
8	Respuesta en frecuencia del ruido blanco gaussiano a la entrada del filtro	15
9	Respuesta en frecuencia del filtro para una frecuencia de corte de 1kHz y un valor 0 de resonancia	15
10	Respuesta en frecuencia del filtro para una frecuencia de corte de 1kHz y un valor 1 resonancia	16
11	Estructura del diseño propuesto de <i>plugin</i> multi-efectos	18
12	Estructura del eco simple con retroalimentación	19
13	Aproximación de la función no lineal de la saturación de cinta magnética (Zölzer, 2011)	20
14	Tono puro de 500 Hz sin efecto de saturación	21
15	Tono puro de 500 Hz con un factor de saturación $\alpha = 60$	21
16	Tono de 500 Hz con un ruido añadido de -20dBW de potencia y filtrado a una frecuencia de corte de 1 kHz	23
17	Esquema del efecto <i>vibrato</i>	24
18	Obtención del efecto <i>flanger</i> con grabadoras magnetofónicas (Bode, 1984)	25
19	Esquema del funcionamiento del efecto <i>flanger</i>	26
20	Interfaz gráfica de usuario generada por Audio Testbench	27

1 Introducción

1.1 Una perspectiva histórica

La historia de la modificación del sonido (o de los efectos de audio) es tan antigua como la historia de los instrumentos musicales electrónicos y los dispositivos de transmisión, grabación y reproducción de audio.

Los medios para modificar sonidos generados de forma electrónica son conocidos desde finales del siglo XIX, cuando Thaddeus Cahill creó su Telarmonio (o Dinamófono) en el año 1896, considerado el primer instrumento enteramente electrónico y polifónico.

Con la llegada de la era electrónica, impulsada por la creación de la válvula termoiónica y el posterior desarrollo de los dispositivos de estado sólido, una inmensa variedad de modificadores del sonido fueron desarrolladas para filtrar, distorsionar, ecualizar, modular la frecuencia y amplitud, comprimir, reveberar, repetir la señal, cambiar el tono mediante desplazamientos de frecuencia, analizar y sintetizar sonidos de origen natural o artificial (Bode, 1984), entre otros.

Con la aparición de los dispositivos de grabación y reproducción basados en la cinta magnética (o magnetófonos) se introdujeron los primeros efectos de repetición que sustituirían los costosos métodos que requerían del uso de cámaras de eco o reverberantes, ya que mediante las líneas de repetición artificiales se generaban estos efectos de forma similar y con el uso de equipos que requerían muy poco espacio gracias a la cinta magnética (Bode, 1984).

Con la llegada de la tecnología digital, las señales de audio, que en el dominio analógico quedan representadas mediante señales eléctricas con valores continuos, pasaron a representarse mediante valores discretos gracias a la técnica del muestreo. Estas muestras, representadas a nivel físico mediante código binario, introducían una capa de abstracción respecto a la señal original, y en muchos casos, implicaba una pérdida de la calidad física, ya que las muestras no representan el sonido en sí, sino que son un medio para almacenar datos que un procesador digital manipula mediante operaciones matemáticas. Pero estos datos sólo pueden representar sonidos mediante su reconstrucción a través de un conversor que los transforme de nuevo al dominio analógico y finalmente puedan ser escuchados mediante su transformación a ondas acústicas a través de un transductor electroacústico (o altavoz), donde se apreciaban en ocasiones diferencias perceptuales con respecto a la señal original.

Otro defecto ocasionado por la tecnología digital fue el error introducido como consecuencia de la discretización, y que se percibía como un ruido de fondo, que a diferencia del presente en los sistemas analógicos, éste queda correlado con la señal original e introduce una distorsión inarmónica (generadas por los productos de intermodulación) que afecta al cuerpo y a la pureza de la señal con la consecuente percepción de un sonido áspero,

frío, de excesiva dureza y en contraposición a la suave distorsión y ruido analógico, que genera el efecto subjetivo de calidez sonora. Esto es debido a que cierto nivel de ruido incorrelado, en ocasiones resulta deseable para enmascarar los defectos desagradables al oído producidos por los conversores digitales. Además, a diferencia del ruido digital, el ruido analógico resulta eufónico, es decir, agradable al oído. Un ejemplo que representa muy bien cómo un sonido objetivamente peor resulta subjetivamente deseable es el sonido de la cinta magnética, considerado aún a día de hoy, el medio más musical por muchos profesionales (Katz, 2015).

Ante el exponencial avance de los procesadores digitales, la cantidad de muestras por segundo que eran capaces de procesar en tiempo real, permitió que estas diferencias fueran cada vez más imperceptibles, y que la señal resultante fuese indistinguible de la original, con el añadido de la desaparición de los problemas físicos asociados con las tecnologías analógicas, como la distorsión o el ruido de fondo, que en ocasiones podían resultar indeseados. Sin embargo, en aplicaciones musicales, una ausencia de ruido puede resultar excesivamente estéril y los medios silenciosos pueden revelar los defectos en el audio introducidos por los músicos durante la manipulación de sus instrumentos musicales y amplificadores (Katz, 2015).

Un exceso de precisión en la representación sonora puede generar una abstracción del medio material y la tendencia a reintroducir elementos físicos con el objetivo de superar la inmaterialidad del audio digital se ha convertido en un tema recurrente en los últimos años. Estas ideas se han combinado con una creciente tendencia de amplia teoría social y política que lleva a muchas personas a posicionarse como antagonistas o críticos respecto a la teoría digital (Hugill, 2018). Ante este fenómeno social, surge el término "post-digital", que recoge estas formas de pensar. Este término fue introducido en el año 2000, en un famoso artículo de Kim Cascone donde afirmó lo siguiente:

La estética post-digital fue desarrollada en parte como resultado de la experiencia inmersiva de trabajar en un ambiente lleno de tecnología digital: el zumbido de los ventiladores de los ordenadores, las impresoras láser produciendo documentos, la sonificación de las interfaces de usuario y el tenue sonido de los discos duros. Pero más específicamente, debido a los defectos en la tecnología digital que surgieron con esta nueva forma de trabajo: *glitches*, *bugs*, errores de programa, caídas del sistema, clipeados, *aliasing*, distorsión, ruido de cuantificación, o incluso el ruido de fondo de las tarjetas de sonido resultaron materiales que los compositores buscaron incorporar en su música. (Cascone, 2000)

El énfasis post-digital puesto en los errores de las cualidades físicas del medio ha resultado ser una nueva manera de entender la naturaleza eléctrica de la circuitería digital

e incluso una exageración de los aspectos físicos de este medio. Estos defectos o errores digitales fueron motivo de interés para algunos compositores en la década de los 90, que comenzaron a usarlos a modo de percusión. Esta idea puede extenderse al ruido de la superficie de un vinilo o al sonido producido al tocar accidentalmente el conector *jack* de una guitarra. Como resultado, nació el género *glitch*, cuyo nombre hace referencia a estos errores (Hugill, 2018).

Gracias a la versatilidad de cálculo que permite el dominio digital en cuanto a la modificación sonora, junto a las exponenciales mejoras en la velocidad y el rendimiento de los procesadores, hoy en día es posible introducir artificialmente y en tiempo real estas cualidades físicas del medio digital, emular las cualidades físicas del medio analógico o incluso mediante el modelado físico, simular estos medios con el objetivo de crear una aproximación realista. Estas simulaciones son un tipo de síntesis sustractiva, ya que se basan en filtrar una fuente sonora que inicialmente es rica en espectro frecuencial y a la que se le sustrae parte de su contenido. Ya que los músicos a partir de los 90 empezaron a mostrar interés por esta síntesis sustractiva y la calidez sonora de sus resultados, la empresa sueca Clavia introdujo en 1995 el sintetizador NordLead, que generaba sonidos mediante síntesis sustractiva pero empleando herramientas de procesado digital. Éste fue el primer sintetizador digital en emular el sonido de la primera generación de sintetizadores y junto a su lanzamiento apareció por primera vez el término “analógico virtual” para representar la simulación digital de los equipos de audio analógicos (Pekonen y col., 2011).

En los últimos años, las investigaciones en síntesis y efectos analógicos virtuales han aumentado drásticamente, sobre todo en torno a 2010 (Pekonen y col., 2011), en correlación con una tendencia general por recuperar la estética de las diferentes décadas del siglo XX, ya sea por un sentimiento de nostalgia por épocas vividas o por el interés de las nuevas generaciones de artistas (impulsado por el atemporal y fácil acceso a la música que ofrece internet) por transmitir el descontento con esta época, protagonizada por un periodo de crisis económica y política, donde cada vez más personas perciben las nuevas tecnologías como deshumanizadas y estériles, y que bajo un contexto socio-económico capitalista generan más problemas psico-sociales de los que intentan solucionar.

1.2 Objetivos

El principal objetivo de este proyecto es el de la creación de herramientas de usuario para la aplicación de efectos digitales en tiempo real en forma de prototipo de *plugin* cuya futura compilación pueda conectarse y ejecutarse de forma compatible con la mayoría de entornos de trabajo de audio digital.

Gracias al procesado por bloques en tiempo real, el usuario podrá modificar los parámetros de cada efecto y comprobar cómo estos afectan al resultado sonoro final a partir

de un fichero de entrada que actuará de *sample* con el que el usuario pueda realizar composiciones derivadas aplicando estos efectos de forma interpretativa. Para ello, el tamaño de cada bloque de muestras durante el procesado será mínimo para reducir la latencia y que de esta forma el usuario pueda sincronizar la manipulación de los parámetros con el posible tempo del fichero musical de entrada. Estos efectos pretenden ser de tipo analógico virtual, ya que emularán el sonido de dispositivos analógicos como el famoso filtro musical Moog Ladder cuya implementación se basa en un modelo digital simplificado y que consigue imitar las características espectrales de forma aproximada. A parte del diseño y prototipado de este filtro analógico virtual, se emulará el efecto de eco basado en cinta magnética, cuya implementación, a pesar de no emplear técnicas de modelado digital para generar una simulación realista, se incluirá en un paquete que contenga otros efectos cuya ejecución simultánea introduzcan características no lineales de los sistemas analógicos como la saturación no lineal, el ruido de fondo incorrelado o las variaciones de tono oscilantes introducidas por los defectos e irregularidades en la oscilación de las bobinas de cinta y de los cabezales. También se incluirá en este paquete multi-efectos una implementación digital del efecto *flanger* cuyo origen se sitúa junto a la aparición del magnetófono y cuya estética sonora encaja con el resto de efectos que podrán ser usados por el usuario a través de un prototipo de *plugin* multi-efectos.

1.3 Estructura del documento

En primer lugar, el apartado 2 describe las herramientas utilizadas para el desarrollo del proyecto.

A continuación, la descripción del desarrollo mediante el diseño e implementación de los efectos se divide en dos apartados cuyos efectos se incluyen en forma de prototipo de *plugin* cuyo funcionamiento es independiente. Estos son los apartados 3 y 4.

El apartado 3 describe el diseño y prototipado de un *plugin* que aplica en tiempo real un efecto de filtrado virtual analógico basado en un modelo digital del filtro analógico Moog Ladder. Este apartado describe tanto el modelo analógico original, como los diferentes modelos digitales propuestos para su simulación, incluyendo el modelo simplificado con características no lineales en el que se basa la adaptación para su ejecución tiempo real de la implementación propuesta.

El apartado 4 describe el diseño y prototipado de un *plugin* multi-efectos que incluye una serie de efectos que en su conjunto emulan el sonido de los equipos de eco de cinta, incluyendo un efecto de saturación no lineal, una etapa de ruido filtrado, un efecto vibrato que emula el efecto *wow* de la cinta magnética, así como un efecto *flanger* cuya ejecución es alternativa a la del vibrato. Todos estos efectos podrán ser activados o desactivados de forma independiente a través del prototipo de interfaz de usuario propuesto.

En penúltimo lugar, en el apartado 5 se describirán una serie de mejoras y futuras líneas de trabajo propuestas para la continuación de este proyecto.

Por último, en el apartado 6 se presentan una serie de conclusiones críticas extraídas del desarrollo de este proyecto.

Para la comprobación de las fuentes en las que se han basado los párrafos y fragmentos referenciados, el apartado 7 incluye un listado de referencias que constituye la bibliografía de este documento.

Adicionalmente, al final de este documento (con el objetivo de ofrecer a modo páginas de consulta una serie de definiciones de acrónimos y términos para los lectores que puedan no estar habituados a la terminología técnica empleada), se incluye el apéndice A para definiciones de acrónimos, y el apéndice B para definiciones de términos técnicos.

2 Herramientas de trabajo

Para el desarrollo de este proyecto se ha empleado el entorno de programación MATLAB 2018a ejecutado en un equipo bajo el Sistema Operativo Windows 10 del laboratorio de Tratamiento Digital de la Señal (B118) de la Escuela Politécnica Superior de Gandía.

La implementación de los efectos se ha realizado con la ayuda de las herramientas proporcionadas por el paquete Audio Toolbox (MathWorks, 2016) que incluye MATLAB. Gracias a las herramientas incluidas en este paquete y a la posibilidad de una programación orientada a objeto, se ha empleado el objeto de clase *plugin*, en cuya función de procesado se han integrado los algoritmos de cada efecto y que facilita y optimizan su ejecución en tiempo real, permitiendo además, a través de una interfaz, manipular los parámetros de cada *plugin* y escuchar las modificaciones sonoras resultantes en tiempo real.

Para el prototipado de cada Interfaz Gráfica de Usuario (GUI) se ha empleado la herramienta Audio Testbench incluida en el mismo paquete Audio Toolbox y que permite una rápida comprobación del funcionamiento de cada procesado sin necesidad de la programación de una interfaz gráfica de usuario propia, ya que proporciona una serie de controles gráficos que permiten manipular los parámetros de la interfaz del *plugin* y ejecutar cada efecto en tiempo real.

El análisis auditivo de los resultados se ha realizado mediante el uso de unos auriculares de la marca y modelo Sennheiser HD 449.

3 Diseño y prototipado de un filtro analógico virtual

3.1 El filtro *Moog Ladder*

En 1965 el Dr. Robert Moog propuso con su trabajo publicado en el artículo (Moog, 1965) un filtro paso-bajo analógico para sus diseños de sintetizadores que acabaría convirtiéndose en una de las herramientas más famosas e influyentes en la música electrónica, sobre todo al ser conocido como el primer filtro musical. A día de hoy, los músicos siguen apreciando la calidez sonora generada por este filtro gracias a las funciones no lineales presentes en su diseño compuesto por transistores.

Este filtro está compuesto por cuatro secciones que constituyen cada una un filtro de transistores de un polo único. Estas secciones quedan conectadas en entre sí y de forma global mediante un bucle de realimentación negativa. La estructura de este filtro recibe el nombre de “escalera” (del inglés *Ladder*) por la forma en que las secciones están conectadas en cascada. Este filtro incluye un factor de ganancia k , que puede variar de 0 a 4. Este rango se debe a que la ganancia en la frecuencia de corte de cada sección del filtro es de $1/4$. Cuando $k = 4$ el filtro auto-oscila a la frecuencia de corte y mediante el control de la ganancia k este filtro permite controlar la resonancia, es decir la ganancia de este pico a la frecuencia de corte debido a la oscilación.

La Figura 1 muestra el diseño del circuito de transistores que constituyen la estructura de filtros en escalera, cuyo par de transistores diferenciales situados en la parte inferior del circuito quedan impulsados por la corriente de control I_{ctl} y el voltaje de entrada ΔV_4 , multiplicado por el factor k con el que se manipula la retroalimentación negativa y con ello aumenta la resonancia del filtro. En la figura 2 pueden observarse las mediciones realizadas por Stilson y Smith (Stilson y col., 1996) que muestran la respuesta de este filtro para diferentes valores de ganancia k , donde se observa un aumento de la resonancia en la frecuencia de corte a mayor ganancia.

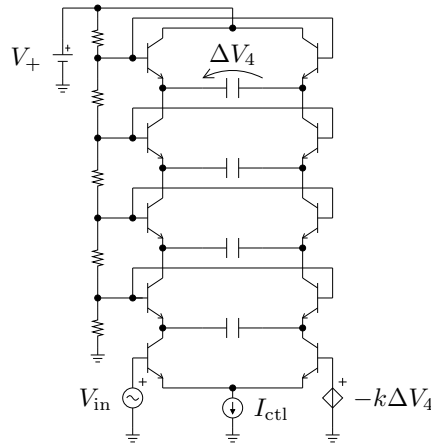


Figura 1: Circuito Moog Ladder (Moog, 1965)

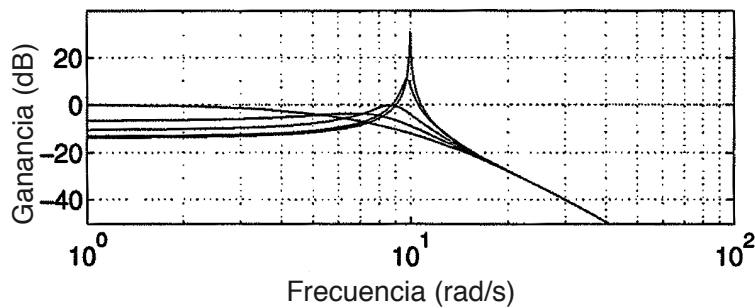


Figura 2: Respuesta en frecuencia del filtro Moog para diferentes valores de ganancia de retroalimentación $k = [0, 0.3, 0.6, 0.9, 0.99]$ (Stilson y col., 1996)

Ya que el objetivo es imitar la respuesta en frecuencia de este circuito para obtener un sonido que emule la calidez de este filtrado paso-bajo, es necesario recurrir a un modelo equivalente en el dominio digital.

3.2 Modelos digitales del filtro *Moog Ladder*

El primer modelo digital que imitase la respuesta en frecuencia de este filtro fue propuesto en 1996 por Tim Stilson y Julius O. Smith, en cuya publicación (Stilson y col., 1996) consideraron varios métodos de cálculo para convertir el filtro *Moog Ladder* en un equivalente digital. Su estructura resultante queda representada en la Figura 3, en la que se pueden observar 4 filtros de un polo con respuesta en frecuencia según la Ecuación 1 conectados en cascada y realimentados negativamente añadiendo una unidad extra de retardo, por lo que el filtro resultante sería de quinto orden y la respuesta global quedaría definida por la Ecuación(2).

$$G_1(s) = \frac{1}{1 + s/\omega_c} \quad (1)$$

$$H(s) = \frac{1}{k + (1 + s/\omega_c)^4} \quad (2)$$

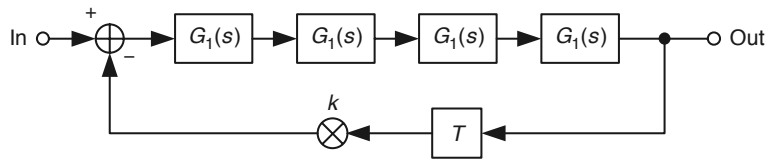


Figura 3: Modelo digital del filtro Moog (Stilson y col., 1996)

Para la discretización de este diseño Stilson y Smith propusieron diferentes métodos de cálculo basados en la transformada bilineal y diferencial. Sin embargo estos métodos presentan dificultades en cuanto al control de la frecuencia de corte y la precisión del filtrado, que se verían comprometidos para altos valores de frecuencia de corte y resonancia, además de la incapacidad del filtro de auto-oscilar (Välimäki y Huovilainen, 2006)

Posteriormente a lo largo de los años varias correcciones y mejoras de este modelo fueron propuestas, pero una de ellas, desarrollada por Antti Huovilainen (Huovilainen, 2004) introdujo una implementación digital del comportamiento no lineal de los transistores implicados en el diseño analógico. El modelo contiene cinco funciones no lineales en un bucle de realimentación, una por cada sección de filtrado de primer orden y una por la realimentación. Se descubrió que era necesario un sobremuestreo de al menos un factor 2 para reducir los problemas de aliasing cuando las funciones no lineales eran usadas (Huovilainen, 2004). Esta estructura auto-oscila cuando la ganancia de la realimentación es 1.0 y este valor puede ser mayor, ya que las funciones no lineales impiden que el sistema se vuelva inestable. Este modelo mejorado representaba una simulación más próxima a las características sonoras del circuito original de Moog. La desventaja sería la necesidad de usar cinco funciones de tipo tangente hiperbólica (*tanh*) lo cual incrementaría el coste computacional y dificultaría su implementación para aplicaciones en tiempo real. Ante este inconveniente, en un posterior trabajo de Huovilainen junto a Välimäki se desarrolló un modelo extendido alternativo (Välimäki y Huovilainen, 2006) cuya estructura puede observarse en el esquema de la Figura 4, en la que las múltiples funciones no lineales incluidas en cada sección son sustituidas por una sola función no lineal. Para ello usaron la función *tanh*, aunque cualquier otra función que produzca una saturación suave puede ser usada. Aunque existen diferencias en el sonido comparando con la versión completa del modelo digital no lineal (Huovilainen, 2004), este nuevo modelo puede emular la mayor parte del comportamiento, como es el caso de la auto-oscilación (Välimäki y Huovilainen, 2006).

Una comparación entre el modelo lineal (Stilson y col., 1996), el modelo no lineal completo (Huovilainen, 2004) y el modelo no lineal simplificado (Välimäki y Huovilainen, 2006) puede observarse en la Figura 5, cuyos resultados fueron obtenidos tras pasar por cada modelo

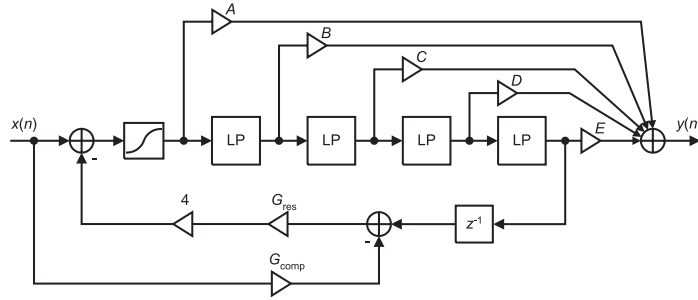


Figura 4: Modelo digital mejorado del filtro *Moog* (Välämäki y Huovilainen, 2006)

una señal de tipo diente de sierra a modo de demostración, que fue publicada en 2006 en la revista *Computer Music Journal DVD* y cuyos resultados han sido extraídos del libro *DAFX* (Zölzer, 2011).

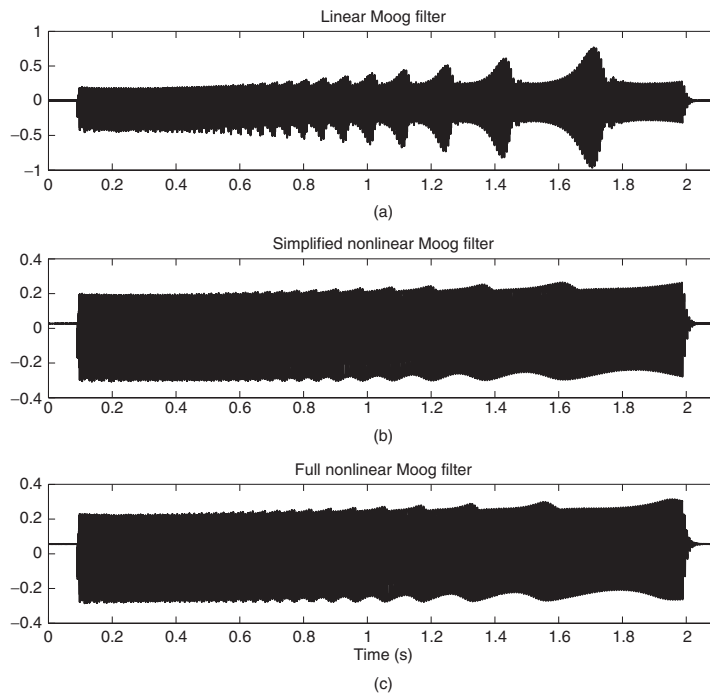


Figura 5: Señal dientes de sierra filtrada con los tres modelos digitales de filtro *Moog*: (a) el modelo lineal *Stilson-Smith* de la Figura 3, (b) el modelo no lineal simplificado de la Figura 4, y (c) el modelo completo no lineal propuesto en (Huovilainen, 2004)

3.3 Adaptación del modelo no lineal simplificado para tiempo real

Para la implementación de este filtro y su ejecución en tiempo real, esta parte del trabajo ha sido desarrollada en el entorno MATLAB y está basada en el modelo simplificado propuesto en la segunda edición del libro DAFX (Zölzer, 2011), cuyo esquema queda representado en la Figura 6 y se basa a su vez en el modelo propuesto en (Välimäki y Huovilainen, 2006).

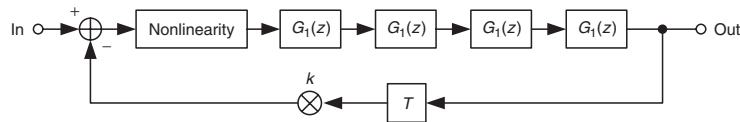


Figura 6: Estructura simplificada de filtro Moog no lineal (Zölzer, 2011)

Como ya se ha demostrado en el apartado anterior, este modelo difiere en cuanto características sonoras del modelo digital no lineal completo (Huovilainen, 2004) pero introduce una auto-oscilación en torno a la frecuencia de corte del filtro cuando el valor de resonancia se aproxima a 1, y esta frecuencia de resonancia disminuye gradualmente a lo largo del tiempo. Además, el nivel de la señal aumenta cada vez que la frecuencia de resonancia coincide con uno de los armónicos de la señal, sin embargo la compresión producida por el uso de la función no lineal \tanh disminuye estas variaciones de amplitud. El algoritmo original en el que se ha basado esta parte del proyecto puede consultarse en el fragmento de Código 1.

Código 1: Algoritmo filtro Moog no lineal simplificado (Zölzer, 2011)

```

1 function [out2,out2,in2,g,h0,h1] = moogvcf(in,fc,res)
2   % Autores: Välimäki, Bilbao, Smith, Abel, Pakarinen, Berners
3   % Parámetros:
4   % in = señal de entrada
5   % fc= frecuencia de corte (Hz)
6   % res = resonancia (0...1 o mayor para auto-oscilación)
7   fs = 44100; % Frecuencia de muestreo
8   fs2 = 2*fs; % Frecuencia de muestreo interna
9   % Señal de entrada sobremuestreada x2:
10  in = in(:); in2 = zeros(1,2*length(in)); in2(1:2:end) = in;
11  % Filtro Anti-imaging
12  h = fir1(10,0.5); in2 = filter(h,1,in2);
13  Gcomp = 0.5; % Ganancia de compensación
14  g = 2*pi*fc/fs2; % Retroalimentación del IIR a fs2
15  Gres = res; % Mapeado directo
16  h0 = g/1.3; h1 = g*0.3/1.3; % FIR con ganancia g
17  w = [0 0 0 0 0]; % Variable de 5 estados
18  wold = [0 0 0 0 0]; % Estados previos (1 unidad de retardo)
19  out = zeros(size(in)); out2 = zeros(size(in2));
20  for n = 1:length(in2),
21    % Entrada y retroalimentación
22    u = in2(n) - 4*Gres*(wold(5) - Gcomp*in2(n));

```

3.3 Adaptación del modelo no lineal simplificado para tiempo real

```
23 w(1) = tanh(u); % Saturación no lineal
24 w(2) = h0*w(1) + h1*wold(1) + (1-g)*wold(2); % Primer IIR1
25 w(3) = h0*w(2) + h1*wold(2) + (1-g)*wold(3); % Segundo IIR1
26 w(4) = h0*w(3) + h1*wold(3) + (1-g)*wold(4); % Tercer IIR1
27 w(5) = h0*w(4) + h1*wold(4) + (1-g)*wold(5); % Cuarto IIR1
28 out2(n) = w(5); % Salida del filtro
29 wold = w; % Desplazamiento en 1 unidad de retardo
30 end
31 out2 = filter(h,1,out2); % Filtro anti-aliasing a fs2
32 out =out2(1:2:end); % Diezmado factor 2 (vuelta a la fs original)
```

Para la ejecución de este algoritmo en tiempo real se han realizado varias modificaciones para adaptar este código a un procesado por bloques de un tamaño de 1024 muestras.

En primer lugar, se ha duplicado el número de canales de entrada y salida para que el filtrado se realice sobre una señal estéreo.

En segundo lugar, se ha modificado el método de filtrado, pues se ha observado que el empleo de la función `filter()` introduce clicks al final de cada bloque. Para solucionar este problema y mejorar la eficiencia se ha implementado un filtrado en dominio frecuencial creando en primer lugar dos búferes por canal: uno para guardar el bloque presente, y otro para guardar el bloque anterior. El filtrado se realiza transformando el bloque de datos al dominio frecuencial mediante el uso de la Transformada Rápida de Fourier (FFT) (función `fft()` en MATLAB), multiplicándolo por los coeficientes de la Transformada Discreta de Fourier (DFT) del filtro, también transformada al dominio frecuencial para seguidamente realizar la Transformada Inversa Rápida de Fourier (IFFT) (función `ifft()` en MATLAB) y mediante el método solapamiento y almacenamiento poder recuperar la señal filtrada en tiempo real sin la aparición de clicks, ya que gracias al uso de los búferes es posible solapar la cola del bloque anterior (formada por las muestras que exceden el tamaño del bloque como resultado del filtrado y cuyo tamaño es igual al de la respuesta al impulso del filtro mas una muestra) al inicio del bloque presente. De esta forma el filtrado se realiza disminuyendo el coste computacional y desaparecen los clicks y la reproducción resulta más fluída. Este proceso se realiza tanto a la entrada del bucle que realiza los procesos de saturación no lineal y filtrado múltiple en cascadada, como a la salida del mismo, es decir, en las etapas de filtrado *anti-imaging* y *anti-aliasing*, necesarias en este caso debido al sobremuestreo factor 2 en el bucle de filtros en escalera.

Realizando estas modificaciones al código original (Código 1), la ejecución en tiempo real funciona sin problemas, y no es necesario el uso de una Tabla de Consulta (LUT) para el mapeado de la función no lineal *tanh*, tal y como se propone en (Zölzer, 2011).

El código resultante puede consultarse en el fichero (`moogladderRT.m`).

3.4 Prototipado de un *plugin* de filtrado en tiempo real

Para ejecutar el algoritmo descrito en la sección anterior y analizar su funcionamiento en tiempo real se ha hecho uso de las herramientas ofrecidas por el paquete Audio Toolbox de MATLAB (MathWorks, 2016), ya que, mediante la programación orientada a objeto, se permite construir un objeto de clase `audioPlugin`, cuya función central `process()` permite integrar el algoritmo de procesado. Este objeto requiere además definir los parámetros configurables por el usuario o bien parámetros de tipo privado o que dependan de otros parámetros. Además, permite construir una interfaz a partir de los parámetros configurables de forma que puedan ser accesibles desde la GUI y modificar sus valores para testear el funcionamiento del filtro en tiempo real.

En lugar de diseñar una GUI propia, se ha empleado la herramienta `AudioTestBench` que permite un rápido y sencillo prototipado generando una interfaz gráfica genérica con controles deslizantes para manipular los valores de los parámetros de frecuencia de corte y resonancia en tiempo real. A través de `AudioTestBench` se permite configurar los objetos de lectura y escritura antes de la ejecución del programa para cambiar el tamaño de bloque y establecerlo, en el caso de este diseño, a un valor de 1024 muestras, y la frecuencia de muestreo en función de la del archivo de entrada.

Además de los parámetros de frecuencia de corte y resonancia, en la interfaz se ha incluido un parámetro de tipo lógico que puede ser modificado desde la interfaz gráfica a través de una casilla que al marcarla activará el funcionamiento del filtro a modo de *passthrough* y al desmarcarla, bajopaseará el filtro para comparar de forma auditiva la diferencia con la entrada sin filtrar.

La interfaz del prototipo generada por `AudioTestBench` puede observarse en la Figura 7, y el código del *plugin* está presente en el fichero (`filtroEscalera.m`).

3.5 Análisis frecuencial del filtro resultante

Para la caracterización de este filtro y comprobación de su funcionamiento, éste ha sido excitado por un entrada de ruido blanco gaussiano cuya respuesta plana en frecuencia (Figura 8) caracterizará la respuesta del filtro. El nivel de este ruido se mantendrá constante en todas las pruebas, con un valor de -20dBm.

En primer lugar se comprueba el funcionamiento del filtro para una frecuencia de corte de 1 kHz y un valor de resonancia de 0. La respuesta en frecuencia resultante queda representada en la Figura 9, donde puede comprobarse el incremento de la atenuación a partir de la frecuencia de corte.

En segundo lugar se comprueba el funcionamiento de la auto-oscilación con un valor de resonancia 1. La respuesta en frecuencia resultante queda representada en la Figura 10, en la que se observa un considerable aumento de la ganancia en torno a

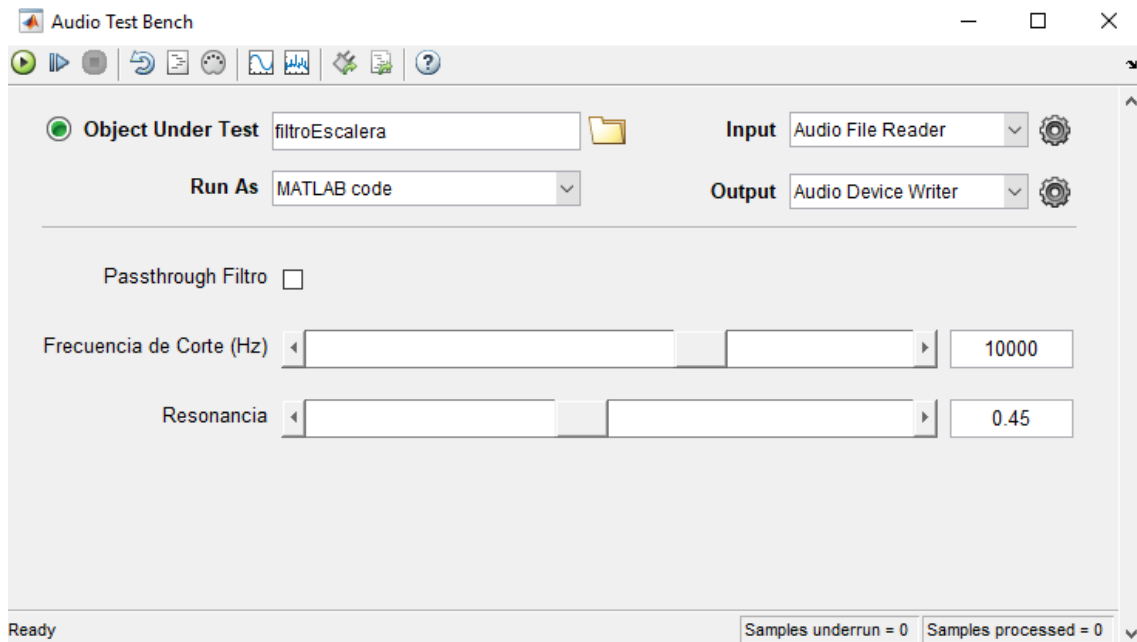


Figura 7: Interfaz gráfica del plugin de filtrado en escalera generada por AudioTestBench

la frecuencia de corte como consecuencia del efecto de resonancia provocada por la auto-oscilación del filtro.

Respecto a la percepción auditiva, el filtrado realiza este efecto de calidez sonora como consecuencia de la pérdida de frecuencias agudas y que recuerda al sonido analógico. Se ha observado además, que para valores de resonancia por encima de aproximadamente 0.85 se percibe un armónico con suficiente amplitud como para generar un pitido que para valores cercanos a 1 resulta desagradable al oído. Sin embargo este tono provocado por la auto-oscilación, a valores moderados de resonancia genera un efecto metálico que recuerda a las grabaciones analógicas y compensa la percepción de pérdida de nitidez.

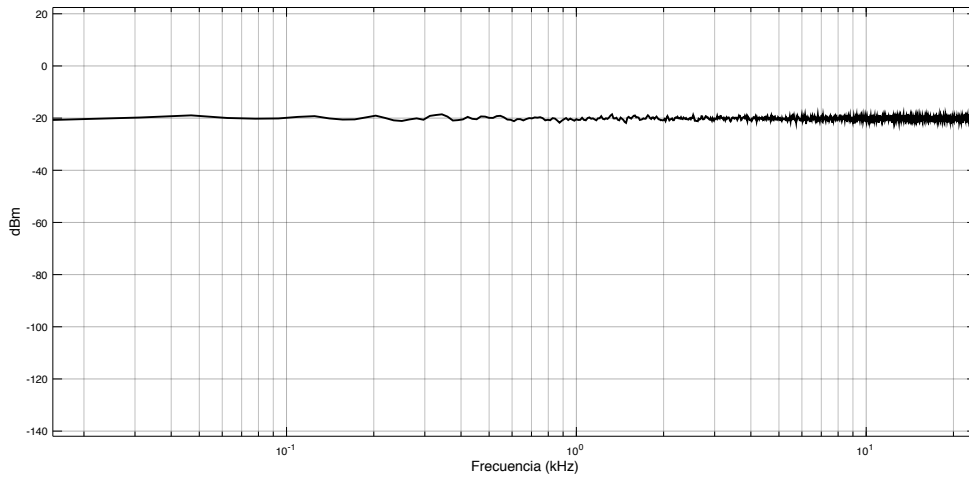


Figura 8: Respuesta en frecuencia del ruido blanco gaussiano a la entrada del filtro

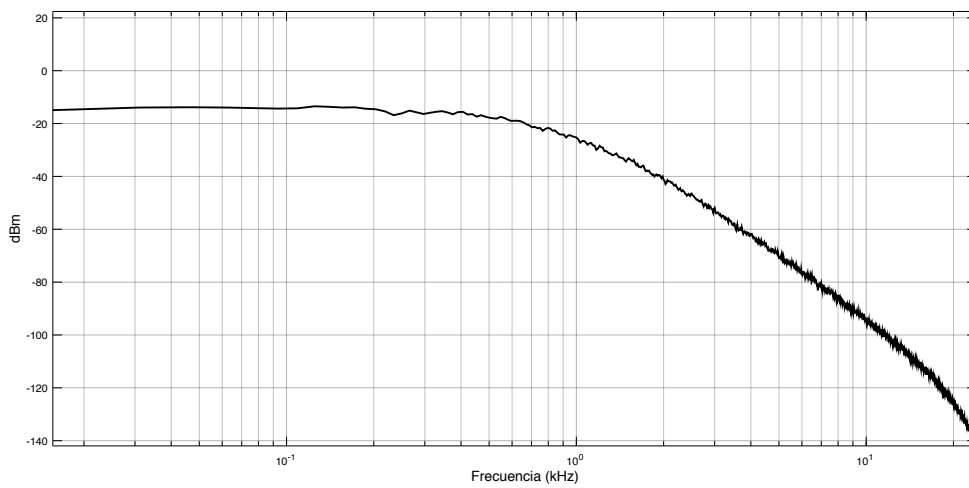


Figura 9: Respuesta en frecuencia del filtro para una frecuencia de corte de 1kHz y un valor 0 de resonancia

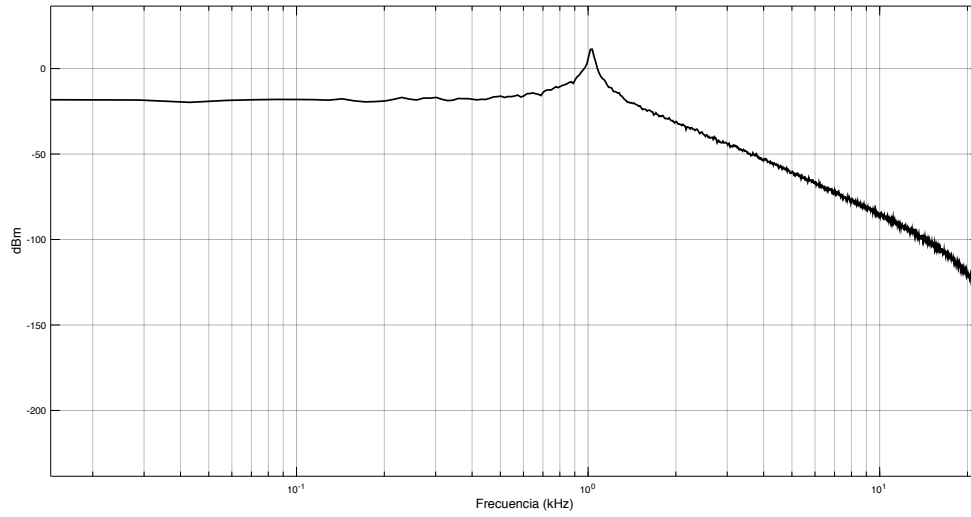


Figura 10: Respuesta en frecuencia del filtro para una frecuencia de corte de 1kHz y un valor 1 resonancia

4 Diseño y prototipado de un *plugin* multi-efectos

4.1 Introducción y aspectos sobre el diseño

Esta segunda parte del proyecto consiste en el diseño de un *plugin* adicional e independiente del *plugin* de filtrado en escalera desarrollado en la sección 3.4. Este *plugin* incluye efectos y emulaciones inspirados en equipos analógicos como el eco de cinta magnética o el efecto *flanger*, que proporcionan al sonido una estética de estilo espacial, futurista o de ciencia ficción, que junto a las emulaciones analógicas se combinan para generar una estética sonora conjunta que podría denominarse como retro-futurista.

En primer lugar, se describe el desarrollo de un eco digital simple con realimentación de ganancia controlable y en cuya línea de retardo se introducirá un efecto que emula la saturación de cinta magnética que afecte únicamente al eco y se implemente mediante el uso de la función no lineal $\text{atan}(\alpha)$, donde α será el factor de saturación. A la salida se añadirá una etapa de ruido gaussiano filtrado que emule el ruido de fondo o ruido *hiss* de los equipos analógicos, pudiendo el usuario ajustar su respuesta en frecuencia mediante un filtro paso-bajo. A continuación, la salida podrá pasar opcionalmente por una de las dos tipos de modulaciones basadas en retardo variable propuestas: el *vibrato* o el *flanger*. Este primer efecto, a una baja frecuencia y profundidad pretende emular las variaciones de tono provocadas por los defectos en los aparatos de cinta magnética, cuyas oscilaciones irregulares en sus partes móviles producen en la escucha un sutil desafinado periódico. Respecto al *flanger*, podrá ser seleccionado alternativamente al *vibrato*, pero no simultáneamente dada su naturaleza incompatible, pues su ejecución simultánea provoca que el efecto *vibrato* quede enmascarado por el efecto *flanger*.

Opcionalmente, este *plugin* podrá conectarse en cascada junto al *plugin* de filtrado en escalera desarrollado anteriormente, sin embargo, la complejidad de los cálculos que ambos diseños requieren dificultan su ejecución bajo un mismo bucle de procesado por bloques y provocan un exceso de latencia que genera pérdidas de muestras en la reproducción (*underruns*) debido a que la velocidad de escritura es superior al tiempo necesario para que cada bloque pase por dos algoritmos de procesado que requieran de un bucle que opere muestra a muestra, es decir, tanto la etapa de saturación no lineal del filtro en escalera como los efectos basados en retardo variable (*vibrato* o *flanger*) resultan incompatibles en tiempo real en una misma clase *plugin* de MATLAB. La Figura 11 muestra el diseño global propuesto y representado a través de un esquema.

Adicionalmente, el diseño incluirá un control *Wet/Dry* cuyo valor, de 0 a 1, ajustado desde la interfaz, permitirá al usuario mezclar la línea de eco (con o sin los efectos de ruido y saturación) con la entrada limpia, y así poder controlar la fracción de estos efectos a la entrada de la etapa de modulación de retardo (*vibrato* o *flanger*), o bien a la salida si esta etapa no ha sido activada.

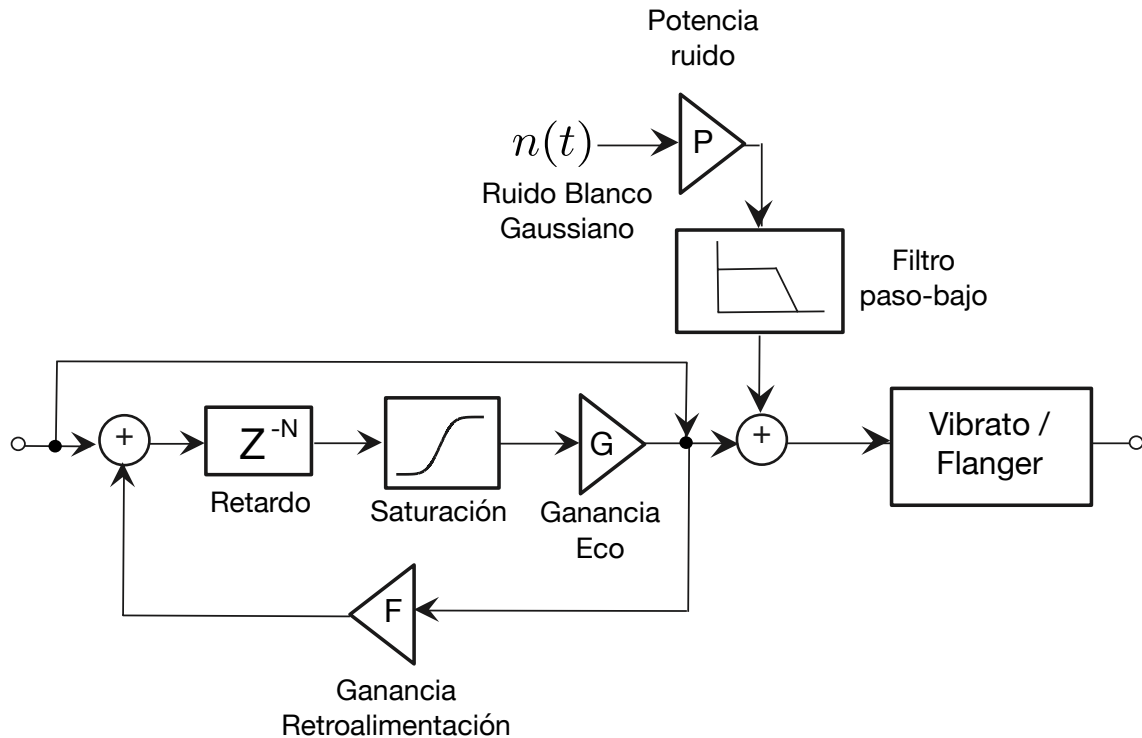


Figura 11: Estructura del diseño propuesto de plugin multi-efectos

4.2 Efecto de eco simple con retroalimentación

En primer lugar, se ha desarrollado un algoritmo de eco simple con retroalimentación basado en la mezcla de la señal de entrada con la propia señal retardada en función del parámetro de tiempo de retardo, configurable en tiempo real desde la GUI y medido en segundos. Para obtener esta versión de la entrada retardada se ha hecho uso de un objeto incluido en el paquete Audio Toolbox de MATLAB MathWorks, 2016 llamado `audioexample.DelayFilter` que realiza la tarea de retardo mediante el uso de un filtro digital y permite además controlar la cantidad de retroalimentación de este retardo mediante el parámetro `FeedbackLevel`, cuyo valor por defecto se establece en 0.25, ya que a partir de 0.5 esta realimentación provoca un aumento gradual en la amplitud de salida, pues al aumentar la ganancia del bucle de retroalimentación por encima de éste valor, las repeticiones del eco se incrementan multiplicando la amplitud final. Al aproximarse al valor de 1, el sistema se vuelve inestable y la amplitud de salida crece de forma exponencial, sin embargo, desde el diseño de la interfaz se ha limitado el rango de valores hasta llegar al 1 con el objetivo de dar la posibilidad al usuario de este efecto que puede resultarle interesante por motivos artísticos. Este factor de realimentación, en los equipos analógicos originarios, recibe el nombre de *Feedback* o *Repeats* y auditivamente, si el tiempo de retardo es lo suficientemente corto (por debajo de los 0.1 segundos) se aproxima al efecto de reverberación pues se trataría de una repetición de la misma entrada

retardada múltiples veces y a menos de 0.1 segundos de distancia entre repeticiones, el oído lo percibe como una reverberación, aunque puede resultar, en términos subjetivos, un tanto metálica y robótica. La estructura de este diseño propuesto puede observarse en el esquema de la Figura 12.

Adicionalmente, en la interfaz del *plugin* se incluye un control de ganancia de la línea de retardo para que el usuario pueda activar o desactivar este efecto de forma gradual y así mezclar el efecto de eco junto a las posibles repeticiones de la realimentación con la línea de entrada, y así controlar la proporción del efecto conjunto a la salida de esta etapa.

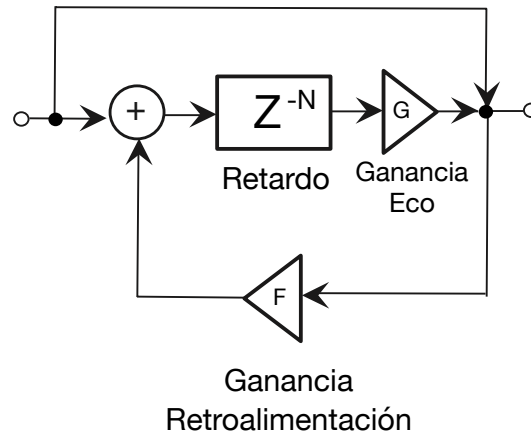


Figura 12: Estructura del eco simple con retroalimentación

4.3 Emulación de saturación de cinta magnética

En esta sección se describe la implementación de una etapa de saturación basada en el empleo de una función no lineal de tipo arcotangente que emule la saturación de los equipos de audio analógico basados en cinta magnética. Esta función ha sido escogida debido a que su forma se aproxima a la función obtenida a partir de la saturación característica de la cinta magnética obtenida por aproximación con el uso la función de error de Gauss, $M = erf(H)$, que elimina el efecto de histéresis magnética (H) en la función resultante que se muestra representada en la Figura 13 y obtenida en (Zölzer, 2011).

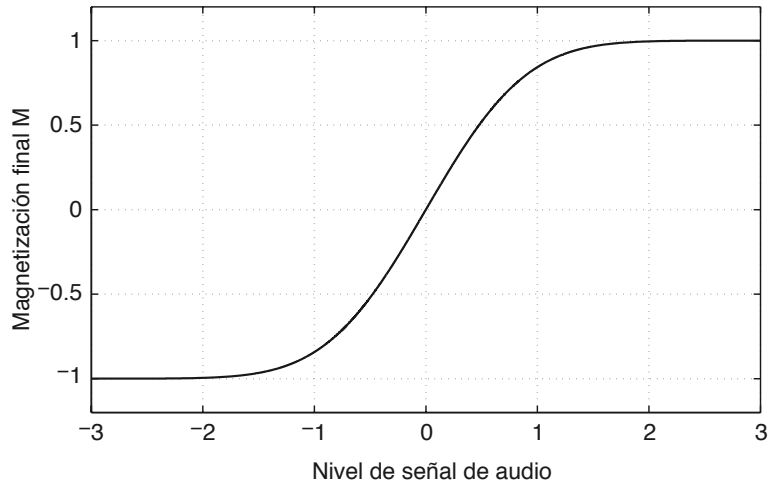


Figura 13: Aproximación de la función no lineal de la saturación de cinta magnética (Zölzer, 2011)

La implementación consiste en hacer pasar la entrada previamente retardada (x_{ret}) por el argumento de la función arcotangente, donde es multiplicada por un factor de saturación α que aumentará la amplitud e incrementará este efecto. La salida (y) obtenida a la salida de esta etapa viene dada por la Ecuación 3, y puesto que esta implementación está orientada para su ejecución en tiempo real, el procesado se realizará por bloques de 1024 muestras.

$$y = \frac{2}{\pi} \text{atan}(\alpha x_{ret}) \quad (3)$$

Para comprobar el funcionamiento de este procesado se introduce a la entrada de la etapa de retardo un tono puro de 500 Hz para una fácil visualización de este efecto. Este tono queda representado en la Figura 14 sin la etapa de saturación activada para su comparación tras pasar por esta etapa con un factor de saturación muy grande, de un valor arbitrario de 60, escogido únicamente a efectos visuales para la representación de la forma de onda resultante (Figura 15) pero que a efectos sonoros sería un valor excesivo.

Desde la GUI es posible ajustar el factor α desde un valor 1, que no representaría una ganancia en la salida, hasta un valor de 10, en el que habría un tope de ganancia a la salida, la cual se ha considerado suficiente para no provocar un efecto excesivo.

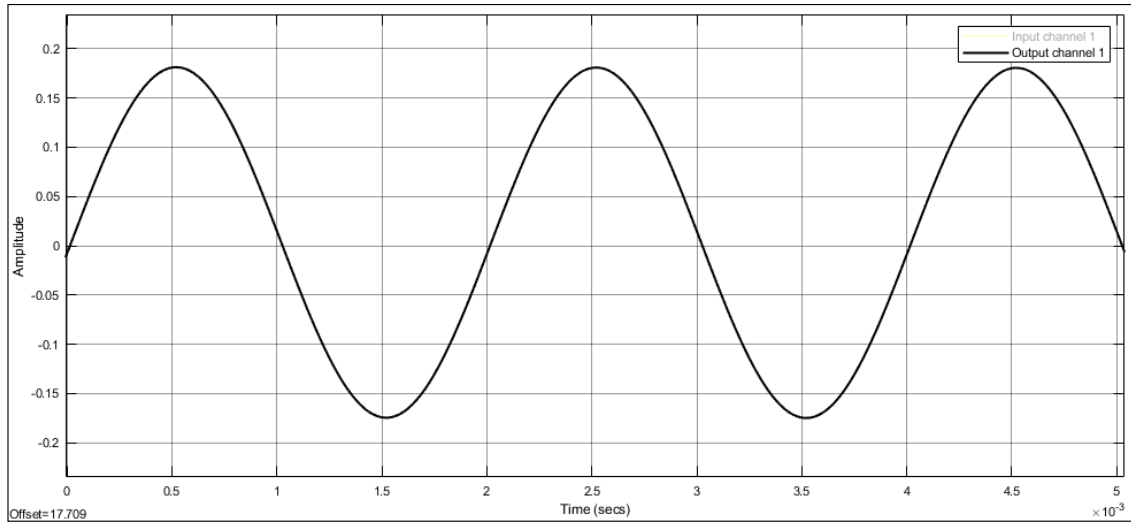


Figura 14: Tono puro de 500 Hz sin efecto de saturación

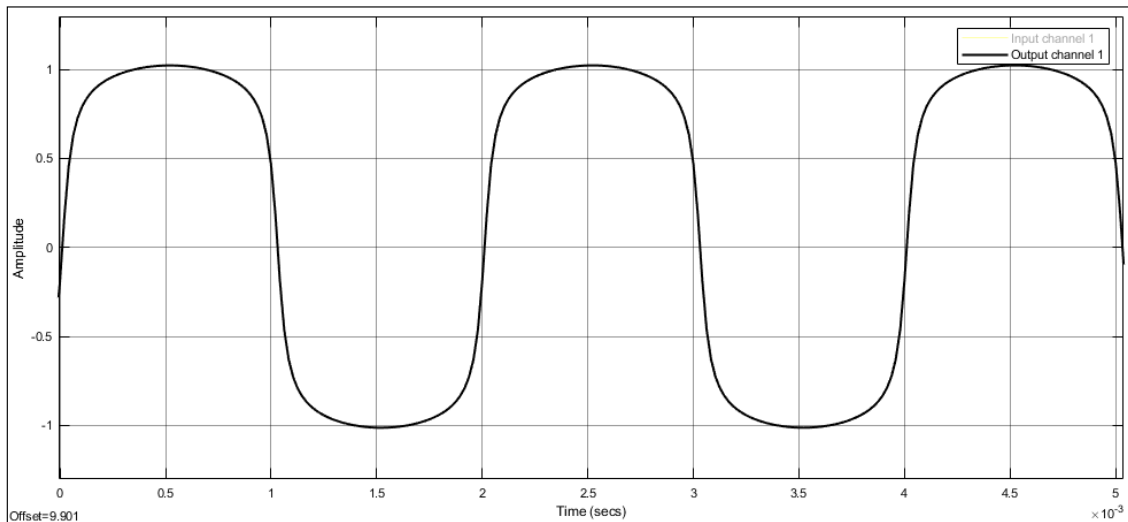


Figura 15: Tono puro de 500 Hz con un factor de saturación $\alpha = 60$

4.4 Emulación de ruido *hiss*

Uno de las características presentes en los equipos de audio antiguos desde los gramófonos, los tocadiscos a los equipos de grabación y reproducción basados en cinta magnética, es la presencia de degradaciones globales en forma de ruido de fondo estacionario que se aprecia sobre todo durante los silencios de la grabación y recibe el nombre de *hiss*. Este ruido, aunque técnica y objetivamente supone un defecto en los aparatos y en muchas ocasiones pueda resultar indeseado, puede resultar subjetivamente interesantes en aplicaciones musicales por motivos estéticos.

En esta parte del proyecto se procede a emular este ruido *hiss* en tiempo real mediante

el uso de una etapa de ruido blanco gaussiano filtrado con el uso de un filtro paso-bajo de tipo FIR y orden 10. Este ruido es añadido a la señal de salida del bucle de eco y saturación no lineal, detallados en los apartados 4.2 y 4.3, y de acuerdo con el esquema global propuesto y representado en la Figura 11. El ruido blanco gaussiano es generado por la función de MATLAB `wgn()`, que permite introducir como parámetros de entrada, a parte del tamaño del vector de salida, que en este caso es un bloque de 1024 muestras por cada canal de entrada, la potencia de ruido en unidades de dBW para una potencia de referencia de 1W.

Respecto al filtro, su respuesta en frecuencia se obtiene mediante el uso de la función de MATLAB `fir1()` y el filtrado es aplicado al bloque de entrada mediante el uso de la función `filter()`, que aunque introduzca clicks al final de cada bloque, estos resultan inapreciables para bajos niveles de potencia y debido al enmascaramiento del ruido, aunque en futuras mejoras de esta etapa, se sugiere el filtrado en el dominio frecuencial para una mayor eficiencia, y el método de solapamiento y almacenamiento para la eliminación de los clicks, tal y como se hizo en el efecto Moog Ladder (apartado 3.3).

En la interfaz del *plugin* se incluyen los parámetros de potencia de ruido y frecuencia de corte del filtro, con los que el usuario podrá ajustar sus valores mediante los controles deslizantes, con un rango de -50 dBW a -20 dBW en el caso de la potencia de ruido y de 100Hz a 15kHz en el caso de la frecuencia de corte del filtro paso-bajo. Además, antes de estos controles se ha incluído un parámetro de tipo *passthrough* para activar o desactivar esta etapa de ruido marcando o desmarcando una casilla en la GUI.

Para visualizar gráficamente el efecto de la adición de esta etapa, se ha usado el mismo tono de 500 Hz representado en la Figura 14 con un ruido añadido de una potencia muy elevada y de un valor de -20dBW y con una frecuencia de corte del filtro a 1 kHz. El resultado puede observarse en la Figura 16

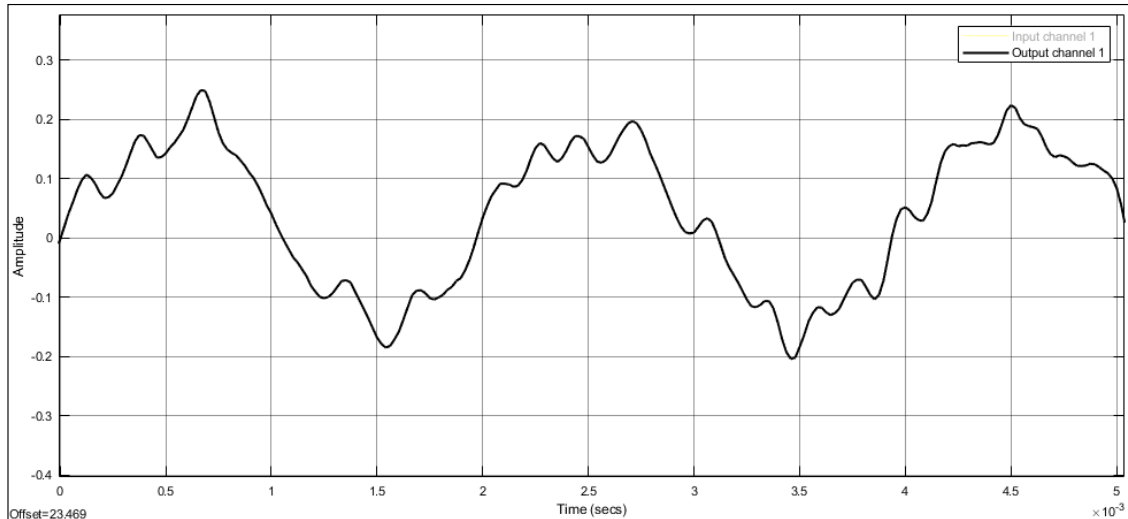


Figura 16: Tono de 500 Hz con un ruido añadido de -20dBW de potencia y filtrado a una frecuencia de corte de 1 kHz

A pesar de que la Figura 16 representa unos niveles de ruido no representativos de este tipo de ruido de fondo, ya que se añade con demasiada amplitud a la señal, para aplicaciones musicales que incluyan esta emulación de ruido tipo *hiss*, se recomienda una potencia de ruido de valores entre los -40dBW y los -30 dBW aproximadamente, aunque estos valores dependerán del nivel de la señal de audio y de la libre elección del usuario.

Respecto a la frecuencia de corte del filtro paso bajo, se recomienda un valor nominal de 1 kHz, para un ruido suave, cálido y agradable al oído, aunque esto dependerá de la subjetividad de cada usuario y del contenido frecuencial de la señal a la que se quiera añadir, por ello se ha dado libertad, a través de la interfaz del *plugin*, para filtrar desde los 100 Hz hasta los 15 kHz.

4.5 Efecto *Vibrato*

En esta sección se describe la implementación de un efecto de *vibrato* basado en un retardo variable de la señal controlado por una función senoidal cuya frecuencia es manipulable en tiempo real y actúa como moduladora del retardo. Como consecuencia de este retardo oscilante se produce un efecto Doppler que modifica periódicamente el tono de la señal de audio y cuya velocidad de variación viene dada por la frecuencia de la función moduladora f_m de acuerdo con la Ecuación 4, donde N representa la cantidad de retardo en número de muestras n y D representa la profundidad del *vibrato*, que viene dada por la amplitud de la señal moduladora.

La señal resultante a la salida ($y(n)$) viene dada por la Ecuación 5 y el esquema del

funcionamiento de este procesado puede observarse en la Figura 17.

$$N = D \sin(2\pi f_m n) \quad (4)$$

$$y(n) = x(n - N) \quad (5)$$

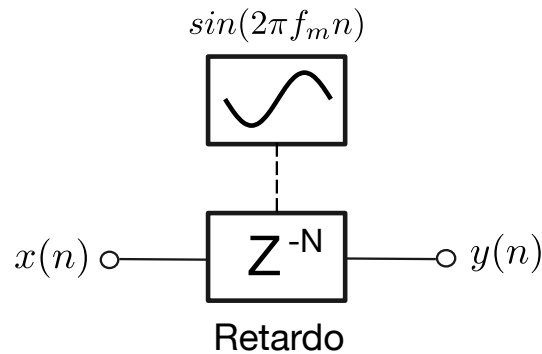


Figura 17: Esquema del efecto vibrato

El rango de valores propuestos, desde la interfaz del *plugin*, para la frecuencia del *vibrato* va de los 5 Hz a los 14 Hz mientras que la profundidad del mismo va desde los 0.2 ms hasta los 2 ms. Ya que el objetivo de este efecto es el de complementar a los anteriores añadiendo un efecto de emulación de la variación de tono que los equipos de cinta magnética introducen como defecto, para lograr esta emulación se recomiendan valores de frecuencia de los 5 Hz a los 7 Hz (aproximadamente) y una profundidad de *vibrato* con valores entre 0.2 y 0.5 ms con el objetivo de obtener un efecto sutil y con poca velocidad, de forma que imite los defectos en el giro de los discos de vinilo, en las bobinas de cinta magnética o las variaciones de velocidad angular en los cabezales, que son los transductores encargados de transformar la energía eléctrica en flujo magnético y viceversa. Este efecto también recibe el nombre de *wow*, sin embargo para un resultado más fiel al efecto original sería necesario el empleo de funciones que manipulasen de forma aleatoria la frecuencia y la amplitud de la función que modula el retardo, de acuerdo con la Ecuación 6, modelo desarrollado y propuesto en (Välimäki; González y col., 2008), donde $p(n)$, en este caso, es la función que varía el tono y que en el dominio digital controlará por tanto el retardo de muestras.

$$p(n) = 1 + A(n) \sin(2\pi f(n)n/f_s) \quad (6)$$

Además, dada la aleatoriedad de los valores de las funciones $A(n)$ y $f(n)$ es necesario un sobremuestreo e interpolado de la curva resultante para obtener un resultado mas suave y similar al efecto analógico original. Sin embargo, debido al incremento de la complejidad de cálculo que esto supondría y por tanto, el elevado coste computacional, para una aplicación en tiempo real resulta ineficiente y se ha considerado suficiente el

empleo de un *vibrato* sutil como alternativa que se asemeja al efecto *wow*. A parte de esto, el rango de valores de frecuencias y profundidades del *vibrato* se ha establecido de forma amplia para que el usuario tenga libertad de emplear este efecto para la finalidad estética que desee.

4.6 Efecto *flanger*

En este apartado se describe la implementación de un efecto *flanger* que se incluirá como alternativa al efecto *vibrato* en la interfaz del *plugin*, y cuyo funcionamiento se basa en sumar a la entrada una copia de la misma con retardo variable y basado en la misma técnica de retardo modulado que el *vibrato*. El efecto resultante consiste en un filtro de peine que respeta los armónicos y que se expande y se contrae de forma oscilante a lo largo del espectro frecuencial gracias al retardo variable controlado por una función senoidal.

Este efecto acústico puede ser comúnmente observado al aire libre cuando una avión pasa por encima del observador y el sonido directo se suma en su oído con el sonido reflejado en el suelo, con la resultante cancelación de ciertas frecuencias y la generación de un efecto de filtro de peine.

En la historia de la tecnología de grabación de audio, se descubrió por error al intentar grabar por duplicado empleando dos grabadoras magnetofónicas con ligeras diferencias en la velocidad de cada una y tras la reproducción simultánea. Tiempo después, este efecto pasó a realizarse empleando dos grabadoras de cinta, una de las cuales incluía un mecanismo de control de velocidad variable (Figura 18) (Bode, 1984).

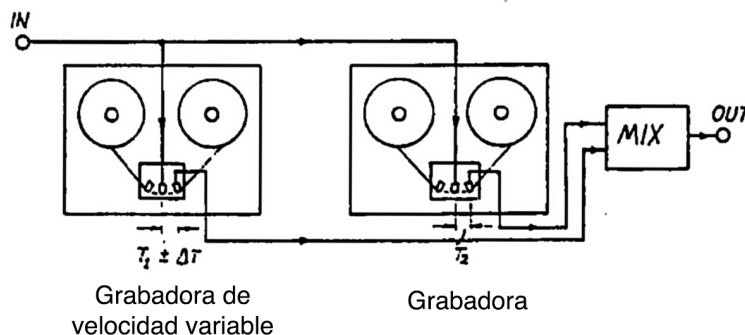


Figura 18: Obtención del efecto flanger con grabadoras magnetofónicas (Bode, 1984)

La implementación digital de este filtro incluye los ajustes de profundidad y frecuencia heredados del vibrato, y estos determinan tanto la velocidad de oscilación como la cantidad de atenuación del filtro de peine resultante. El retardo variable es controlado mediante la misma función que la del vibrato, de acuerdo con la Ecuación 4 y la salida ($y(n)$) se

obtiene en función de la entrada de acuerdo con la Ecuación 7, donde se añade un factor de ganancia g con valores que van de 0 a 1 y que incrementa o disminuye la cantidad de la señal retardada en la salida (y por tanto la proporción audible de este efecto). Este parámetro recibe el nombre de retroalimentación o *Feedback*.

El esquema del funcionamiento de este procesado queda representado en la Figura 19

$$y(n) = x(n) + gx(n - N) \quad (7)$$

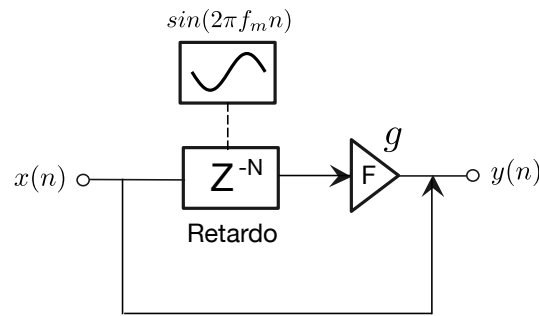


Figura 19: Esquema del funcionamiento del efecto flanger

Dado que este efecto recuerda al sonido del sobrevuelo de un avión, suele estar asociado a temáticas de ciencia ficción y provocan en la señal de salida una sensación de espacialidad y de movimiento oscilante, complementando el resto de efectos añadiendo un posible toque de estilo futurista que pueda resultar interesante al usuario por motivos estéticos o incluso prácticos, si por ejemplo se desea simular el vuelo de una nave o avión mediante la introducción de una señal de ruido a la entrada de este *plugin*. Es interesante además, combinar este efecto con la etapa de ruido para aplicaciones musicales, pues incrementa la percepción del efecto al percibir mejor la cancelación frecuencial gracias al ruido.

4.7 Prototipo de la interfaz obtenida con la herramienta Audio Testbench

En este apartado se describen las características de la interfaz global del *plugin* multi-efectos propuesto a través del prototipo generado por la herramienta Audio Testbench de MATLAB, incluida en el paquete Audio Toolbox (MathWorks, 2016).

Una vez integrados todos los efectos bajo el mismo bucle de procesado por bloques (`process()`) de la clase `plugin` y establecidos los parámetros configurables desde la interfaz, los parámetros privados y propios del procesado interno y los que dependen del cálculo o la obtención de otros parámetros, se procede a ejecutar este objeto mediante el comando `audiotestbench(pluginWave)`, siendo `pluginWave` el nombre dado al *plugin* que contiene todos los efectos desarrollados en esta parte del proyecto y cuyo código

go puede consultarse en el fichero (`pluginWave.m`). La GUI resultante puede observarse en la Figura 20

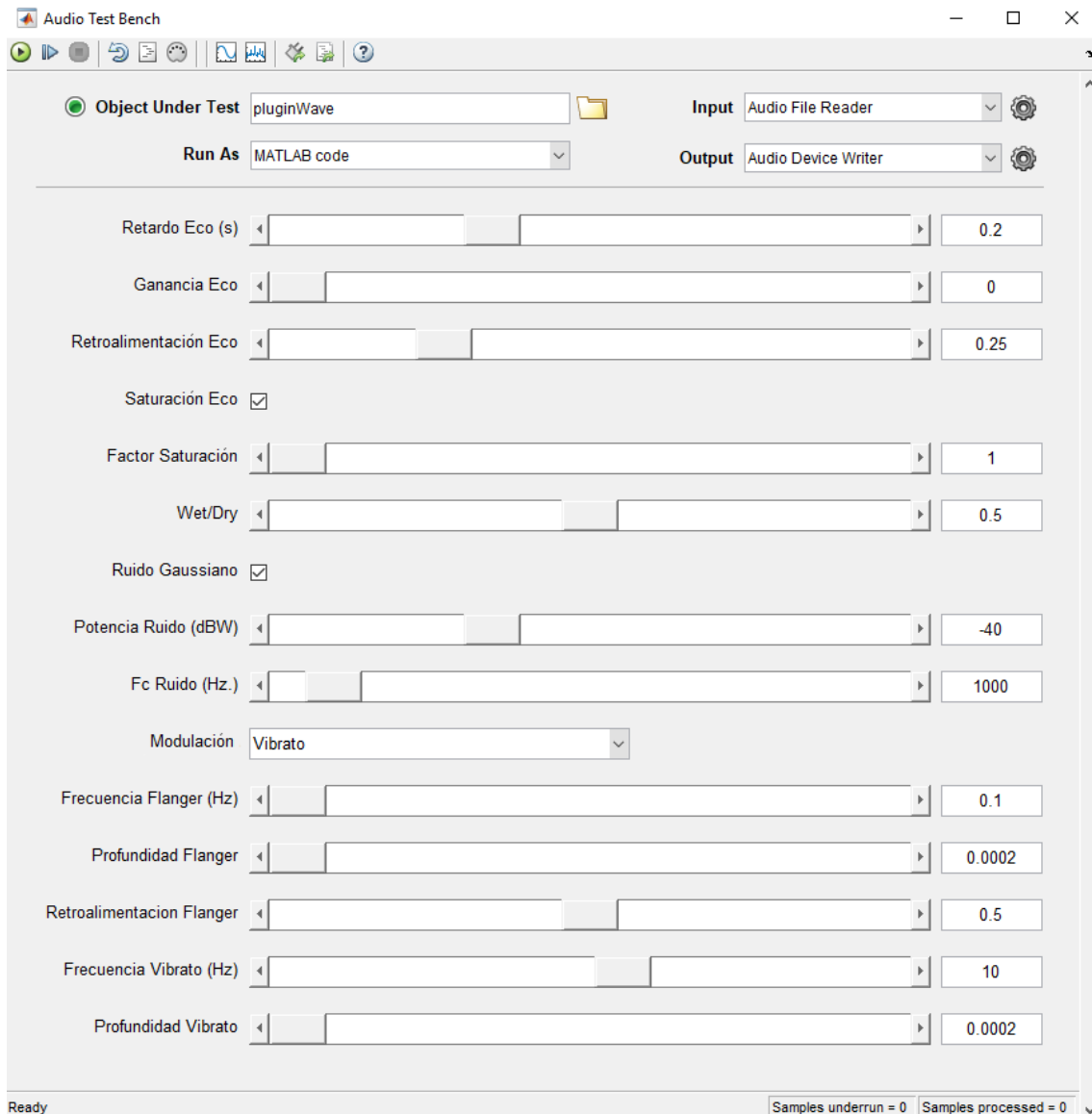


Figura 20: Interfaz gráfica de usuario generada por Audio Testbench

Los tres primeros controles deslizantes configuran los parámetros del efecto eco, cuya ganancia por defecto queda establecida a un valor de 0, ya que este efecto deberá activarse ajustando el segundo control deslizante [Ganancia Eco] para añadir este efecto a la salida de forma gradual hasta alcanzar el valor de 1, y cuyo tiempo de retardo en unidades de segundos puede ajustarse con el primer control deslizante [Retardo Eco(s)]. En tercer lugar se sitúa el control deslizante [Retroalimentación Eco] que ajusta la cantidad de repeticiones de Eco en la señal de salida o *Feedback*.

Justo debajo de los controles del eco se sitúa la casilla de *passthrough* para activar el

efecto de saturación sobre la línea de eco, cuyo factor de saturación α puede ser modificado mediante el control deslizante [Factor Saturación]. Debajo del mismo se sitúa el control de *Wet/Dry* que modifica la proporción de eco, ya sea saturado o sin saturar, en la señal de salida. Este control permite compensar el aumento de ganancia provocado por el factor de saturación y así únicamente obtener el efecto de clipeado suave característico de los equipos analógicos, aunque es independiente de la activación de la etapa de saturación y se centra en controlar la proporción de eco realimentado a la salida de esta etapa.

Seguidamente se sitúa la casilla [Ruido Gaussiano] que activa la etapa de ruido que podrá ser modificado con los controles inferiores: [Potencia Ruido] para aumentar su potencia y [Fc Ruido] para ajustar la frecuencia de corte del filtro paso-bajo integrado.

A continuación se sitúa un desplegable llamado [Modulación], en cuyo interior se pueden seleccionar tres opciones: Sin modulación, (que baipasea esta etapa), Flanger y Vibración, que activa los respectivos efectos, cuyos controles se sitúan en la parte inferior, y que, con la activación o desactivación de uno de los dos efectos, los respectivos controles del efecto desactivado quedan inutilizados. [Frecuencia Flanger] y [Frecuencia Vibrato], con dos rangos de valores diferentes adaptados a la naturaleza de cada efecto, modifican la velocidad de oscilación de la función senoidal encargada de modular el retardo. [Profundidad Flanger] y [Profundidad Vibrato] aumentan la amplitud de la moduladora y por tanto generan un efecto más exagerado. Además, el efecto *flanger* tiene el parámetro particular [Retrealimentación Flanger] cuya incremento de su valor aumenta la ganancia de la línea con retardo variable y por tanto la proporción del efecto en la salida.

Para un correcto funcionamiento de todos los efectos, antes de la ejecución del *plugin* es necesario configurar el objeto de lectura `Audio File Reader` pulsando el botón con forma de engranaje situado en la esquina superior derecha, donde es posible seleccionar tanto el archivo de entrada como el tamaño de bloque, que deberá ser ajustado a un valor de 1024 muestras, pues la implementación de todos los efectos ha sido optimizada para este valor, que permite una baja latencia en la manipulación de los parámetros y la escucha del resultado en tiempo real.

Finalmente es necesario asegurarse de la correcta configuración del objeto de escritura `Audio Device Writer` y seleccionar el driver adecuado para la tarjeta de sonido y la correspondiente frecuencia de muestreo, normalmente a un valor de 44100 Hz. Para el Sistema Operativo Windows se recomienda el uso del driver ASIO, que asegura la compatibilidad con el objeto de escritura de MATLAB.

5 Futuras líneas de trabajo

A lo largo de este proyecto se ha realizado un prototipado básico de los efectos propuestos en forma de *plugins*, sin embargo, debido a limitaciones de tiempo, muchos aspectos a mejorar y problemas encontrados durante su implementación se han dejado para posibles versiones futuras de este proyecto. A continuación se enumera un listado de aspectos a completar o mejorar en futuras líneas de trabajo:

- A pesar de que la ejecución en tiempo real del filtro analógico virtual basado en el modelo Moog Ladder no genera problemas incluyéndolo en el bucle de procesado de un *plugin* independiente, el coste computacional de ejecutar la función \tanh en el bucle de la etapa de saturación para cada bloque de muestras sobremuestreado en un factor 2, es elevado y su ejecución resulta ineficiente, además de provocar *underruns* al ejecutarse simultáneamente con otro algoritmo de procesado que requiera el uso de un bucle que ejecute operaciones muestra a muestra. Tal y como queda propuesto en (Zölzer, 2011), una solución sería la del empleo de una LUT para mapear la función no lineal \tanh de una forma más eficiente.
- Una posible mejora que permita ejecutar el *plugin* de filtrado con el *plugin* multiefectos de forma simultánea, aunque no paralela, consistiría en la conexión en cascada de estos dos *plugins*, una vez mejorada la eficiencia del *plugin* de filtrado, aunque no se ha comprobado si afectaría negativamente a la latencia. MATLAB permite esta conexión mediante la programación de un banco de pruebas propio y una posterior implementación de una GUI propia.
- Del mismo modo que se propone para el caso del empleo de la función \tanh en la saturación incluida en el *plugin* de filtrado “en escalera”, la función no lineal \arctan empleada en la etapa de saturación del *plugin* multi-efectos resulta ineficiente en cuanto a coste computacional por su empleo muestra a muestra, por lo que el uso de una LUT correspondiente mejoraría la eficiencia de este procesado.
- Como ya se ha explicado en el apartado 4.4, el filtrado por bloques se ha realizado de forma ineficiente para la construcción del *plugin* debido al uso de la función `filter()`, además de los clicks introducidos, que aunque resulten inapreciables y enmascarados bajo el ruido, es posible evitarlos mediante un filtrado en el dominio frecuencial empleando la FFT y el método de solapamiento y suma (o solapamiento y almacenamiento) para reconstruir la señal a partir de los bloques filtrados, tal y como se ha realizado en el filtrado presente en el *plugin* de filtrado virtual analógico, pero que no se ha llegado a realizar de la misma forma en la etapa de ruido por limitaciones de tiempo.

- Tal y como se menciona en el apartado 4.5, a pesar de que el vibrato bajo ciertos parámetros produzca un resultado similar al efecto *wow* presente en los equipos basados en cinta magnética, es posible una simulación más realista de acuerdo con el modelo propuesto en (Välimäki; González y col., 2008) sin embargo, la respectiva implementación de este modelo no se ha llegado a completar con éxito y el resultado no resulta tan realista, ya que se basa en un vibrato simple con el empleo de una senoidal como moduladora del retardo instantáneo.
- Tras múltiples intentos de compilación de cada uno de los dos *plugins* para su exportación en formato VST, compatible con la mayoría de entornos de edición de audio, se han encontrado problemas de compilación que no han sido superados debido a causas desconocidas y relacionadas con el compilador empleado. Se propone como principal línea de trabajo futura que complete este proyecto, resolver estos problemas de compilación hasta conseguir un resultado final en formato VST que quede al acceso de cualquier usuario que disponga de un DAW sin la necesidad de ejecutarlo en el entorno MATLAB. De esta forma se superaría la fase de desarrollo de prototipado y se alcanzaría la fase de producto final al acceso del usuario medio.

6 Conclusiones

A través de este proyecto se han revisado y adquirido conocimientos del campo del procesamiento digital de audio para aplicaciones musicales, y mediante el uso de la programación orientada a objetos se han implementado diferentes algoritmos, intentando en la medida de lo posible, su optimización para un procesamiento en tiempo real mediante la técnica de procesamiento por bloques de un tamaño de 1024 muestras. Este tamaño ha sido escogido con el objetivo de reducir al máximo la latencia de procesamiento, que para una frecuencia de muestreo de 44100 Hz tendrá un valor de $1024/44100 = 23,22ms$. Este reducido valor permite al usuario manipular los parámetros y comprobar instantáneamente el resultado de estas manipulaciones en el sonido, facilitando su sincronización con el posible tempo del fichero musical de entrada.

Se ha comprobado la eficiencia y el buen funcionamiento de las herramientas de programación orientadas a objetos en el entorno MATLAB y más concretamente, mediante el uso de la clase *plugin* proporcionada en el paquete de herramientas Audio Toolbox, la cual permite que este procesamiento por bloques se realice de forma automática sin necesidad de implementar manualmente procesos como el de lectura y escritura, ya que el paquete incluye objetos de biblioteca que sólo es necesario configurar para un rápido prototipado de un procesamiento en tiempo real. Esta técnica permite poner el foco en el desarrollo del algoritmo y delega el resto del trabajo a las funciones de biblioteca y herramientas proporcionadas por MATLAB.

El objeto de clase *plugin*, al permitir incluir en su interfaz programática unos parámetros modificables en tiempo real con sus respectivas funciones de llamada, que pueden ser accedidas a través de una GUI propia, y mediante el uso de la herramienta Audio Testbench, permite acelerar el desarrollo al proporcionar una GUI por defecto en la que se incluyen los controles de los parámetros de forma que los procesados puedan ser testeados para comprobar su funcionamiento en tiempo real de una forma rápida y sencilla, sin necesidad de construir la GUI desde cero. Y aunque esto pueda suponer una ventaja al reducir la cantidad de trabajo a realizar, la versatilidad de este banco de pruebas queda limitado a las funciones que MATLAB ha propuesto, y pueden suponer una pérdida de control y generar complicaciones para las que el desarrollador no pueda tener acceso para solucionarlas. Un ejemplo de esto observado en este proyecto ha sido la dificultad de ejecutar procesados en cascada, ya que bajo una única clase *plugin* sólo existe un bucle de procesamiento que pueda ser puesto a prueba en un sólo Audio Testbench. Puesto que la complejidad de cálculo de algunos algoritmos ha impedido su ejecución en paralelo, no ha sido posible comprobar su funcionamiento mediante su conexión en cascada. Para ello, en lugar de usar la herramienta Audio Testbench, sería necesario interconectar varias clases de tipo *plugin* a través de un banco de pruebas propio que ejecute una GUI diseñada desde

cero. Es decir, por una parte este tipo de herramientas aceleran el desarrollo, pero por otra parte, en ocasiones generan un excesivo nivel de abstracción sobre los procesos que se están llevando a cabo de forma desconocida e inaccesible al desarrollador. Otro ejemplo representativo de este problema ha consistido en intentar manipular la frecuencia de muestreo interna en tiempo real con el objetivo de implementar un efecto de variación de velocidad de reproducción y tono típica de los aparatos de reproducción analógica como los tocadiscos o los magnetófonos. El objeto de escritura, una vez ejecutado no permitía manipular esta frecuencia durante el procesado y a pesar de la sencillez de esta operación, su implementación ha resultado imposible debido a las limitaciones impuestas por el diseño del objeto de librería.

Pese a que uno de los principales objetivos de este proyecto era el de alcanzar una fase de desarrollo final en la que la clase *plugin* pudiera ser exportada al formato VST, esto no ha sido alcanzado debido a problemas de compilación que no han sido resueltos en el tiempo dispuesto para este proyecto. Debido a esto, el producto final y al acceso del usuario no ha podido ser testeado desde un DAW para comprobar su total funcionamiento más allá del prototipo ejecutado bajo el entorno de MATLAB. A pesar de esto, se han cumplido el resto de objetivos que consistían en poner a prueba los conocimientos adquiridos para implementar una emulación analógica que relacionase el campo de la tecnología digital con el de la analógica. Estableciendo este enlace entre el presente y el pasado de la tecnología de audio es posible recuperar de forma aproximada estas características analógicas que muchos usuarios echan en falta en los equipos digitales, y que gracias a la potencia de procesado de la que hoy disponemos resulta relativamente sencillo emular, obteniendo un resultado muy aproximado y ofreciendo estas características que poseían equipos de un elevado coste económico, ahora al acceso de un usuario medio y a través de su ordenador personal. De esta forma se ha potenciado que en la última década se multiplique el número de productores que, sin a penas recursos, son capaces de realizar producciones musicales avanzadas. Esto supone una democratización de la creación musical a través de la tecnología digital y gracias a los efectos virtuales analógicos se ofrece un acceso de forma atemporal a cualquier estética musical sin necesidad de disponer de hardware específico o mecanismos considerados hoy en día anticuados o técnicamente obsoletos, pero que bajo determinados contextos socioculturales (mencionados en el apartado 1.1) puedan ser de interés para el usuario.

7 Referencias

- BODE, Harald, 1984. History of electronic sound modification. *Journal of the Audio Engineering Society*. Vol. 32, n.º 10, pp. 730-739.
- CASCONE, Kim, 2000. The Aesthetics of Failure: "Post-Digital" Tendencies in Contemporary Computer Music. *Computer Music Journal*. Vol. 24, n.º 4, pp. 12-18.
- HUGILL, Andrew, 2018. *The digital musician*. Routledge.
- HUOVILAINEN, Antti, 2004. Non-linear digital implementation of the Moog ladder filter. In: *Non-linear digital implementation of the Moog ladder filter. Proceedings of the International Conference on Digital Audio Effects (DAFx-04)*, pp. 61-64.
- KATZ, Bob, 2015. *Mastering audio: the art and the science*. 3rd ed.. Burlington, MA ; Abingdon, RN: Focal Press. ISBN 978-0-240-81896-2.
- MATHWORKS, Inc., 2016. *Audio Toolbox User's Guide*.
- MOOG, Robert A., 1965. A voltage-controlled low-pass high-pass filter for audio signal processing. In: *A voltage-controlled low-pass high-pass filter for audio signal processing. Audio Engineering Society Convention 17*. Audio Engineering Society.
- PEKONEN, Jussi y VÄLIMÄKI, Vesa, 2011. The brief history of virtual analog synthesis, pp. 461-466.
- STILSON, Tim y SMITH, Julius O., 1996. Analyzing the Moog VCF with considerations for digital implementation. In: *Analyzing the Moog VCF with considerations for digital implementation. Proceedings of the 1996 International Computer Music Conference, Hong Kong, Computer Music Association*.
- VÄLIMÄKI, Vesa; GONZÁLEZ, Sira; KIMMELMA, Ossi y PARVIAINEN, Jukka, 2008. Digital audio antiquing-signal processing methods for imitating the sound quality of historical recordings. *Journal of the Audio Engineering Society*. Vol. 56, n.º 3, pp. 115-139.
- VÄLIMÄKI, Vesa y HUOVILAINEN, Antti, 2006. Oscillator and filter algorithms for virtual analog synthesis. *Computer Music Journal*. Vol. 30, n.º 2, pp. 19-31.
- ZÖLZER, Udo, 2011. *DAFX: digital audio effects*. John Wiley & Sons.

A Glosario de Acrónimos

DAW Entorno de Trabajo de Audio Digital (*Digital Audio Workstation*)

DFT Transformada Discreta de Fourier (*Discrete Fourier Transform*)

FFT Transformada Rápida de Fourier (*Fast Fourier Transform*)

GUI Interfaz Gráfica de Usuario (*Graphic User Interface*)

IFFT Transformada Inversa Rápida de Fourier (*Inverse Fast Fourier Transform*)

LUT Tabla de Consulta (*Look Up Table*)

B Glosario de Términos

Aliasing También llamado solapamiento, es un efecto que causa que las señales continuas distintas se tornen indistinguibles cuando se muestrean digitalmente. Según el teorema de Nyquist, esto se produce cuando la frecuencia de muestreo es inferior al doble del valor máximo de las componentes frecuenciales de la señal. Como resultado, la señal original no puede ser reconstruida de forma unequívoca a partir de la señal digital. A diferencia del *imaging*, esta ambigüedad se produce durante el diezmado en muestras de la señal.

Auto-oscilación Un sistema es auto-oscilante cuando genera una señal periódica como resultado de una inestabilidad lineal ocasionada por una retroalimentación entre la oscilación y la fuente de alimentación del sistema. A diferencia con los sistemas oscilantes, no es necesaria una potencia externa que sostenga la oscilación.

Baipasear (Del inglés *bypass*). Se dice que una etapa del procesado queda baipaseada si la señal no pasa por ella y es sorteada.

Búfer (Del inglés *buffer*) Memoria intermedia que almacena datos de forma temporal, especialmente entre dos procesos que no trabajan al mismo ritmo.

Bug Error de programa que desencadena un resultado indeseado.

Click Pequeño ruido impulsivo agudo, normalmente debido a un error de procesado digital.

Clipeado (Del inglés *clipping*). Distorsión de la señal provocada cuando la amplitud es superior al rango de valores que el sistema puede soportar.

Cuantificación Etapa de la conversión analógico-digital en el que se toman valores discretos de amplitud. El ruido de cuantificación es resultado del error introducido al discretizar estos valores.

Dry Señal previa al procesado y en relación a su mezcla con la señal procesada llamada *wet*.

Glitch Error que afecta negativamente al rendimiento o estabilidad de un sistema digital. No se considera un error de programa sino una característica no prevista.

Eco Versiones retardadas en el tiempo de una señal sonora

Efecto de audio Modificación del sonido mediante técnicas de procesado de la señal de audio.

Efecto Doppler En acústica, producido cuando una fuente sonora mueve con respecto al observador, que percibe cambios en su tono debido a los cambios en sus frecuencias aparentes.

Filtro de peine Efecto de filtrado que produce huecos frecuenciales a intervalos periódicos en el espectro.

Imaging Efecto que ocurre de la misma forma que el *aliasing*, pero durante la reconstrucción de una señal que ha sido sobremuestreada mediante un rellenado por ceros.

Mapear Técnica de programación que permite convertir datos mediante el uso de una tabla o base de datos que relaciona los datos de entrada con los correspondientes datos de salida.

Modulación Proceso de alterado de un parámetro, normalmente mediante un oscilador automático o programado.

Muestreo Proceso de tomar y almacenar valores de una señal continua cada cierto periodo de tiempo, inversamente proporcional a la frecuencia de muestreo.

Multi-efectos Procesado de señal que contiene diferentes efectos en un sólo paquete.

Passtrough Mecanismo que deja pasar la señal a través de una etapa de procesado.

Productos de intermodulación distorsión generada por las no linealidades del amplificador de salida de un transmisor.

Plugin O *plug-in*, es un programa que se inserta dentro de otro para proporcionarle funciones adicionales y específicas. En audio digital, este programa suele integrarse en un DAW.

Retroalimentación Envío de la salida de una etapa de un sistema de vuelta a la entrada de la misma.

Ruido blanco gaussiano Señal aleatoria cuya densidad espectral corresponde con la distribución normal (o distribución de Gauss).

Sample A lo largo de este documento, este término se traduce del inglés como muestra para referirse a un único dato que compone la señal digital, pero sin traducir, se refiere a un fragmento o fichero de audio con el que se suelen realizar composiciones musicales derivadas.

Tempo Velocidad a la que se interpreta una composición musical. Se mide en pulsaciones por minuto o *bpm* (del inglés *beats per minute*)

Tiempo real Se dice que un sistema de procesado se ejecuta en tiempo real cuando procesa la señal en los instantes en los que se recibe y con la suficiente velocidad para ejeturar los cálculos dentro de un periodo de procesado.

Tono Percepción subjetiva de la frecuencia de un sonido.

Underruns Pérdidas de muestras a la salida de un búfer debido a que la velocidad de lectura es superior a la velocidad con la que el búfer es llenado.

Vibrato Variación cíclica a una frecuencia de unos pocos Hercios, típicamente de 3 a 8 Hz.

Wet Señal procesada en relación a su mezcla con la señal inicial llamada *dry*.

Wow Variación instantánea a baja frecuencia de la velocidad de reproducción y por tanto, del tono. Este efecto es característico de los dispositivos equipos de grabación y reproducción basados en cinta magnética debido a las irregulares oscilaciones de las bobinas y las variaciones en la velocidad angular de los cabezales.