

# ANÁLISIS DE UN FILTRO CONTROLADO POR TENSIÓN EN EL SINTETIZADOR ANALÓGICO RSF KOBOL EXPANDER

Martín Fernández, Ricard

Curso 2021-22

Director: Perfecto Herrera  
Xavier Lizarraga

GRADO EN INGENIERÍA DE SISTEMAS AUDIOVISUALES



Universitat  
Pompeu Fabra  
Barcelona

Escola  
d'Enginyeria

## Trabajo de Fin de Grado



Universitat  
Pompeu Fabra  
Barcelona

Escola  
d'Enginyeria



*A mi familia y mis amigos músicos*



## **Agradecimientos**

Quiero agradecer sobre todo a mis tutores, Perfecto Herrera y Xavier Lizarraga, por su ayuda durante todo el proyecto. Asimismo, a mi familia por el apoyo incondicional que me han dado durante este trabajo final y durante toda la carrera. También me gustaría agradecer a mis amigos y a mis compañeros de mis grupos musicales por animarme y ayudarme a desconectar de vez en cuando.



## Resumen

El objetivo de este trabajo es analizar el comportamiento del filtro controlado por voltaje<sup>1</sup> de una unidad del sintetizador analógico RSF Kobol Expander. Para ello, primero hemos revisado la documentación relacionada con la unidad para conocer las especificaciones técnicas de cada control y su funcionamiento. A continuación, hemos creado una base de datos, los cuales hemos usado para el posterior análisis. Ésta se ha obtenido enviando señales de audio desde un ordenador hasta el filtro, grabando la salida habiendo manipulado los potenciómetros de este, tanto manualmente como con una fuente de voltaje externa. En la tercera y última parte del proyecto, hemos analizado detalladamente estas salidas, usando diversos códigos escritos en *Python*. Con este análisis, podemos comparar los valores de los potenciómetros con los valores encontrados empíricamente y conocer el comportamiento real del filtro con vistas a una futura virtualización del mismo, entre otras aplicaciones posibles.

## Resum

L'objectiu d'aquest treball és analitzar el comportament del filtre controlat per voltatge<sup>2</sup> d'una unitat del sintetitzador analògic RSF Kobol Expander. Per això, primer hem revisat la documentació relacionada amb la unitat per conèixer les especificacions tècniques de cada control i el seu funcionament. A continuació, hem creat una base de dades, les quals hem fet servir pel posterior anàlisi. Aquesta s'ha obtingut enviant senyals d'àudio des d'un ordinador fins al filtre, gravant la sortida havent manipulat els potenciómetres d'aquest, tant manualment com amb una font de voltatge externa. A la tercera i última part del projecte, hem analitzat detalladament aquestes sortides, fent servir diversos codis escrits en *Python*. Amb aquest anàlisi, podem comparar els valors dels potenciómetres amb els valors trobats empíricament i conèixer el comportament real del filtre amb vista a una futura virtualització del mateix, entre altres aplicacions possibles.

## Abstract

The objective of this work is to analyze the behavior of the voltage controlled filter<sup>3</sup> of an RSF Kobol Expander analog synthesizer unit. To do this, first we have reviewed the documentation related to the unit to know the technical specifications of each control and how it works. Next, we have created a database, which we have used for further analysis. This has been obtained by sending audio signals from a computer to the filter, recording the output having manipulated its potentiometers, both manually and with an external voltage source. In the third and last part of the project, we have analyzed these outputs in detail, using various codes written in Python. With this analysis, we can compare the values potentiometer with the values found empirically and know the real behavior of the filter with a view to its future virtualization, among other possible applications.

---

<sup>1</sup> en inglés conocido como *Voltage Controlled Filter* (VCF) significa que se pueden regular sus parámetros mediante un aparato externo

<sup>2</sup> en anglès conegut com *Voltage Controlled Filter* (VCF) significa que es poden regular els seus paràmetres mitjançant un aparell extern

<sup>3</sup> in english known as *Voltage Controlled Filter* (VCF) meaning that its parameters can be regulated by an external device





# Índice

1. Introducción	pag 13
2. RSF Kobol Expander	pag 15
2.1. Historia y funcionamiento general	pag 15
2.2. Parámetros y controles	pag 16
3. Metodología	pag 19
3.1. Preguntas de investigación	pag 19
3.2. Ajustes del equipo	pag 19
3.3. Señales de prueba y nomenclatura	pag 20
3.4. Método de captura	pag 21
3.5. Procesamiento de señales	pag 23
3.6. Función de transferencia	pag 23
3.7. Obtención de Datos	pag 24
4. Resultados	pag 27
4.1. Análisis global	pag 27
4.2. Caracterización paramétrica del filtro	pag 31
5. Conclusiones	pag 45
Bibliografía	pag 47
Apéndices	pag 49
1. Figuras	pag 49
2. Nomenclatura	pag 56
3. Ecuaciones y fórmulas	pag 57
4. Tablas	pag 58
5. Explicaciones adicionales	pag 65
6. Listado del material de grabación	pag 67
7. Resumen extendido	pag 68

## Lista de figuras

Figura 2.1 del Apéndice 1: RSF Kobol  
Figura 2.2 del Apéndice 1: RSF Kobol Expander I  
Figura 2.3 del Apéndice 1: VCF  
Figura 3.1 del Apéndice 1: Configuración inicial  
Figura 3.2 del Apéndice 1: Loopback  
Figura 3.3 del Apéndice 1: Montaje de grabación  
Figura 3.4 del Apéndice 1: Función de transferencia de un filtro  
Figura 4.1 del Apéndice 1: tone100 (input)  
Figura 4.2 del Apéndice 1: Bypass\_tone100 (output)  
Figura 4.3 del Apéndice 1: tone440 (input)  
Figura 4.4 del Apéndice 1: Bypass\_tone440 (output)  
Figura 4.5 del Apéndice 1: tone1k (input)  
Figura 4.6 del Apéndice 1: Bypass\_tone1k (output)  
Figura 4.7 del Apéndice 1: tone10k (input)  
Figura 4.8 del Apéndice 1: Bypass\_tone10k (output)  
Figura 4.9 del Apéndice 1: THD  
Figura 4.10 del Apéndice 1: VCF\_4096F\_allRes\_sweepstat  
Figura 4.11 del Apéndice 1: FFT\_Noise\_Tones  
Figura 4.12: TF\_512F\_0R\_sweepstat  
Figura 4.13: Frecuencias\_de\_corte\_sweepstat  
Figura 4.14: TF\_1024F\_10R\_sweepstat  
Figura 4.15: Rango\_Resonancia\_sweepstat  
Figura 4.16: Resonancias\_sweepstat  
Figura 4.17 del Apéndice 1: TF\_64F\_0R\_sweepvar  
Figura 4.18 del Apéndice 1: TF\_512F\_5R\_sweepvar  
Figura 4.19: TF\_CV\_freq\_1V\_sweepstat  
Figura 4.20: TF\_CV\_res\_7V\_sweepstat  
Figura 4.21: Frecuencias de corte en control manual vs Frecuencias de corte en CV  
Figura 4.22: Resonancias en control manual vs Resonancias CV

## Lista de tablas

Tabla 4.1: SNR

Tabla 4.2: THD

Tabla 4.3 del Apéndice 4: Frecuencia\_sweepstat

Tabla 4.4 del Apéndice 4: Resonancia\_sweepstat

Tabla 4.5 del Apéndice 4: Frecuencia\_sweepvar

Tabla 4.6 del Apéndice 4: Resonancia\_sweepvar

Tabla 4.7 del Apéndice 4: Frecuencia\_CV\_sweepstat

Tabla 4.8 del Apéndice 4: Resonancia\_CV\_sweepstat

Tabla 4.9 del Apéndice 4: Frecuencia\_CV\_sweepvar

Tabla 4.10 del Apéndice 4: Resonancia\_CV\_sweepvar



# 1. Introducción

En este primer apartado del trabajo expondré la motivación de realizar dicho proyecto y cuáles son sus objetivos. En este proyecto me propongo analizar el funcionamiento del filtro pasa-bajos (en inglés abreviado como LPF) controlado por tensión de una unidad del sintetizador analógico *RSF Kobol Expander*.

Un sintetizador es un instrumento musical de tipo electrónico que, a través de circuitos, genera señales eléctricas que son convertidas en sonidos. Una característica que diferencia al sintetizador de otros instrumentos electrónicos es que éstos pueden imitar otros instrumentos o generar nuevos. El sintetizador analógico usa tensión continua, generada y aplicada de diferentes maneras, para crear y controlar sonidos [1]. Esta capacidad de generar muchos sonidos diferentes hace del sintetizador un instrumento musical de gran calibre, que se ha usado en muchos géneros musicales, entre los cuales están, sobre todo, la música electrónica o el synthpop, pero también en géneros como el industrial metal o el nu metal.

La creación de sonido sintético tiene al *Theremin* y al órgano *Hammond* como interesantes exponentes primigenios (aún usados actualmente), pero la síntesis analógica se desarrolla y se utiliza desde mediados del siglo XX. El sintetizador controlado por tensión es un invento de *Robert Moog* de 1964. Su uso se extendió a lo largo de los años 70 y principios de los 80, para caer en cierto olvido hasta el revival producido gracias al formato *Euro rack* hará unos 10 años o más. Muchos otros fabricantes como *Roland*, *Arturia* y *Behringer* en los últimos años no han parado de lanzar productos y réplicas al mercado.

Un dispositivo analógico genera una señal continua, no discretizada y es imperfecto sónicamente, ya que padece de desviaciones, ruidos y otros aspectos técnicos que normalmente son inexistentes en el audio digital (como la latencia y el *aliasing*). De hecho, es casi imposible que una unidad del mismo modelo suene exactamente igual que otra. Pueden ser similares pero el estado de cada componente, su diseño y el control de calidad durante la producción definen esas pequeñas diferencias. Actualmente, los dispositivos analógicos siguen consumiéndose y van al alza en un entorno profesional, y una prueba de ello es que en la mayoría de los estudios de grabación apuestan por un sistema híbrido donde conviven el mundo analógico y el digital.

Uno de los elementos más característicos y diferenciadores de un sintetizador es su filtro controlado por tensión, a menudo de tipo pasa-bajos. Un filtro pasa-bajos es un proceso que se aplica a una señal para mantener las frecuencias graves y eliminar las agudas.

Otra motivación a destacar es que al ser un sintetizador muy antiguo (capítulo 2) es posible que haya componentes que estén en mal estado, por lo que el sintetizador puede tener un comportamiento muy diferente al esperado. Hay que tener en cuenta la importancia de conocer bien el VCF para la preservación del equipamiento clásico de audio mediante la virtualización del proceso a través del modelado o del análisis.

Un VCF es un filtro cuya frecuencia de corte depende del valor del voltaje de la señal de control de entrada [2]. En otras palabras, modifica la respuesta de la señal en función de la envolvente de la señal de control, que puede ser externa o procedente de otro módulo del sintetizador.

La extracción de características o el *footprint* de cada configuración del VCF, la haremos extrayendo respuestas impulsionales (IR), (capítulos 3 y 4). Para conocer el comportamiento del VCF, es necesario disponer de una base de datos con muestras, obtenidas a partir de señales de prueba, que hemos analizado para extraer resultados. La calidad de esta base de datos (resolución y calidad del audio) definirá la calidad del resultado final del análisis. Analizando las muestras registradas podremos conocer el valor de la frecuencia de corte del filtro, también el valor de su pendiente y de los parámetros de la resonancia para cada ajuste de sus controles. El objetivo es capturar el efecto de cada parámetro por separado para conocer su rango, desviación y comportamiento o respuesta (si es lineal, logarítmica o exponencial).

Por otro lado, también es importante conocer las especificaciones técnicas de la unidad proporcionadas por el fabricante y analizar aspectos técnicos como la *Total Harmonic Distortion* (THD) o la *Signal to Noise Ratio* (SNR), puesto que también forman parte de la señal de salida y nos pueden ayudar a obtener un modelado más realista.

En el capítulo 2, veremos cómo funcionan los parámetros del filtro, además de explicar brevemente la historia y el funcionamiento general del sintetizador *RSF Kobo! Expander I*, y daremos más detalles sobre la captura de muestras en el capítulo 3. En el 4, nos centramos en el análisis. Finalmente, en el capítulo 5 extraemos conclusiones de los resultados más relevantes. Cabe mencionar que, además, este documento cuenta con [Apéndices](#) con ecuaciones, gráficos y resultados, nomenclatura y explicaciones extras, así como la versión extendida del resumen y una lista del equipo usado para la captación.

## 2. RSF Kobil Expander

### 2.1 Historia y funcionamiento general

Generalmente, un VCF se basa en los siguientes pasos:

- Existe una señal de entrada que excita una envolvente temporal. Ésta puede ser externa, que es como lo haremos en este proyecto, o interna (por ejemplo, la salida producida por un generador de ondas).
- Esta envolvente temporal se mapea a un valor en frecuencia.
- Este valor define la frecuencia de corte del filtro.

Es decir, tenemos un filtro pasa bajos variable según la envolvente de la señal de entrada. Los niveles de tensión analógicos producidos por otro instrumento o por una fuente externa de tensión continua (DC) se pueden utilizar para controlar las funciones y los parámetros de un módulo. Por ejemplo, si un módulo de sintetizador presenta una entrada de Control por Voltaje (CV) en sus parámetros, como la resonancia. Este mecanismo nos permite controlar la cantidad de resonancia que tiene el filtro usando una fuente de voltaje externa [3]. Como veremos en este capítulo, casi todos los controles del *Kobil* pueden ser regulados por un CV externo.

El *RSF Kobil* es un sintetizador monofónico lanzado en 1978. Fue creado por los hermanos *Ruben* y *Serge Fernandez* (Francia - Toulouse) [4]. En 1981, se lanzó el *RSF Kobil Expander* (ver [Figura 2.1 del Apéndice 1](#) y [Figura 2.2 del Apéndice 1](#)), una versión en rack del *RSF Kobil* original. La unidad contiene 2 *Voltage Control Oscillator* (VCO), 2 generadores de envolventes (ADS), 1 *Low Frequency Oscillator* (LFO), 1 VCF, 1 *Voltage Controlled Amplifier* (VCA), 1 generador de ruido y 1 procesador de voltaje [5]. La versión en formato rack ofrecía cuatro componentes diferentes. El *Kobil Expander 1* contenía, esencialmente, las secciones VCO/VCF/VCA/LFO del *Kobil* en un módulo de rack. El *Expander 2* fue un complemento del *Kobil* que agregó algunos módulos de procesamiento. El *Programmer* contenía circuitos de memoria para almacenar secuencias. Por último, el *KM8* era un mezclador de montaje en rack de 8 canales. Estos cuatro componentes se pueden unir o se pueden usar por separado.

El *Kobil* era atractivo y fue rápidamente acogido por la industria musical. Al ser una pequeña compañía, RSF nunca pudo producirlos en masa en el mismo tipo de escala que *Moog*, *Arp*, *Roland* o *Korg*. Se fabricaron alrededor de 800 *RSF Kobil Expander I*. La escasez y la baja disponibilidad de este modelo ha hecho que con el tiempo sean sintetizadores muy valorados (alrededor de 5000 euros una unidad [6]). Por otro lado, despiertan nostalgia e interés en los usuarios por su firma sónica, puesto que han sido utilizados por grandes artistas y grupos de música como Vince Clarke, Vangelis, Jean-Michel Jarre, Depeche Mode, Peter Gabriel y Hans Zimmer [7]. Debido al éxito que tuvo y a la cantidad de grandes músicos que lo usaron, así como a la moda de virtualizar y modelar equipamiento clásico por las marcas de *plugins*, se han creado diversos *plugins* y simuladores que emulan el funcionamiento de este sintetizador. El más importante es un *plugin* llamado *PolyKB II* [8], con algunas modificaciones respecto al diseño original, pero con un sonido y una estética muy similar.

Como introducción al siguiente subapartado, vamos a definir brevemente los módulos que forman el modelo *RSF Kobol Expander I*:

- **VCA**: Es un amplificador cuya ganancia depende del valor del voltaje de la señal de control de entrada.
- **VCF**: Es un filtro pasa bajos cuya frecuencia de corte depende del valor del voltaje de la señal de control de entrada.
- **VCO**: Es un oscilador que genera una onda con una frecuencia que depende del valor del voltaje de la señal de control de entrada
- **LFO**: Es un oscilador que genera una onda de baja frecuencia en su salida (es un tipo particular de VCO).
- **Generador de Envolvente**: Consiste en una serie de parámetros (*Attack*, *Decay*, *Sustain*) que modelan la envolvente temporal de la señal de salida [2].

## 2.2. Parámetros y controles

El *RSF Kobol Expander* tiene tres modos de funcionamiento: manual, de memoria y secuenciador. El manual permite utilizar el sintetizador con los controles manuales. El modo de memoria sirve para grabar y almacenar las modificaciones hechas en los ajustes o configuraciones en modo manual o secuenciador. Por último, el modo secuenciador sirve para reproducir una secuencia de notas [9]. En este trabajo hemos utilizado el modo manual.

- *Voltage Controlled Filter (VCF)*

En la [Figura 2.3 del Apéndice 1](#) podemos ver con claridad el módulo VCF, que es en el que nos centramos en este proyecto. Éste es un filtro pasa-bajos (LPF) de 4º orden, con una pendiente de 24 dB/octava. Dispone de cuatro controles rotatorios asociados con el filtro:

- Frecuencia de corte (*VCF Frequency*),
- Resonancia (*Resonancy*),
- Cantidad de ADS (*Attack*, *Decay* y *Sustain*)
- Control de teclado (*KeyB Control*).

También hay entradas para CV externo para cada uno de estos controles y una entrada de audio para procesar una señal externa [10]. De estos 4 parámetros, dos determinan el comportamiento del VCF:

- **VCF FREQUENCY**: define la frecuencia de corte.
- **RESONANCE**: define la campana de resonancia que se genera en la frecuencia de corte para dar un cierto carácter al sonido de salida. Si el potenciómetro está en el extremo izquierdo, el filtro eliminará gradualmente las oscilaciones de los armónicos superiores. Si está en el extremo derecho, el filtro entrará en resonancia infinita creando una onda sinusoidal audible, cuyo tono estará determinado por la frecuencia de corte [9].



Los otros dos controles modifican el sonido global del VCF y su relación con un teclado externo:

- *KEYB. CONTROL*: Realiza un seguimiento de la frecuencia central que le transmiten las notas de un teclado y desplaza la frecuencia de corte en consonancia a ésta. En otras palabras, permite que el VCF se sintonice con el teclado.
- *ADS CONTROL*: controla la envolvente temporal del VCF. La frecuencia de corte del filtro evolucionará según la envolvente creada con los controles de *Attack*, *Decay* y *Sustain*.

Para conseguir una virtualización realista del módulo VCF, durante el análisis, es importante conocer el rango exacto del parámetro *VCF FREQUENCY*, y la respuesta de sus controles (exponencial, logarítmica o lineal). En la serigrafía del control de la frecuencia de corte se indica un rango de 16 Hz a 16 kHz, pero, en la práctica, puede haber desviaciones o simplemente que esto no sea cierto. Con el parámetro de resonancia el objetivo es conocer el ancho de banda y la altura de la campana para ajustes discretos del control de este parámetro.

Por otro lado, como ya hemos mencionado, el filtro se puede controlar con señales externas. Para aislar el VCF de las señales que genera el VCO, utilizaremos esta entrada para conocer la respuesta del VCF. Para ello podemos rutear señales de prueba, ya conocidas, directamente a esta entrada de señal externa y ver cómo responde el filtro (capítulos 3 y 4).

Sin embargo, hay otros parámetros y factores internos que afectan al sonido resultante, añadiendo interferencias, tales como ruido y distorsión, que inducen comportamientos no lineales. Por tanto, también nos interesa caracterizar estos parámetros. Estos son tanto factores secundarios como la pendiente, el ancho de banda o la magnitud de la resonancia, que están relacionados con los potenciómetros de frecuencia y resonancia o factores más internos como el nivel y la envolvente espectral del ruido de fondo y el rango dinámico, los cuales están relacionados con el SNR. Al tratarse de hardware, estos últimos dos parámetros también son definitorios del resultado sonoro del filtro y su estimación permitirá alcanzar una virtualización más realista. Por otro lado, los parámetros de la resonancia, tales como la *Q* (ancho de banda) y la frecuencia de resonancia, aparecen definidos en el capítulo 4.



### 3. Metodología

En esta sección explicaremos con detalle qué procesos hemos realizado con tal de obtener los resultados que analizaremos en el capítulo 4.

#### 3.1. Preguntas de investigación

El primer paso es crear una base de datos con tal de conocer la respuesta del VCF ante una señal externa. Para ello, tenemos que preparar las señales de entrada. Éstas las definimos conforme a una serie de preguntas o hipótesis que nos hacemos sobre su comportamiento y se responden en el capítulo 4. Éstas nos ayudan a establecer un esquema o una guía para el análisis. Dicha serie de preguntas son las siguientes:

1. ¿Hay diferencias en la respuesta en frecuencia del filtro para [diferentes niveles](#) de señal de audio de entrada?
2. ¿Cuál es el rango de la frecuencia de corte?
3. ¿Cuál es el rango de la resonancia?
4. ¿Cuál es el comportamiento de los parámetros? ¿Lineal, logarítmico, exponencial?
5. ¿Cuál es la *Signal to Noise Ratio* (SNR) del filtro?
6. ¿Cuál es el nivel de *Total Harmonic Distortion* (THD) del filtro a la salida?
7. ¿Cuál es su pendiente real?
8. ¿Cuál es el rango de la Q definida por la resonancia?
9. ¿Responden igual los parámetros controlados por [CV](#)?

Las muestras grabadas se han capturado enviando señales de prueba, previamente creadas, desde un ordenador hasta el VCF, grabando su salida para diferentes valores de cada parámetro. Tanto manualmente como con una fuente de voltaje externa de laboratorio. Además, también hemos captado la salida en *bypass* (sin ningún proceso, con el potenciómetro de la frecuencia de corte al máximo y el de la resonancia al mínimo, es decir, con el filtro abierto) para disponer de una señal de referencia de la respuesta del sistema con el filtro inhabilitado. Nos servirá para comparar con los niveles de las tomas procesadas y obtener la función de transferencia del sistema (apartado 3.6 y 4.2). Facilitando la estimación de los parámetros de los controles de frecuencia de corte y de resonancia. Por otro lado, también se precisa una grabación del paso de la señal por la tarjeta de sonido, que contiene tanto la señal de control como el ruido y coloración aportados por la tarjeta <sup>4</sup>.

#### 3.2. Ajustes del equipo

Antes de realizar las muestras se precisa una calibración del sistema de grabación. Es importante asegurarnos que siempre estamos en las mismas condiciones de nivel. Para este paso, hemos tenido en cuenta el nivel de referencia, el cual depende de la tarjeta de sonido que usemos. Ésta es una unidad del modelo *Edirol UA-101*. La cual, usando las entradas balanceadas TRS, tiene un nivel de voltaje de salida máximo de 16 dBu y un voltaje nominal (de referencia) de 4 dBu [\[11\]](#). Recordemos que los dBu son los dB referenciados a 0.775 V, con lo cual, son medidas del nivel de tensión.

---

<sup>4</sup> Esa grabación es el denominado *loopback*

El nivel máximo se corresponde con los 0 dBFS de la señal digital (los dBFS hacen referencia al nivel del audio en un dispositivo digital sin ningún tipo de ponderación), a la cual, para tener la señal en el nivel de referencia, hay que restarle 12 dBFS (que es la diferencia entre el nivel máximo y el RMS de la tarjeta). Por otro lado, si queremos el voltaje de pico, tenemos que sumarle 3 dBFS (ya que ésta es la diferencia entre dB de pico y dB RMS), con lo cual, finalmente, tenemos que restar 9 dBFS a la señal original, con tal de obtener los mismos niveles de salida y de entrada preservando un cierto *headroom*. Por otro lado, más adelante veremos que, con cables desbalanceados hay una diferencia de 6 dB, con lo cual, nunca podemos tener los 16 dBu de salida, siempre estaremos entregando 10 dBu de salida, como máximo.

Otro ajuste importante es sobre los parámetros del VCF. Hemos hablado de 4 parámetros del VCF. Sin embargo, dos de ellos (KB y ADS) son parámetros que añaden detalles al filtro pero que no son parte de su funcionamiento básico, sino que sirven para controlar el filtro desde un teclado. Por tanto, ambos parámetros han de estar a 0, para que no afecten al comportamiento del VCF. De esta manera, en otras palabras, bajo nuestra configuración, el filtro solo responde a la frecuencia de corte y a la resonancia en esta frecuencia.

En el [Apéndice 6](#) podemos encontrar un listado del material usado para la captación de las muestras usadas en el análisis.

### 3.3. Señales de prueba y nomenclatura

Con tal de caracterizar el VCF, necesitamos crear una serie de señales de audio que hemos usado como señales de entrada o de prueba. Éstas entran en el filtro y se procesan con los dos potenciómetros con tal de obtener las muestras que analizaremos en el siguiente capítulo. Las hemos creado usando los *plugins* de *Aurora*<sup>5</sup> en *Audacity* [12]. Éstas son *sweeps* porque para poder observar el comportamiento del filtro según los parámetros de la frecuencia de corte y la resonancia necesitamos una señal que recorra todo el espectro frecuencial que nos interesa. Son señales usadas en el campo de la medición de audio para describir el cambio en un valor de salida medido sobre un parámetro de entrada progresivo. El parámetro de entrada progresivo más utilizado es la variación de frecuencia sobre el ancho de banda de audio estándar, de 20 Hz a 20 kHz [13]. Es importante mencionar que una *sweep* nos permite estimar la respuesta impulsional de un sistema mediante la convolución con su filtro inverso (sección 3.5).

A lo largo de éste y el siguiente capítulo, cuando se menciona la *sweep* estática nos referimos a una *sweep* que no cambia de nivel de magnitud, para diferenciarla de la que sí que cambia (*sweep* variable). Realizando los ajustes del subapartado anterior (3.2), las señales de entrada que tenemos para las diferentes preguntas son las siguientes:

**Pregunta 1:** Tren de *Sweeps* variables con 3 amplitudes diferentes (-9 dBFS, -19 dBFS y -29 dBFS), empezando con el nivel máximo referenciado (-9 dBFS), una diferencia de 10 dB y 5 segundos entre una señal y la siguiente.

**Preguntas 2, 3 y 4:** *Sweep* estática al nivel máximo (-9 dBFS). Para este caso se ha generado un tren de 11 *sweeps* estáticas con 5 segundos de silencio entre ellas, para poder grabar la salida con 11 ajustes diferentes del control de la frecuencia de corte y de la resonancia, manipulados manualmente.

---

<sup>5</sup> <http://www.aurora-plugins.com/>

[Preguntas 5 y 6](#): 4 tonos puros de 100 Hz, 440 Hz, 1 kHz y 10 kHz. (todos al nivel de referencia de -9 dBFS). Con esta elección, conseguimos conocer el comportamiento en tonos graves y en tonos agudos.

[Preguntas 7 y 8](#): En este caso no hace falta ninguna grabación y, por lo tanto, ninguna señal de entrada. Ya que son preguntas que responderemos con el análisis de las tomas para las preguntas 2, 3 y 4.

[Pregunta 9](#): Usamos las mismas señales que hemos definido en las preguntas 1, 2, 3 y 4. En vez de cambiar los parámetros manualmente, lo hacemos usando una fuente de voltaje externa de laboratorio.

Todas las señales de prueba se han generado con los siguientes parámetros:

Sample rate	Profundidad de bits	Formato	Canales	Duración
48 kHz	24 bits	WAV	1	15 segundos

Por otro lado, hemos definido una nomenclatura para identificar las grabaciones. Podemos ver ejemplos en el [Apéndice 2](#). Las señales de entrada tienen el nombre del tipo de señal que son, por ejemplo, *sweep\_static*. Por otro lado, las salidas incluyen en el nombre los parámetros que hemos manipulado, añadiendo su valor inferido. Asumimos que cada punto del potenciómetro de frecuencia se corresponde con una frecuencia de corte concreta (F o R hacen referencia a Frecuencia o Resonancia, respectivamente), y el tipo de señal de entrada de origen. Esto es importante porque, para llevar a cabo el análisis, hemos generado muchos archivos, por lo que es importante tener clara la nomenclatura, y que ésta sea fácil y comprensible para alguien externo al proyecto.

Como ya hemos detallado anteriormente, las señales de entrada se conectan directamente a la entrada de audio del VCF (AUDIO IN), evitando así la contribución de los módulos anteriores. No obstante, la salida de la señal (ésta sale por el *máster* del sintetizador), sí que pasa por módulos posteriores, como el VCA, y pueden falsear las muestras. Concretamente, descubrimos que el *Sustain* del ADS 1 debe ajustarse al máximo. Sucede igual con el ajuste del volumen de salida. La [Figura 3.1 del Apéndice 1](#) muestra la configuración inicial de los parámetros de frecuencia de corte y resonancia del filtro, de manera que el filtro está desactivado y no debería afectar al sonido entrante<sup>6</sup>.

### 3.4. Método de captura

El método de captura consiste en enviar la señal al módulo desde un ordenador y grabar la salida del VCF, con ajustes de frecuencia y resonancia concretos. Cada pregunta de las planteadas en la sección 3.1 tiene su propio método, aunque en todas (excepto la 9), la variación de los ajustes se ha hecho manualmente:

[Pregunta 1](#): (*sweep variable*)

Para la presente hipótesis hemos discretizado el rango del parámetro de frecuencia de corte en 4 puntos, que hemos llamado 0 CP, 1 CP, 2 CP y 3 CP (C de *cutoff* (frecuencia de corte en inglés) y P de parámetro) separados uniformemente y con 5 de resonancia, llamados 0 RP, 2 RP, 5 RP, 8 RP, 10 RP (R de resonancia y P de parámetro).

---

<sup>6</sup> la salida con este ajuste es la que definimos como *bypass*

Para ello hemos captado la señal de salida con las 20 combinaciones posibles: 0 CP y 0 RP, 0 CP y 2 RP, ..., 3 CP y 10 RP. A partir de la serigrafía del potenciómetro de frecuencia de corte y de la cantidad de puntos que hay en él, asumimos que estas posiciones se pueden mapear a 16 Hz, 64Hz, 512 Hz y 4096 Hz respectivamente (en el capítulo 4 lo comprobaremos). La frecuencia máxima del control indicada por el fabricante (16 kHz), no la tenemos en cuenta porque equivale al *bypass*.

#### Preguntas 2, 3 y 4: (*sweep* estática)

En este caso queremos ver cómo responde cada frecuencia al cambio de resonancia, para conocer el comportamiento del filtro con una señal a un mismo nivel y cuál es el rango real de cada control. Para ello, para cada frecuencia captamos la salida con 11 valores de resonancia (de 0 a 10), que corresponden a las secciones radiales marcadas en su control. Ajustamos el potenciómetro de la resonancia de menos a más, es decir, de izquierda a derecha. Igual que en el caso anterior, asumimos que cada posición del potenciómetro de frecuencia se puede mapear a una frecuencia de corte concreta de entre 16 Hz y 16 kHz.

#### Preguntas 5 y 6: (tonos puros)

Para conocer la *Signal to Noise Ratio* (SNR) y la *Total Harmonic Distortion* (THD) es preciso conocer la salida del VCF sin que éste actúe. Es decir, captamos el *bypass*, para poder compararlo con la señal de entrada original. Para ello, usamos tonos puros discretizando algunas bandas del espectro.

#### Pregunta 9: (fuente de voltaje externa)

En este caso, usamos una fuente de voltaje externa (la *PROMAX FAC-363B*) para modificar los parámetros controlados por CV. Hemos hecho la captación para la frecuencia de corte y resonancia. En ambos casos, hemos hecho 4 grabaciones enviando 4 niveles de voltaje de señal continua (DC) diferentes (1V, 4V, 7V y 10V). Sospechamos que estos niveles de voltajes se pueden mapear, respectivamente, con 32 Hz, 256 Hz, 2048 Hz y 16 kHz ya que el sintetizador puede recibir 1 Voltio/octava [8] y con 0V tenemos la frecuencia de corte mínima (16 Hz). Sin embargo, es otra asunción que habrá que certificar en el análisis.

Hemos de seguir enviando una señal de audio de entrada. Ya que esta señal de voltaje únicamente sirve para controlar los valores de los parámetros frecuencia y resonancia. Hemos realizado los dos tipos de grabaciones con la *sweep* estática y con la *sweep* variable.

Además, previamente hay que discretizar el comportamiento del VCF con una señal de entrada con la configuración de la [Figura 3.1 del Apéndice 1](#) (filtro abierto sin procesado) y la señal *loopback*, que contiene la señal original con la aportación de la tarjeta de sonido que utilizamos para la captación. La cual se utiliza en el análisis para obtener la función de transferencia de cada ajuste o muestra (apartados 3.6 y 4.2). Es esencial que el *loopback* se grabe cada día después de la calibración del equipo. También es importante registrar el *loopback* para cada señal de control, ya que puede variar en función de la señal de entrada. Por ejemplo, para la *sweep* variable y fija, el *loopback* tiene que ser registrado con el nivel máximo (-9 dBFS), ya que es el que está referenciado. Éste se graba usando un cable *jack-jack TS desbalanceado* que sale de la tarjeta y entra en ella de nuevo, tal como se ve en la [Figura 3.2 del Apéndice 1](#), donde el *loopback* corresponde al cable negro. En esta misma imagen, el cable más claro es la señal de entrada del filtro, que sale de la tarjeta, y el gris es la salida del filtro, que retorna a la tarjeta. Más adelante mostraremos un diagrama completo del sistema de grabación usado.

Existe una diferencia entre usar un cable balanceado o no balanceado a la hora de grabar el *loopback*. Con un cable no balanceado, hay una pérdida de, aproximadamente, 6 dB entre la señal y el *loopback*. En cambio, con un cable balanceado, no hay ninguna diferencia de niveles (ganancia unitaria). Teniendo esto en cuenta, hay que enfatizar que es necesario realizar todas las grabaciones con cable desbalanceado, ya que los conectores del sintetizador son de instrumento y, por tanto, desbalanceados. Siempre podríamos haber utilizado una caja de inyección (DI) para balancear la señal, pero no queríamos contaminar las muestras/mediciones.

Hay que mencionar que en la [Figura 3.3 del Apéndice 1](#) tenemos una imagen del sistema de grabación que hemos usado para casi todas las mediciones (a la izquierda), excepto las correspondientes a la pregunta 9 (a la derecha). Básicamente, se envía la señal de prueba desde la tarjeta de sonido externa al VCF. A continuación, la salida hace el recorrido inverso, volviendo a entrar a la tarjeta. En la derecha vemos el mismo montaje, pero añadiendo la fuente de voltaje externa.

En [este link](#) se pueden encontrar todos los proyectos de *Reaper*<sup>7</sup> con las grabaciones pertinentes, así como las señales de salida registradas para cada pregunta. Obtenidas manipulando los potenciómetros tal como hemos explicado previamente en esta sección. También podemos encontrar la documentación consultada, y fotografías hechas durante el proceso de captación. En el capítulo 4, se adjunta otro enlace con el código y los resultados alcanzados.

### 3.5. Procesamiento de señales

Una vez disponemos de los registros, es primordial realizar cierto procesamiento de señales para analizar los parámetros de la frecuencia de corte y la resonancia. Éste consiste en generar el filtro inverso de la sweep exponencial con el *plugin* de *Aurora* (*Aurora Sine Sweep Generator*) y convolucionarla con la señal *procesada por el VCF* (*Aurora Convolver*). El resultado de esta convolución es la respuesta impulsional (IR) de la salida. Calculando su transformada de *Fourier*, la función de transferencia (apartados 3.6 y 4.2) y aplicando ciertas operaciones que veremos en el capítulo 4, podemos estimar los valores de la frecuencia de corte, la pendiente (y orden), el ancho de banda, etc. de cada medición. Los archivos generados tienen una nomenclatura específica, que coincide también en las salidas, pero en vez de indicar VCF, indica IR (ver [Apéndice 2](#)).

Por otro lado, el mencionado filtro inverso de la sweep exponencial se define, como la inversión temporal de la señal de entrada, ecualizado con una pendiente de 6 dB/octava [\[14\]](#). Es necesario para poder obtener las IRs para el análisis.

### 3.6. Función de transferencia

La función de transferencia describe la relación entre la señal de salida y la de entrada en términos de voltaje, tal como se muestra en la [Figura 3.4 del Apéndice 1](#). Se puede descomponer en magnitud y fase [\[15\]](#).

En el análisis frecuencial de sistemas representa la amplitud de la salida en función de la respuesta en frecuencia de la señal de entrada [\[16\]](#). En nuestro caso, la función de transferencia de una salida es la transformada de *Fourier* (FFT) de la IR de la medida ponderada entre la FFT de la IR de la señal *loopback*. De su resultado podemos calcular la magnitud y la fase. La magnitud es la característica que utilizamos para calcular los valores de cada parámetro en las muestras registradas.

---

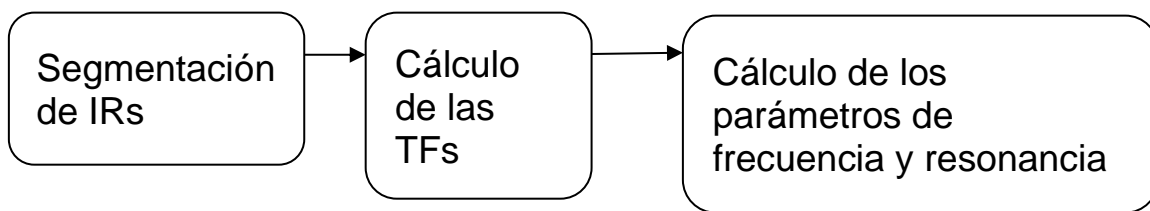
<sup>7</sup> <https://www.reaper.fm/download.php>



Para este procedimiento también obtenemos una serie de archivos, para la que definimos una nomenclatura concreta. En vez de empezar con VCF o con IR, las figuras generadas empiezan con las letras TF (*Transfer Function*). En el caso de la sweep variable, añadimos otra subcadena de caracteres en el nombre (*ampmax*, *ampmid* o *ampmin*) que indica la sweep de amplitud máxima, media o mínima, respectivamente. Podemos ver ejemplos en el [Apéndice 2](#).

### 3. 7. Obtención de los Datos

En el siguiente diagrama de bloques vemos, muy esquemáticamente, los pasos que seguiremos para llegar a los resultados obtenidos:



Recordemos que cada medida contiene varias IRs, como por ejemplo las IRs que provienen del tren de *sweeps* estáticas y variable (preguntas 1, 2, 3 y 4), por ejemplo, *IR\_16F\_allR\_sweepstat* o bien *IR\_64F\_2R\_sweepvar*. En el primer caso, el archivo contiene 11 IRs, una para cada ajuste de resonancia. En el segundo caso, cada archivo contiene 3 IRs con distinta amplitud. La explicación del método aplicado se encuentra en el [Apéndice 5](#).

Una vez separamos las IRs, hemos comprobado que no hay diferencia de magnitud entre la salida y la entrada (la IR de salida del VCF y IR de la señal *loopback*). Los resultados de esta comparación entre *output* e *input* (solo se incluyen los que conllevan un cambio respecto a los anteriores) están disponibles, en el [Apéndice 5](#). Se aprecia un patrón que se repite en todos los casos, según el cual a medida que la frecuencia de corte aumenta, la diferencia entre el nivel de magnitud media entre la entrada y la salida disminuye. Es posible que haya algún tipo de mecanismo interno del VCF para compensar la pérdida de energía por el filtrado con el cual, cuanto más se filtra la señal de entrada, más nivel medio tenemos a la salida con el fin de mantener el nivel de sonoridad. Básicamente, la magnitud se incrementa en la banda pasante del filtro, y este incremento es mayor cuanto más se estrecha por el efecto del filtro.

Para simplificar este problema, hemos normalizado esta ganancia para que la función de transferencia esté alrededor del nivel unitario (0 dB), la función de transferencia muestra la relación entre la entrada y la salida, y queremos que la diferencia entre ambas sea mínima, y poder ver el comportamiento en los mismos niveles. Para ello, reducimos la magnitud de las 2 TF (la del *bypass* y la del *output*) hasta que ambas estén a 0 dB. Necesitamos conocer la región donde la magnitud es plana y su nivel. Ésta la calculamos como la media del nivel de la señal en la región comprendida entre el punto en el que deja de aparecer el ruido inicial (20 Hz) y la parte en la que la TF empieza a bajar de nivel. Una parte de esta región es común en todas las TF (de 20 a 45 Hz). Para la referencia podemos usar un rango mayor (de 500 a 2000 Hz) ya que ésta es plana en casi todo su espectro. Esta operación la realizaremos en el *MAIN.py*, antes de llamar a los *scripts* en los que calculamos los valores de los parámetros de la resonancia y la frecuencia de corte.



Antes de reducir esta ganancia, hay que calcular las funciones de transferencia (TFs). Para ello, el primer paso es importar las IRs necesarias (medida y loopback, *IR\_input* y *IR\_output* respectivamente) para calcular la función de transferencia y, con ella, los parámetros del filtro, en un nuevo *script* (llamado *Transfer\_Function*). Éste contiene un código que calcula una única función de transferencia (TF).

Estas 2 IRs tienen que estar perfectamente alineadas, es decir, con el impulso en las mismas muestras (en dominio temporal), para que no haya desfases entre ellas, y han de tener el mismo tamaño, para realizar el cálculo. Para alinear las IR, encontramos la posición por índice del impulso de cada una, usando una librería de *Python*, *numpy*<sup>8</sup>. Concretamente, con *argmax()* i *abs* (únicamente nos interesa la parte positiva) calculamos la diferencia entre el índice de los dos picos detectados y recortamos la señal de la IR más retrasada. A continuación, nos aseguramos de que las dos señales tienen el mismo tamaño, recortándolas para que tengan el tamaño de la señal de la IR de menor duración. Además, hay que tener en cuenta que el algoritmo del cálculo de la FFT que hemos utilizado solo funciona si el tamaño es par y positivo de modo que también hay que asegurarse de ello. Para evitar valores de *infinito* o *nan* en el cálculo de la TF, hacemos una adición del valor más pequeño que puede generar el sistema (*np.finfo()*<sup>9</sup>).

Para calcular las FFTs de las dos IRs y la TF (sección 3.6 y [Ecuación 3.1 del Apéndice 3](#)), usaremos los algoritmos FFT. Además, usaremos el algoritmo *CartesianToPolar* para acceder a la magnitud y la fase (en el capítulo 4 veremos en que librería se encuentran estos algoritmos). Finalmente, convertimos la magnitud a decibelios, usando [Ecuación 3.2 del Apéndice 3](#), donde *TF\_mag* es la magnitud lineal. Es importante mencionar que el algoritmo de la FFT utilizado sólo recibe un parámetro, el tamaño de la FFT, y este lo asignamos a la duración de las IRs. Además, para el cálculo usamos una ventana rectangular, dado que no necesitamos hacer *overlap*.

Para hacerlo de manera más eficiente, se ha automatizado y generalizado este código, es decir, para procesar todas las IRs y obtener las estimaciones de los parámetros en cualquier muestra. Para ello, se ha utilizado la librería *Pathlib* [17], el código completo está disponible en el *script* *MAIN.py*. Gracias a *Pathlib* obtenemos una lista de todos los archivos *.wav* que hay en el directorio, y clasificamos las salidas en 4 listas (*Loopback*, *Bypass*, *\_OR\_*, *otros*). En *\_OR\_* van todos los archivos sin resonancia, que se usarán para calcular los parámetros de la frecuencia en el *script* correspondiente. La estimación de la frecuencia de corte y la pendiente (orden del filtro) se realiza con señales que no contienen resonancia, porque ésta tiende a deformar el filtro y, por tanto, la frecuencia de corte y la pendiente se ven afectados. Por otro lado, en la lista de *otros*, encontramos los archivos con resonancia, que se usarán en el *script* correspondiente para calcular los parámetros de la resonancia. En las otras dos listas (*Loopback* y *Bypass*) únicamente hay dos archivos (una en cada), que se usan para calcular las funciones de transferencia.

Después de esta clasificación, simplemente llamamos a las funciones de *transfer\_function()*, *frequency()* y *resonance()*, cada una definida en el *script* correspondiente. Para la salida en *bypass*, únicamente hay que llamar una vez a la función *transfer\_function*, ya que solo hay una en cada directorio de IRs. Para ejecutar los archivos *Frecuencia.py* y *Resonancia.py*, recursivamente procesamos los ficheros de las salidas correspondientes, calculamos la TF y estimamos los parámetros correspondientes a cada salida.

---

<sup>8</sup> <https://numpy.org/>

<sup>9</sup> <https://numpy.org/doc/stable/reference/generated/numpy.finfo.html>

Finalmente, guardamos los parámetros más importantes de cada una de estas dos funciones (la frecuencia de corte y la magnitud máxima de la campana de la resonancia) con tal de mostrarlos y conocer si el comportamiento control de cada parámetro del VCF es lineal, logarítmico o exponencial (sección 4.2).

## 4. Resultados

En esta sección vamos a analizar las grabaciones obtenidas en el apartado anterior, con el fin de caracterizar el comportamiento del filtro. A continuación, mostraremos los pasos que hemos seguido y los conceptos que hemos analizado. Esto nos sirve como guía para esta sección:

### Análisis global

- Estimación de la THD con el sistema de muestreo calibrado
- Estimación de la SNR
- Estimación del ruido de fondo
- Estimación del rango dinámico

### Caracterización paramétrica del filtro

- Análisis del control frecuencia
  - Tipología del filtro
  - Rango en frecuencia de corte
  - Desviación en frecuencia de corte
  - Orden del filtro
  - Respuesta del control de frecuencia: lineal, logarítmico, exponencial.
- Análisis de control de resonancia
  - Factor Q y ancho de banda para cada frecuencia central
  - Amplitud para cada nivel de resonancia
  - Respuesta de control de resonancia: lineal, logarítmico, exponencial
  - Rango de la resonancia
- Análisis de los controles con *sweeps* variables
- Análisis de los controles por *Control Voltage* (CV)
  - Resultados con diferentes niveles de amplitud de la señal de control.
  - Respuesta de los controles: lineal, logarítmico, exponencial.

A continuación, toda la documentación consultada, así como todos los resultados que se han generado y el código que se ha usado para obtener dichos resultados se adjunta en el siguiente [enlace](#).

### 4.1. Análisis global

Hemos usado las grabaciones correspondientes a las salidas del VCF con las señales sinusoidales (tonos puros) como señal de entrada. Recordemos que estas muestras se captaron en modo *bypass*, ajustando la frecuencia de corte al máximo y la resonancia al mínimo.

En las [Figuras 4.1 a 4.8 del Apéndice 1](#), observamos que hay una diferencia de unos 3 dB entre la entrada y la salida. Y cuanto más alta es su frecuencia, mayor es la diferencia. Esto es debido a que y a que el VCF tiene un comportamiento no lineal en la banda pasante y presenta una tendencia a reducir el nivel de las frecuencias agudas. También muestra que hay algo de ruido de fondo y cierta distorsión armónica.

### Estimación de la THD con el sistema de muestreo calibrado

La Distorsión Armónica Total (THD) es una medida que indica la cantidad de distorsión que introduce un sistema [18]. La THD consiste, esencialmente, en la deformación de la señal original a causa de la aparición de armónicos (frecuencias múltiples de la fundamental). La causa de estos efectos es el paso de la señal a través de un sistema no lineal como puede ser el filtro controlado por tensión. El THD aparece en dos magnitudes, corriente (THDI) y voltaje (THDV). En nuestro caso, obviamente, se trata del segundo tipo. La THD normalmente se define para 1 kHz.

Las [Figuras 4.1 a 4.8 del Apéndice 1](#) representan el espectro frecuencial de la entrada y la salida. Podemos ver claramente un ejemplo de THD en la última de ellas (el armónico generado por la distorsión se encuentra en 20 kHz). En la [Figura 4.9 del Apéndice 1](#), podemos observar el efecto que tiene la THD, la onda verde hace referencia a la señal de entrada. La roja es el armónico que aparece al pasar por el filtro y la negra es la señal de salida [19]. Se puede observar claramente que la señal de salida está deformada respecto a la entrada.

La THD se expresa como el porcentaje del cociente de la suma del nivel de cada armónico respecto al nivel de la frecuencia fundamental [19] (ver [Ecuación 4.1 del Apéndice 3](#)). Dado que tenemos diferentes señales de entrada, podemos calcular la THD para cada uno de ellos y después hacer la media aritmética para conocer el porcentaje de THD del filtro. No obstante, en las [Figuras 4.1 a 4.8 del Apéndice 1](#) vemos que únicamente aparece distorsión armónica para la señal de 10 kHz.

Dado que el cálculo se produce en dominio lineal, no logarítmico, es preferible convertirlos a magnitudes lineales, tales como voltaje o potencia. En este caso, lo más sencillo es convertir los dBFS en dBu y de dBu a Voltios, y lo hacemos de la siguiente manera [20] (ver [Ecuaciones 4.2 y 4.3 del Apéndice 3](#)). En la primera ecuación, sumamos 10 dB ya que éste es el nivel máximo que podemos obtener a la salida, debido al uso de cables desbalanceados, que causan una diferencia de 6 dBs entre entrada y salida. El resultado obtenido es el siguiente:

$$THD = \frac{-64 \text{ dBFS}}{-15 \text{ dBFS}} = \frac{-54 \text{ dBu}}{-5 \text{ dBu}} * 100 = \frac{0.000155 \text{ V}}{0.043 \text{ V}} * 100 = 0.35\%$$

Las [Figuras 4.1 a 4.8 del Apéndice 1](#) muestran que la THD del filtro es muy baja. Dado que hemos tomado muestras de frecuencias específicas, no podemos determinar con exactitud a partir de qué frecuencia se genera distorsión. Para ello podemos observar también las sweeps. Observando algunas de las salidas obtenidas con sweep estática o variable hemos visto que, concretamente, la distorsión armónica aparece a partir de los 2 o 3 kHz, tal como vemos, por ejemplo, en la [Figura 4.10 del Apéndice 1](#). En este caso, no hemos observado las IRs de las salidas, simplemente el espectro frecuencial de éstas.

## Estimación de la SNR

La *Signal to Noise Ratio* (SNR) es la diferencia entre el nivel *Root Mean Square* (RMS)<sup>10</sup> de la señal y el nivel RMS del ruido que aporta el sistema y se define en decibelios. En las [Figuras 4.1 a 4.8 del Apéndice 1](#), en la salida se aprecia un nivel de ruido inexistente en la entrada. Generalmente, cuanto mayor es la SNR, tenemos mayor rango dinámico y menos ruido de fondo. En las muestras, también hemos captado el ruido del filtro, por lo que hemos creado un código en *Python* que importa los archivos del *bypass* correspondiente y el ruido de fondo, ejecutando los cálculos necesarios (éste es el *script* llamado *SNR\_DR.py*).

Matemáticamente, la SNR se calcula con la [Ecuación 4.4 del Apéndice 3](#). Dado que se mide en decibelios, esta se convierte en la [Ecuación 4.5 del Apéndice 3](#), que a su vez se convierte en la [Ecuación 4.6 del Apéndice 3](#) si trabajamos con voltajes. Para realizar estos cálculos podemos usar la librería de *Essentia*<sup>11</sup> [21], la cual es un conjunto de algoritmos en código abierto que se usan en análisis musical, sonoro y técnico de señales de audio. Para este cálculo, usaremos los algoritmos de *Monoloader* (para importar los audios) y *RMS* (para calcular los niveles RMS de la salida y del ruido). De esta librería también usamos los algoritmos para necesarios calcular la función de transferencia (secciones 3.6 y 3.7).

La SNR del aparato es el resultado obtenido más 18 dB. Recordemos que la señal de entrada está referenciada a -9 dBFS de pico, es decir, -12 dBFS RMS. Por otro lado, la señal de entrada podría, sin estar calibrada, llegar a 0 dBFS, con lo cual tenemos este espacio de 12 dB entre la señal que enviamos y el máximo nivel. Hay que recordar, además, que los cables desbalanceados introducen una diferencia de unos 6 dB entre la entrada y la salida. Podemos ver los resultados en la Tabla 4.1:

	<i>Bypass_tone</i> 100	<i>Bypass_tone</i> 440	<i>Bypass_tone</i> 1k	<i>Bypass_tone</i> 10k
<i>RMS_tone</i> (dB)	-16.33	-16.24	-16.20	-18.47
<i>RMS_noise</i> (dB)	-77.27	-77.27	-77.27	-77.27
<i>SNR</i> (dB)	78.94	79.03	79.07	76.80

Tabla 4.1.: Efecto de la frecuencia del tono sobre la SNR

La SNR medio del filtro es:  $(78.94 + 79.03 + 79.07 + 76.80) / 4 = \mathbf{78.46 \text{ dB}}$

Podemos decir que es un nivel aceptable de SNR, ya que quiere decir que hay una diferencia de casi 80 dB entre la señal y el ruido. Esto implica que la señal es perfectamente reconocible y el ruido apenas es audible, y cuadra con las [Figuras 4.1 a 4.8 del Apéndice 1](#), donde vemos claramente la forma del tono puro a pesar del ruido.

<sup>10</sup> <https://unicrom.com/valor-rms-promedio-pico/>

<sup>11</sup> <https://github.com/MTG/essentia>

También observamos que, a mayor frecuencia, el nivel de la señal disminuye, con lo cual (dado que el ruido se mantiene constante) la SNR también disminuye.

- Estimación del ruido de fondo

Para estimar el ruido de fondo, el procedimiento más intuitivo consiste en restar las dos señales (el *bypass* y el tono de entrada correspondiente). El resultado obtenido es el ruido. Para su estimación, hemos implementado un *script* (*Ruido\_Fondo.py*) que computa la FFT de ambas señales para sustraer la magnitud de los espectros frecuenciales, o, que es lo mismo, calcular la división logarítmica de ellas. Para ello utilizaremos *Essentia* [21].

En la [Figura 4.11 del Apéndice 1](#) podemos ver los resultados, donde observamos que en las frecuencias en las que se encuentra el tono puro, hay menos ruido (excepto en la última figura, en la que el nivel del ruido de fondo aumenta). En las [Figuras 4.1 a 4.8 del Apéndice 1](#) se aprecia que el ruido es constante en todas las frecuencias. Sin embargo, en las altas frecuencias aparece mayor nivel. Aunque cada módulo sea independiente de los otros, un poco del ruido electrónico interno de cada uno se acaba sumando a la salida, por lo que estas medidas contienen el ruido de fondo de los VCO y el resto de los módulos.

- Estimación del rango dinámico

El rango dinámico es la diferencia de nivel de presión sonora existente entre los niveles más bajos y los más altos de una señal [22]. También se mide en decibelios y es mejor cuanto mayor es su valor. Para este cálculo hemos usado el *software* *Essentia*, reutilizando el script *SNR\_DR.py*. Por otro lado, también es necesario añadir 18 dB, por el mismo motivo que había que hacerlo con el cálculo de la SNR (visto en la página anterior). Podemos ver los resultados en la Tabla 4.2

	<i>Bypass_tone</i> <i>100</i>	<i>Bypass_tone</i> <i>440</i>	<i>Bypass_tone</i> <i>1k</i>	<i>Bypass_tone</i> <i>10k</i>
<b><i>RMS_noise</i></b> <b><i>(dB)</i></b>	-77.27	-77.27	-77.27	-77.27
<b><i>Peak</i></b> <b><i>(dB)</i></b>	-12.95	-12.88	-12.86	-15.16
<b><i>Dynamic</i></b> <b><i>Range</i></b> <b><i>(dB)</i></b>	82.31	82.39	82.41	80.11

Tabla 4.2: THD: Efecto de la frecuencia del tono sobre la THD

$$DR_{total} = (82.31 + 82.39 + 82.41 + 80.11) / 4 = \mathbf{81.805 \text{ dB}}$$

Vemos que el resultado es muy próximo al de la SNR, concretamente, hay 3 dB de diferencia, que es la diferencia entre el nivel de pico y el de RMS de una señal. También vemos que hay el mismo patrón en cuanto al nivel del *bypass*. Sin embargo, el rango dinámico es más bajo en las últimas octavas.

## 4.2. Caracterización paramétrica del filtro

Como hemos mencionado, el VCF tiene dos parámetros principales, la frecuencia de corte del filtro y la resonancia. No obstante, hay diversas características secundarias. Éstas son: la pendiente del filtro (cuando se llega a la frecuencia de corte, el sonido no es repentinamente eliminado, sino que es un proceso gradual), el ancho de banda de la resonancia (que frecuencias ocupa la campana de resonancia, cuyo rango se indica con los parámetros  $f_1$  y  $f_2$ , en Hercios(Hz)), el factor  $Q$ <sup>12</sup> (indica la forma de la señal en la resonancia: cuanto más alto es este valor, más estrecha es la forma), la magnitud de la resonancia (la diferencia entre el máximo nivel de esta resonancia y el nivel de la referencia), la frecuencia central y la de resonancia (en qué frecuencia se encuentra el nivel máximo de la resonancia, no siempre coinciden).

- **Análisis del control frecuencia**

En esta subsección hablaremos del *script Frecuencia.py* donde calculamos la pendiente y la frecuencia de corte de cada salida. Una vez hechos los cálculos necesarios, hay que guardar los gráficos resultantes en formato *PNG* con el nombre adecuado (ver [Apéndice 2](#)) y los valores en un documento en formato *JSON* (*JavaScript Object Notation*<sup>13</sup>), con el mismo nombre.

La frecuencia de corte de un filtro se define como aquella en la que el nivel estable de la señal desciende 3 dB. Para estimarla, la opción más precisa es hacerlo geométricamente usando el *bypass* (la referencia): atenuando la magnitud de la TF del *bypass* 3 dB y buscando el punto de corte entre la TF de la salida y la del *bypass*. Para encontrar la intersección entre las dos gráficas, podemos usar la siguiente [línea de código](#). Primero, calcula la diferencia de magnitudes y los signos correspondientes usando *np.sign*<sup>14</sup>. En este punto, es necesario filtrar la señal, ya que en los extremos del espectro hay mucho ruido y si queremos generalizar este código, solo puede haber un punto de corte. Aplicando *np.diff* i *np.argwhere*, conocemos las posiciones donde cambia el signo (cosa que ocurre cuando ambas gráficas se cortan).

### Tipología del filtro

Observando los resultados (en la [Tabla 4.3 del Apéndice 4](#)) apreciamos que se trata de un filtro pasa-bajos (LPF), tal como esperábamos, ya que deja pasar las frecuencias inferiores a la frecuencia de corte y cancela las que están por encima, con un cierto pendiente decreciente. La pendiente es positiva, si fuera un filtro pasa-altos (que deja pasar las frecuencias altas y elimina las bajas) la pendiente sería negativa, ya que el nivel crecería en vez de decrecer, como es en nuestro caso.

---

<sup>12</sup> Es la inversa del ancho de banda

<sup>13</sup> <https://es.wikipedia.org/wiki/JSON>

<sup>14</sup> <https://numpy.org/doc/stable/reference/generated/numpy.sign.html>



### Rango en frecuencia de corte

El rango de la frecuencia de corte lo podemos conocer observando los resultados de las preguntas 2,3 y 4 (sección 3.1), obtenidos con la *sweep* estática como señal de entrada (ya que ésta es la señal de referencia), con el potenciómetro de la frecuencia de corte al mínimo y en el penúltimo punto (el máximo es el *bypass*). Estos valores estimados son los siguientes:

*TF\_16F\_0R\_sweepstat.png*: 27.41 Hz

*TF\_8192F\_0R\_sweepstat*: 7802.20 Hz

Cabe mencionar que, dado que el *bypass*, es la referencia, no podemos calcular su frecuencia de corte con el método usado con las otras. Sin embargo, observando su función de transferencia, podemos ver que ésta es, aproximadamente, de 12.6 kHz, con lo que es menor a la marcada en la serigrafía. El rango teórico del potenciómetro de la frecuencia de corte es de 16 Hz a 16 kHz. Sin embargo, después de observar los resultados obtenidos, vemos que, en realidad, este rango va desde los 27 Hz hasta los 13 kHz (si tenemos en cuenta la frecuencia de corte observada en el *bypass*, punto máximo del potenciómetro correspondiente) de modo que hay cierta desviación, sobre todo en el extremo agudo.

### Desviación de la frecuencia de corte

Hemos visto que hay cierta desviación en cuanto al rango, pero también la hay en cada frecuencia de corte, aunque esta es muy pequeña. En el extremo inferior, tenemos una desviación de 11 Hz. Mientras que, a altas frecuencias, la desviación alcanza los 1000 Hz. Vemos que en bajas frecuencias (32 y 64 Hz) está desviación respecto a nuestra asunción inicial es mínima.

En la [Tabla 4.3 del Apéndice 4](#), podemos ver los valores de cada ajuste del potenciómetro de frecuencia. Donde vemos que, a mayor frecuencia, se produce una mayor desviación. Este comportamiento provoca que, cuanto más se quiere filtrar el sonido que entra al filtro, menos control se tiene sobre la frecuencia en la cual empezamos a eliminarlo. También vemos que la frecuencia de corte estimada es superior a la asumida en bajas frecuencias, pero que, a partir del punto medio del potenciómetro (512F), la estimada está por debajo de la asumida.

### Orden del filtro

El orden del filtro depende de la pendiente de éste. Para estimar la pendiente de una salida, usamos las siguientes [líneas de código](#). Se define sustrayendo el nivel de la magnitud en la frecuencia de corte menos el nivel de la magnitud en una y en dos octavas superiores y calculando su media. Dado que no hay ninguna frecuencia de corte que supere la frecuencia de *Nyquist* (24 kHz), podemos calcular el pendiente de la primera y de la segunda octavas, posteriores a la frecuencia de corte, de todos los resultados. Para conocer su valor, calculamos la media de los valores encontrados para cada frecuencia de corte, la cual nos da un valor de 16.84 dB/octava, el cual está casi 8 dB por debajo del valor teórico<sup>15</sup>. Este valor corresponde a un filtro de 3o orden (aunque inexacto, dado que un filtro de 3o orden tiene una pendiente de 18 dB/oct).

---

<sup>15</sup> Según las especificaciones técnicas, el VCF tiene una pendiente teórica de 24 dB/octava.



En la [Tabla 4.3 del Apéndice 4](#), podemos ver los valores de frecuencia de corte y pendiente de cada TF. En la primera columna tenemos el nombre del archivo correspondiente, simplificado. Por ejemplo, el primero sería *Results\_16F\_0R\_sweepstat.json* (ver [Apéndice 2](#)).

En la misma [Tabla 4.3](#), observamos los valores de la desviación estándar (*stdev*), la cual se calcula con la [Ecuación 4.7 del Apéndice 3](#), donde  $x_i$  es el valor de cada pendiente y *mean* es la media aritmética de todas las pendientes. Ésta nos indica cuán alejados de la media están las muestras elegidas, o cuánta dispersión hay en el conjunto de medidas. Vemos que, hasta los 32 Hz, la pendiente está por encima de la media. En cambio, a partir de 64 Hz, la pendiente es menor que la media. Además, cuanto mayor es la frecuencia de corte, menos brusca es la eliminación del sonido (comportamiento no lineal), es decir, menor es la pendiente. En términos generales, en las regiones más graves hay más variación que en la zona central o aguda del espectro.

Cabe decir que no calculamos la media ni la desviación estándar para todos los valores, sino sólo para aquellos resultados cuyos valores tienden a concentrarse alrededor de un punto medio, como la pendiente del filtro o la magnitud máxima de la resonancia en varias frecuencias. En los demás parámetros (frecuencia de corte, central o de resonancia, factor Q y ancho de banda) hay mucha diferencia entre unos valores y otros por lo que calcular la media sería manipular en exceso la realidad. Podemos ver un ejemplo de los comportamientos que hemos explicado en la Figura 4.12, la cual corresponde con la posición central del potenciómetro de frecuencia de corte.

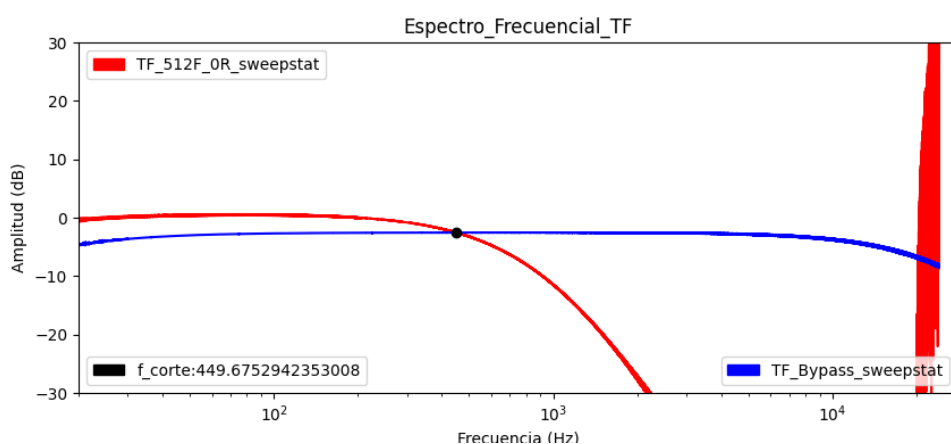


Figura 4.12: TF\_512F\_0R\_sweepstat: TF de la salida obtenida con el potenciómetro de frecuencia en el punto medio, el de resonancia en el mínimo y la *sweep* estática como señal de entrada.

#### Respuesta del control de frecuencia: lineal, logarítmico, exponencial.

Conocer el comportamiento de los controles equivale a mapear las posiciones del potenciómetro seleccionadas con las frecuencias de corte reales que hemos encontrado. Para ello, dibujamos todas las frecuencias de corte en una sola *figura* y observamos su trayectoria frente al número de ajustes del control (*MAIN.py*).

Primero declaramos una nueva lista (*fcorte\_lista*) y recursivamente recorremos la lista de archivos con resonancia 0, y almacenamos la frecuencia de corte de cada archivo en esta nueva lista. Cuando acaba el bucle, declaramos un vector de etiquetas, en el cual estará cada posición del potenciómetro (16, 32, 64, etc). Estos números representan la frecuencia de corte del filtro, asumida, que corresponde a cada posición.

No aparecen en la serigrafía del aparato, sino que las definimos para el proyecto. En el eje y vemos la correspondencia con las frecuencias de corte estimadas.

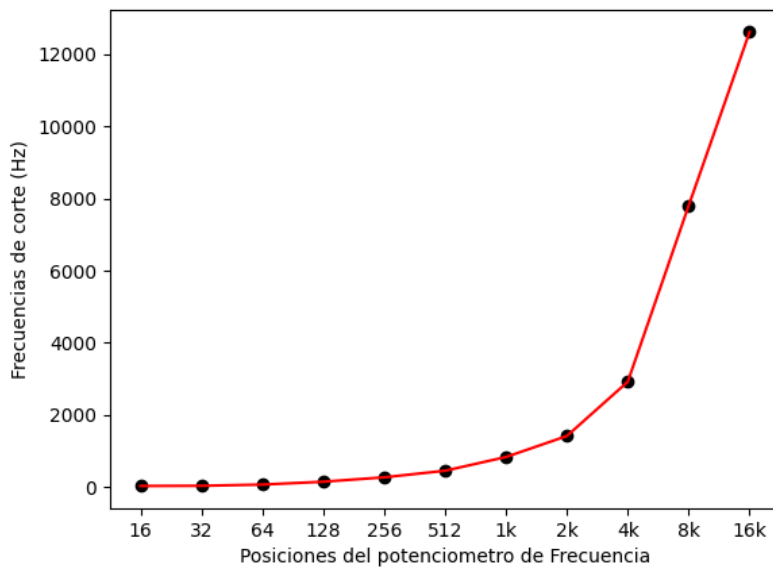


Figura 4.13: Frecuencias\_de\_corte\_sweepstat: mapeo de cada posición del potenciómetro a la frecuencia de corte estimada, con la *sweep* estática como señal de entrada.

Finalmente, dibujamos cada valor respecto al vector de etiquetas. El resultado lo vemos en la Figura 4.13, la cual nos muestra que el potenciómetro de frecuencia de corte tiene un comportamiento exponencial. Esto cuadra con el hecho que, para nosotros, cada posición equivale a una potencia de 2, desde  $2^4$  (16 Hz), hasta  $2^{14}$  (16 kHz), de modo que, por ejemplo, el punto medio del potenciómetro aún se corresponde con una frecuencia de corte baja. En conclusión, vemos un comportamiento que cuadra con la serigrafía y las correspondencias que hemos asumido.

- **Análisis de control de resonancia**

Para estimar los parámetros de la resonancia, implementamos un nuevo *script*, *Resonancia.py*. El primer paso es estimar la frecuencia central y el *gain* (valor máximo de la campana de resonancia). Éstos nos servirán para encontrar los otros parámetros. Como en *Frequency.py*, una vez acabados todos los cálculos, salvamos los gráficos y los valores estimados en formato *PNG* y *JSON* respectivamente. Todos los resultados de este análisis están disponibles en forma de tabla ([Tabla 4.4 del Apéndice 4](#)).

#### Factor Q y ancho de banda para cada frecuencia central

Para estimar el factor Q, hemos aplicado la [Ecuación 4.8 del Apéndice 3](#), donde *fres* se refiere a la frecuencia de resonancia y se calcula de manera directa a partir de *f1* y *f2* (ver [Ecuación 4.8 del Apéndice 3](#)). Por otro lado, *f1* y *f2* determinan el ancho de banda y se calculan de manera parecida a cómo calculamos la frecuencia de corte. Éstas se definen como las frecuencias que están 3 dB por debajo de la magnitud máxima de la campana de resonancia. Para calcularlas, hemos escalado el *bypass* para que coincida con el valor de la magnitud máxima de la campana de resonancia, a lo que, seguidamente, hemos restado 3 dB, y calculado *f1* y *f2* como la intersección de la salida con el *bypass* escalado.

En la [Tabla 4.4 del Apéndice 4](#) podemos encontrar los resultados para estos 6 parámetros ( $f_1$ ,  $f_2$ , frecuencia central, frecuencia de resonancia, nivel de magnitud máxima de la campana de resonancia (*gain*) y el factor  $Q$ ). Hay varias observaciones que podemos hacer y un patrón que es común para todas las frecuencias de corte. En resonancia muy baja, solo hay un punto de corte, o varios puntos muy cercanos, ya que el *bypass* está por debajo de la salida. Por eso  $f_1$  y  $f_2$  coinciden o están muy cerca una de la otra, y eso hace que el factor  $Q$  sea elevado, porque estamos dividiendo la frecuencia de resonancia (que coincide con las  $f_1$ ,  $f_2$ ) entre el ancho de banda ( $f_2 - f_1$ ), el cual es un valor muy pequeño. Sin embargo, a partir de 1k, en resonancias bajas, tenemos un primer punto de corte ( $f_1$ ) muy bajo, dado que se encuentra en la zona ruidosa del inicio de la señal.

Por otro lado, hemos visto que el ancho de banda aumenta cuando el pico de la resonancia está por encima de los 3.5 dB. En ese momento, el *bypass* está por encima del nivel estable de la salida una vez hemos sumado el pico y restado 3 dB. En el punto del potenciómetro en el cual el máximo de la resonancia está por encima de los 3.5 dB, tenemos una resonancia bien definida, que varía a medida que la frecuencia de corte aumenta. Cuando estamos a 16 Hz, este punto no llega hasta que tenemos resonancia 7, y en 32 Hz, éste llega en la resonancia 4, después se estabiliza y siempre tenemos una resonancia definida a partir del 3º punto del potenciómetro de resonancia. Vemos, por tanto, que se trata de un comportamiento no lineal. También en las últimas posiciones hay un comportamiento no lineal muy claro, con lo que tampoco se cumple esta observación.

También hay conflicto cuando alcanzamos los mayores niveles de resonancia (8, 9 y 10). Como explicamos en la sección 2, cuando nos encontramos en resonancia máxima, el sistema comienza a oscilar, generando una onda sinusoidal cuyo tono está determinado por la frecuencia de corte. En la práctica eso significa que la diferencia entre la entrada y la salida es muy grande y en la función de transferencia se traduce en ruido obteniendo una respuesta frecuencial indefinida y ruidosa. Si bien con cierta concordancia con la forma que esperaríamos según la frecuencia de corte (Figura 4.14).

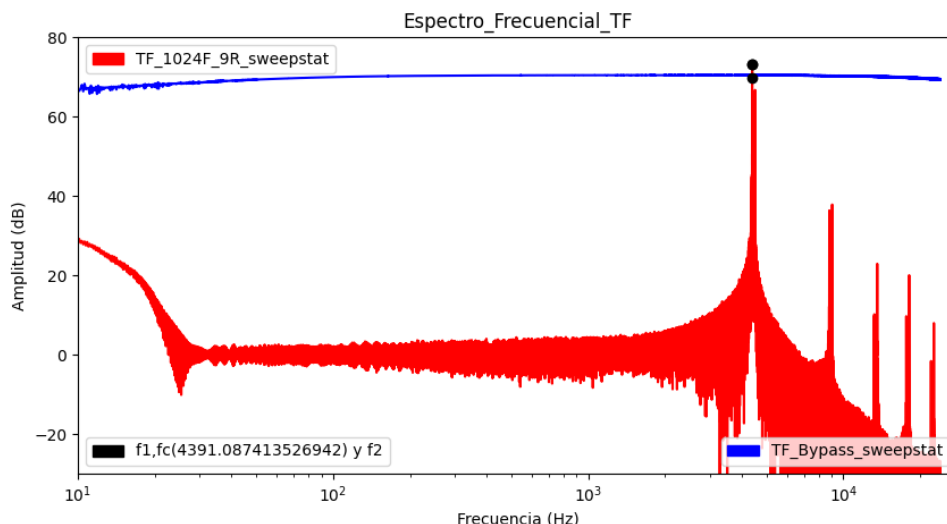


Figura 4.14: TF\_1024F\_9R\_sweepstat: Efecto de la oscilación que aparece en altas resonancias. TF de la salida obtenida con el potenciómetro de frecuencia de corte en la posición 7 y el de resonancia en la posición 9, y con la *sweep* estática como señal de entrada.

En este punto, la magnitud máxima de la campana de resonancia es muy alta y estrecha, por tanto, solo hay un punto de corte entre el *bypass* y el *output*, por lo que tenemos un comportamiento muy parecido al inicial, con  $f_1$  y  $f_2$  iguales o muy cercanas, frecuencia de resonancia igual o muy cercana a  $f_1$  y  $f_2$ , y el factor  $Q$  muy alto (que cuadra con una resonancia muy estrecha). No solo se genera una onda sinusoidal, sino que es un oscilador con cierta distorsión y que genera una onda más compleja.

A partir de  $8192F$ , cuando el potenciómetro de la frecuencia de corte está en el penúltimo punto, esta oscilación desaparece y no se aprecia ese aumento de la campana ni del factor  $Q$ . En cambio, a 16 kHz la salida se parece mucho al *bypass*, con un punto de corte. También vemos que, en estos dos ajustes del potenciómetro de la frecuencia de corte, los valores de la frecuencia central están en el límite del espectro frecuencial, por lo que la resonancia no es visible. En este punto, el filtro no solo no actúa, sino que, además, acentúa los sonidos muy agudos, ya que aparece únicamente la parte del aumento de la resonancia.

Tal como se menciona en secciones anteriores, la resonancia deforma la respuesta frecuencial del VCF. A medida que la resonancia aumenta, la frecuencia central (que equivale a la frecuencia de corte en resonancia 0), así como  $f_1$  y  $f_2$ , se van desplazando hacia la derecha, es decir, los valores son mayores.

#### Amplitud para cada nivel de resonancia

La amplitud de cada resonancia viene determinada por la diferencia de magnitud entre el máximo nivel de la campana y el nivel de la TF del *bypass*, la cual es la misma en todo su dominio, excepto en los extremos, donde se reduce y aumenta a causa del ruido e interferencias de estas dos regiones. Dado que el nivel estable del *bypass* está a 0 dB, podemos decir que la diferencia es la magnitud del nivel máximo de la campana de cada TF. Igual que hemos hecho antes, vamos a analizar estos valores, junto con los de la frecuencia central ([Tabla 4.4 del Apéndice 4](#)).

En teoría, la frecuencia de resonancia y la frecuencia central deberían coincidir o estar cerca una de la otra en todos los casos. Sin embargo, dado que la frecuencia de resonancia depende de  $f_1$  y  $f_2$ , este requisito no se cumple en todos los casos. La frecuencia central, donde se encuentra la magnitud máxima, debería estar entre la  $f_1$  y la  $f_2$ . Eso sucede a partir de un máximo mayor a 3.5 dB, donde  $f_1$  y  $f_2$  son valores coherentes. Antes de llegar a este punto, la frecuencia central en valores de frecuencia muy pequeños y con un nivel de máximo de entre 1 y 2 dB. A medida que aumentamos el ajuste del potenciómetro de resonancia, éste va aumentando y la frecuencia central se va desplazando hacia la derecha.

De cara a mapear los puntos del potenciómetro con los valores estimados de estos parámetros, vemos que, a partir de los 128 Hz asumidos (4o punto del potenciómetro de frecuencia de corte), los puntos de resonancia de 2 a 7 corresponden a valores del doble en el máximo de la campana. Por ejemplo, el 2º punto corresponde a, aproximadamente, 4 dB de máximo, el 4º corresponde a unos 8 dB, y así sucesivamente. Con el 1r punto tenemos poca resonancia y en los 3 últimos puntos (8, 9 y 10) tenemos una diferencia mucho mayor del doble, ya que estamos en un rango donde el sistema oscila. En los extremos agudos del potenciómetro de frecuencia tenemos un comportamiento peculiar: cuando este está en la penúltima posición, el máximo de la campana se encuentra, en todas las posiciones del potenciómetro de resonancia, alrededor de 4dB, mientras que, en el punto máximo de frecuencia de corte, este máximo no llega a los 2 dB. Por debajo de estos valores de frecuencia, podemos observar que cuanto más alta es la frecuencia de corte, más alto es el nivel máximo de la campana de la resonancia.

En otras palabras, cuanto menos actúa el potenciómetro de frecuencia de corte, más actúa el de la resonancia, con tal de compensar y mantener equilibrada la actuación del VCF.

### Rango de la resonancia

Para estimar el rango nos hemos fijado en los valores de pico, ya que es el parámetro más estable y el que mejor define cuando la resonancia es perfectamente notoria. Otros parámetros que también son muy importantes a la hora de caracterizar una resonancia, el ancho de banda y el factor Q, no son tan estables, como hemos explicado anteriormente, sobre todo en los extremos (con poca y mucha resonancia). Si no tenemos en cuenta los extremos, el rango de este potenciómetro va desde 0.47 dB (correspondientes a la salida *TF\_4096F\_1R\_sweepstat*) hasta 14.94 dB (correspondientes a la salida *TF\_512F\_7R\_sweepstat*). Ambas salidas se pueden observar en la [Figura 4.15 del Apéndice 1](#).

### Respuesta de control de resonancia: lineal, logarítmico, exponencial

Para conocer el comportamiento de este control, mapeamos las posiciones del potenciómetro de resonancia con los valores estimados de ésta. Hemos utilizado el mismo código que en el apartado anterior. La diferencia es que ahora consideramos los valores de magnitud máxima de la campana de resonancia, dado que, como hemos explicado anteriormente, éste es el parámetro de resonancia más estable. Sin embargo, antes teníamos un solo valor para cada frecuencia de corte. En cambio, ahora, para cada valor de magnitud máxima, hay 11 valores, correspondientes a las 11 frecuencias de corte. Para que el resultado sea más parecido al obtenido anteriormente, hemos calculado la media de cada sublista, con tal de tener un valor para cada nivel máximo de la campana de resonancia. Igual que antes, el resultado muestra una respuesta exponencial (Figura 4.16). En las dos últimas filas de la [Tabla 4.4 del Apéndice 4](#) vemos estos valores, y podemos extraer algunas observaciones interesantes.

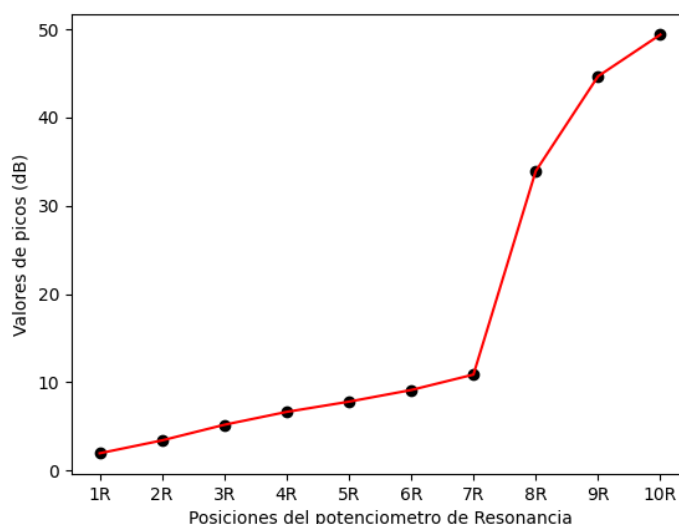


Figura 4.16: *Resonancias\_sweepstat*: Media de la magnitud máxima de la campana de resonancia con cada frecuencia de corte.

La primera es que, como hemos explicado antes, entre la 2a y la 7a posición del potenciómetro se cumple que cada posición corresponde al doble del valor de pico, aproximadamente. Sin embargo, en los extremos del potenciómetro de frecuencia de corte, esto no se cumple, con lo que la media es ligeramente inferior al doble.

No obstante, en el caso de las 3 últimas posiciones del potenciómetro (8R, 9R y 10R), la media es menor a la que debería ser, estamos distorsionando la realidad, ya que hay mucha diferencia entre las magnitudes máximas de diferentes frecuencias de corte. De modo que el valor de este punto podría ignorarse, pero, aun así, el mapeado seguiría siguiendo una evolución exponencial, es decir, sigue un comportamiento no lineal.

Para este mismo parámetro hemos calculado también la desviación estándar (en la última columna de la [Tabla 4.4 del Apéndice 4](#)) y también podemos observar un comportamiento parecido al que hemos visto con la pendiente del filtro. Para empezar, vemos que, en cualquier frecuencia de corte, cuanto mayor es la resonancia, más diferencia hay con la media, por lo que la desviación es más grande. Al igual que con la pendiente, en frecuencias bajas estamos por debajo de la media, pero a partir de 64 Hz, el valor de la magnitud máxima de la campana está por encima de la media. Por último, vemos que, entre los 128 Hz y los 4 kHz, los valores de desviación estándar son muy similares. En 8 kHz y 16 kHz, dado que los valores de pico son muy similares independientemente de la posición del potenciómetro de resonancia en la que estemos, la desviación es muy alta.

Sospechamos que el comportamiento, tanto en resonancia como en frecuencia, será exponencial si la señal de entrada es la *sweep* variable, dado que, como vemos en las Tablas 4.3, 4.4, 4.5 y 4.6 del Apéndice 4, los valores son bastante similares, en general. Sin embargo, repetiremos este proceso para el control con voltaje (CV).

- **Análisis de los controles con *sweeps* variables**

Para este apartado repetimos el mismo proceso que describimos con la *sweep* estática, únicamente difiere en el directorio que contiene las IRs. En los resultados de las [Tablas 4.5 y 4.6 del Apéndice 4](#) observamos que:

Comparando estos valores con los de la [Tabla 4.3 del Apéndice 4](#), vemos que tienen un comportamiento muy similar al obtenido con una *sweep* estática. Sin embargo, vemos dos diferencias importantes entre los diferentes niveles de amplitud de la señal de entrada. La primera es que a medida que ésta disminuye, la frecuencia de corte es inferior, excepto en bajas frecuencias de corte. Por ello el valor estimado se aleja más del valor esperado, ya que, en altas frecuencias, la frecuencia real está por debajo de la asumida. Por tanto, volvemos a ver un comportamiento no lineal, cuando variamos el nivel, el resultado es parecido, pero no igual, sino que tiene desviaciones.

También notamos que la pendiente es menor, más suave, a medida que el nivel de la señal de entrada disminuye. Esto implica que el filtro actúa menos, deja pasar más energía de la banda filtrada, con tal de compensar esta pérdida de magnitud en la entrada. Esto último podemos verlo en la media y la desviación estándar, en las que observamos que la pendiente se va reduciendo. También vemos que, por debajo de 64 Hz, la pendiente está por encima de la media y, después, está por debajo.

La segunda diferencia entre niveles de amplitud es que a medida que este disminuye, hay más ruido en la función de transferencia. En el análisis global vimos que el ruido tiene un nivel constante, es decir, se mantiene estable independientemente de la señal que entre, ya que es parte del filtro, no de la señal de entrada. Sin embargo, dado que el nivel de la señal de entrada se reduce, el SNR también lo hace, y eso provoca que el ruido sea más perceptible. Estas dos diferencias las podemos observar en la [Figura 4.17 del Apéndice 1](#), donde se muestran los tres niveles para una frecuencia de corte teórica de 64 Hz (correspondientes al 3r punto del potenciómetro de frecuencia de corte).



Respecto al control de resonancia, se repiten estas dos diferencias que hemos identificado en la frecuencia de corte. Ésta, en general, disminuye a medida que el nivel de la señal de entrada se reduce, y el ruido es más visible con amplitudes bajas. No obstante, el comportamiento es muy parecido al obtenido con la *sweep* estática. A medida que la frecuencia de corte aumenta, la resonancia se excita más tarde, y tanto  $f_1$ ,  $f_2$ , como la frecuencia central y la de resonancia tienen valores coherentes cuando la magnitud máxima de la campana está por encima de los 3.5 dB. También observamos los efectos que tiene la oscilación generada en resonancia máxima y los valores de  $f_1$  y  $f_2$ , cuando la magnitud máxima está por debajo de 3.5 dB. En ese caso solo hay un punto de corte o bien las dos funciones coinciden y la detección es más compleja. Igual que en el apartado anterior, cada ajuste del potenciómetro de resonancia se puede mapear al doble de valor en el nivel máximo de la magnitud de la campana, aproximadamente, excepto cuando la frecuencia de corte es mínima. Vemos un ejemplo de estos resultados en la [Figura 4.18 del Apéndice 1](#).

Para esta señal de entrada, también hemos calculado la media de los niveles máximos de cada campana en las diferentes amplitudes. En los resultados vemos que, efectivamente, las medias son muy próximas entre ellas, pero diferentes respecto a las obtenidas con la *sweep* estática, sobretodo en altas resonancias. Aun así, la evolución del mapeado entre las posiciones del potenciómetro y los valores reales de éstos también es exponencial. No obstante, en el caso de las últimas posiciones del potenciómetro ( $8R$  y  $10R$ ), la media calculada se corresponde con la media real, a diferencia de lo visto con la *sweep* estática.

Referente a la desviación estándar, también observamos un comportamiento similar. Para empezar, cuanto más resonancia hay, más variación tenemos, debido a que la magnitud máxima de la campana se aleja más de la media. También vemos que, a bajas frecuencias, estamos por debajo de la media y a altas, estamos por encima y con valores muy similares. A medida que crece la frecuencia de corte, crece la desviación, dado que la magnitud de la campana también crece. Como comentamos en el apartado anterior, cuanto mayor es la frecuencia de corte, más actúa la resonancia (la magnitud de la campana es mayor).

- **Análisis de los controles por Control Voltaje (CV)**

Para este apartado, el proceso es el mismo que con la *sweep* estática y la *sweep* variable con algunas modificaciones. Hemos creado un nuevo fichero, *MAIN\_CV.py*, por que, para este análisis, necesitamos dos directorios de entrada, uno para el *bypass* y *loopback* y otro para las salidas. El *bypass* y el *loopback* los obtenemos del directorio correspondiente según enviemos una señal de entrada o la otra (*sweep* estática o *sweep* variable), mientras que las salidas las obtenemos del directorio de la *pregunta\_9*. Podemos ver los resultados obtenidos en las [Tablas 4.7, 4.8, 4.9 y 4.10 del Apéndice 4](#).

En la primera, vemos los resultados obtenidos controlando la frecuencia de corte con una señal CV y la *sweep* estática como señal de audio de entrada. La pendiente es muy similar con respecto a los valores de la [Tabla 4.3 del Apéndice 4](#) (control manual), excepto con el voltaje máximo ( $10V$ ), con el cual el pendiente se reduce mucho. En cambio, la frecuencia de corte se aleja bastante de la asunción que hicimos en la sección 3. Esto implica que, posiblemente, el VCF no recibe  $1V/octava$  y que el máximo de tensión que podemos enviar no es  $10V$ , dado que la frecuencia de corte en este punto es  $10\text{ kHz}$  menor que la máxima frecuencia de corte teórica ( $16\text{ kHz}$ ).

En la primera columna podemos ver el mapeado que asumimos al principio entre el nivel de voltaje que enviamos y la frecuencia de corte establecida (sección 3.4). Con 1V, la frecuencia de corte estimada se acerca mucho a la asumida (aunque menos que con el control manual). Sin embargo, Con 4 y 7V, el resultado se acerca más a la octava anterior (por ejemplo 162 Hz se acerca más a 128 Hz que a 256 Hz). Vemos un ejemplo de estas funciones de transferencia en la Figura 4.19

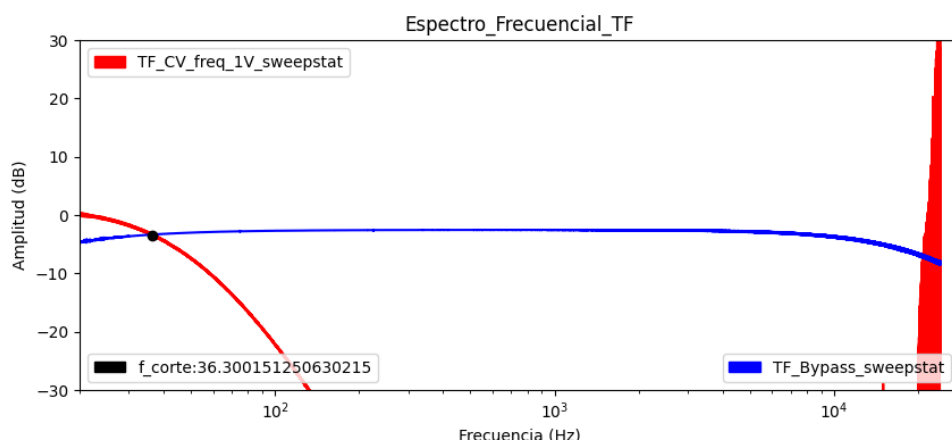


Figura 4.19: TF\_CV\_freq\_1V\_sweepstat: TF de la salida obtenida enviando 1V al potenciómetro de frecuencia de corte y la *sweep* estática como señal de audio de entrada.

Dado que la pendiente es muy similar a la obtenida con control manual, cabe esperar que la desviación estándar y su evolución también sea muy similar a la de la [Tabla 4.3 del Apéndice 4](#), lo cual, viendo los resultados de la [Tabla 4.7 del Apéndice 4](#), se demuestra. Ésta corresponde a valores de salidas sin resonancia, con el CV aplicado al parámetro de frecuencia de corte. En cuanto a la resonancia, observamos un comportamiento bastante parecido al que hemos obtenido con el control manual. Hasta que el nivel máximo de magnitud de la campana no supera los 3.5 dB, no aparece la campana de resonancia bien definida, ya que la estimación de  $f_1$  se complica. En otras palabras, enviando tan solo 1V de tensión, no podemos caracterizar este potenciómetro. Sin embargo, en el segundo punto que hemos sampleado (4V) se observa claramente la campana, aunque la estimación de la  $f_1$  es compleja, dado que el nivel estable de ambas TF coincide. No obstante, hay una diferencia importante, que hace referencia al mapeado: cada punto del potenciómetro no coincide con el doble de la magnitud máxima de la campana, sino que éste último es un poco menor que el valor de la posición, excepto con voltajes altos. Vemos un ejemplo de estas TFs en la Figura 4.20:

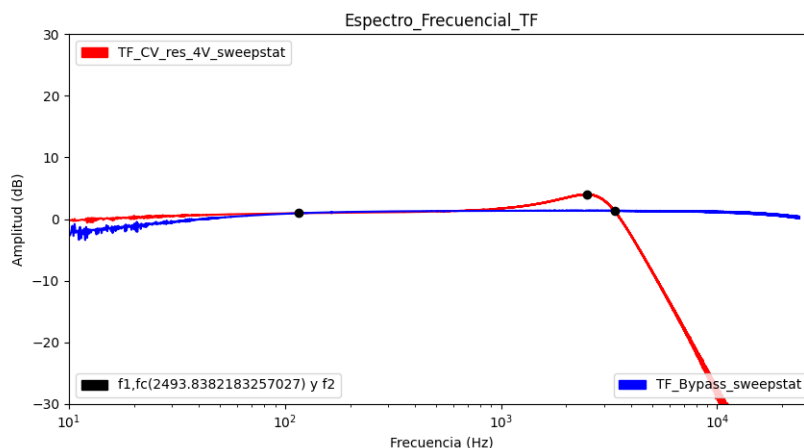


Figura 4.20: TF\_CV\_res\_7V\_sweepstat: TF de la salida obtenida enviando 4V al potenciómetro de resonancia y la *sweep* estática como señal de audio de entrada.



Solo hay un valor máximo de la campana por cada voltaje enviado, dado que no estamos modificando el control de la frecuencia de corte. Como con la frecuencia de corte, no podemos calcular ni la media ni la desviación estándar porque todos los parámetros de resonancia tienen valores muy diferentes, con lo que calculando la media estaríamos distorsionando la realidad y sus valores no se corresponderían con la media real.

Cabe mencionar que enviando la *sweep* estática y controlando el parámetro de la resonancia con CV, los valores estimados se acercan a los obtenidos con la misma señal de entrada y control manual para la frecuencia de corte del 7º punto del potenciómetro correspondiente. Esto es curioso porque, controlando un parámetro por CV, esperaríamos que el otro parámetro estuviera desactivado. En este caso, cabría esperar que la frecuencia de corte fuera máxima y que el filtro no actuase. Sin embargo, vemos que sí que hay una frecuencia de corte y que, este parámetro está activado.

#### Resultados con diferentes niveles de amplitud de la señal de control.

Enviando la *sweep* variable como señal de entrada, vemos un comportamiento muy similar al obtenido con la misma señal con control manual. La pendiente es muy parecida y, de la misma manera, va disminuyendo a medida que el nivel de la señal de entrada decrece (excepto enviando 1V). Como pasaba con CV y la *sweep* estática, si enviamos 10 V (que debería corresponder a 16 kHz de la frecuencia de corte), la salida no corresponde con éste máximo, sino que es mucho más grave. Aun así, los valores de ambos estadísticos son bastante similares a los obtenidos en las [Tabla 4.5 y 4.7 del Apéndice 4](#). También la frecuencia de corte real es más grave que la esperada a medida que la magnitud se reduce. Además, también obtenemos que la SNR disminuye a medida que la señal decrece.

Aunque hay algunas diferencias: la primera tiene relación con el mapeado entre el ajuste del potenciómetro de resonancia y su valor máximo correspondiente. Igual que con la *sweep* estática y CV, éste no es el doble del valor del ajuste, sino que es menor que su posición, excepto con altos voltajes, donde éste es más grande del doble de la posición. Con un nivel diferente en la señal de entrada, no hay ningún comportamiento lineal en el VCF. Lo que sí que vemos que se repite es el efecto que tiene el ruido, que destaca conforme la señal decrece y provoca la detección de muchos puntos de corte entre la referencia y la salida. Vemos también un efecto diferente del oscilador a máxima resonancia (10V), el cual modula la frecuencia central, provocando que ésta sea, ligeramente, más grave que enviando 7V, valor con el que aún no hay oscilación automática y la resonancia está bien definida.

Por último, vemos una diferencia con respecto a los resultados con la *sweep* estática y CV. Con la señal estática, el control de la resonancia por CV correspondía al 7º punto del potenciómetro. Sin embargo, con la señal variable y CV, este control se acerca más al 6º. Es decir que con la *sweep* variable como señal de entrada, el filtro, controlado por CV, realiza el corte a la mitad de frecuencia, en los tres niveles de la *sweep*.

### Respuesta de los controles: lineal, logarítmico, exponencial.

Para finalizar con esta sección, presentamos la evolución del mapeo del potenciómetro a los valores estimados de la frecuencia de corte y la resonancia con la *sweep* estática a la entrada. Con el CV, sospechamos que puede haber cambios en el comportamiento del VCF, dado que el control es por voltaje no por su control manual. Eso podría causar que la evolución de este mapeado fuera diferente. Para comprobarlo, seguimos el mismo procedimiento que en los dos casos anteriores, utilizando el fichero *MAIN\_CV.py*.

En este caso, la diferencia más destacable es que cada nivel de magnitud máximo de la resonancia solo contiene un valor, en vez de la sublista que teníamos con el control manual, ya que hay una sola frecuencia de corte para cada resonancia. En las Figuras 4.21 y 4.22 comparamos las respuestas de las frecuencias de corte y de las resonancias en manual y CV. Observamos que, en realidad, no hay ningún cambio en el tipo de evolución que tenemos en el mapeado: sigue siendo exponencial, y el punto donde crece súbitamente es muy cercano entre los 4 casos, si bien es cierto que con el CV tenemos menos puntos ya que solo muestreamos con 4 niveles de tensión (1V, 4V, 7V, 10V) separados uniformemente, mientras que, con el control manual, tenemos las 11 marcas de la serigrafía de los potenciómetros.

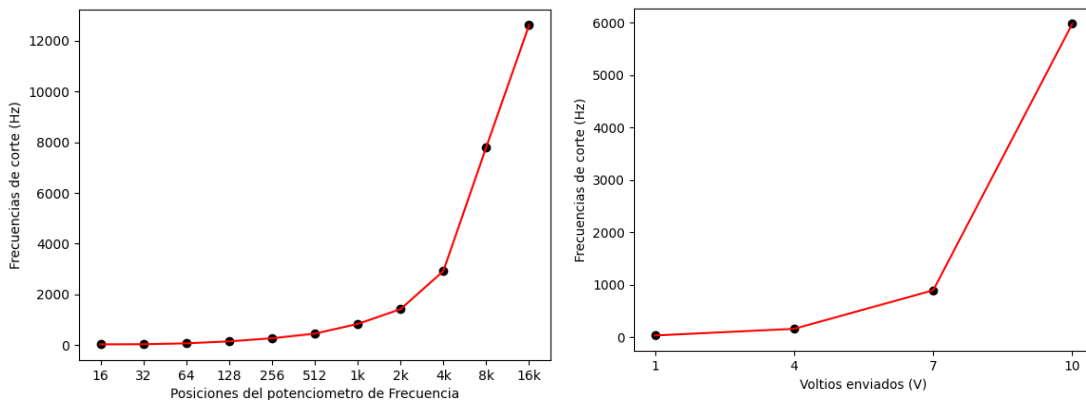


Figura 4.21: Frecuencias de corte en control manual vs Frecuencias de corte en CV

En la Figura 4.21, podemos observar las correspondencias entre cada señal de tensión y la frecuencia de corte elegida con el control manual y podemos comprobar que, con el CV, se aleja mucho de la frecuencia de corte que creemos que corresponde a cada posición. Vemos también que, en control manual, el rango de este potenciómetro llega hasta los 12 kHz, mientras que, con CV, éste (con el muestreo que hemos hecho) se queda a la mitad, un poco menos de 6 kHz. No obstante, recordemos que enviando 10V al parámetro de frecuencia de corte, no obtenemos la frecuencia de corte máxima, por lo que no podemos afirmar que el rango con CV sea de 6 kHz.

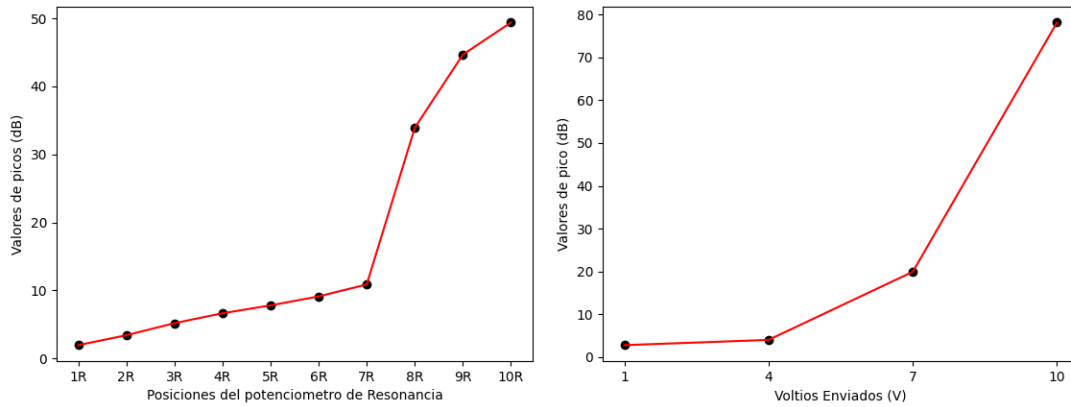


Figura 4.22: Resonancias en control manual vs Resonancias\_en CV

En la Figura 4.22, vemos que el crecimiento de los valores de resonancia es más suave. Esto se debe a que con 4V (correspondiente al 4º punto del potenciómetro) y con 7V (correspondiente al 7º) el nivel máximo de la campana resultante es diferente. Con 4V, el obtenido con CV es más bajo que el obtenido manualmente, y con 7V pasa al revés, el nivel máximo obtenido con CV es más alto que el obtenido manualmente. No obstante, en este caso, cabe recordar que los valores obtenidos manualmente, que vemos en la gráfica de la izquierda, son la media de todas las frecuencias, así que el valor no se corresponde exactamente con el obtenido con CV, que corresponde al valor de una sola frecuencia. Por último, vemos que, a máxima resonancia, (10R y 10V), el máximo de la campana es mucho mayor (casi el doble) en CV respecto al control manual. Aun así, en general, el resultado es muy similar.



## 5. Conclusiones

El objetivo principal de este trabajo era realizar un análisis exhaustivo y comparar las posiciones de los dos potenciómetros principales del filtro del VCF (frecuencia de corte y resonancia), con valores empíricos encontrados para diferentes señales de prueba de entrada y manipulando los potenciómetros (manualmente y con CV).

Después de haber grabado las salidas del filtro, haber obtenido las respuestas impulsionales de cada una de ellas y haber aplicado un proceso para obtener los resultados finales que hemos analizado, podemos decir que los valores asumidos inicialmente se corresponden con los obtenidos con el análisis, aproximadamente. Aun así, hemos visto pequeñas desviaciones, sobre todo con el control por voltaje. Estas desviaciones pueden deberse a muchos factores, tales como el hecho de que se trata de un sintetizador analógico y antiguo al que se le ha hecho poco mantenimiento en las últimas décadas, por lo que algunos componentes pueden haberse deteriorado con el tiempo.

Sin embargo, el motivo más importante es que el filtro tiene un comportamiento no lineal. Dado que no responde igual a distintos puntos del espectro, modula la deformación de la resonancia y la pendiente del filtro dependiendo de la amplitud de la señal de entrada. Al final el VCF no es un filtro estático, es interesante que articule para poder generar diferentes timbres o nuevas texturas a la síntesis de audio. Que sea no lineal puede ser porque se ha diseñado para dar esta respuesta, pero también puede ser porque trabaja con amplificadores operacionales y transistores, y éstos suelen introducir estos comportamientos dependiendo de su papel. Esto es un factor determinante a la hora de virtualizar este módulo, ya que añade mucha complejidad.

En cuanto al potenciómetro de la frecuencia de corte, su rango es menor del que marca la serigrafía de éste (pasamos del teórico de 16 Hz a 16 kHz, a un rango real de 27 Hz hasta 13 kHz). Por otro lado, la pendiente real que hemos encontrado es de, aproximadamente, 7 dB/octava menos de los que menciona el manual (pasamos de 24 dB/octava a 17 dB/octava) y, por tanto, se trata de un filtro de orden 3. También hemos visto que la evolución de los valores de frecuencia de corte a los que corresponde cada posición del potenciómetro representa un comportamiento exponencial (no lineal), lo cual significa que a bajas frecuencias (potenciómetro girado a la izquierda), el valor crece muy lentamente, mientras que a frecuencias altas (potenciómetro girado a la derecha), este crecimiento es mucho más rápido.

Referente al potenciómetro de la resonancia, hemos visto, en primer lugar, que cada posición de éste se corresponde con el doble de valor en la magnitud máxima de la campana de la resonancia (2o punto corresponde con 4 dB, 3r punto corresponde con 6 dB y así sucesivamente) excepto en el primero y en los dos o tres últimos puntos, en los que por un motivo u otro (capítulo 4), este valor es muy bajo o alto. Esto implica que a bajas y a altas resonancias no podemos caracterizar este potenciómetro, debido a que la resonancia o bien no está del todo formada o bien aparece el oscilador automático, el cual deforma la respuesta en frecuencia.

Conociendo este comportamiento, podemos hablar del rango del potenciómetro, el cual conocemos por los valores de la magnitud máxima de la campana de resonancia que corresponden a cada posición del control, hasta la 8a, en la que aparece la resonancia automática. El resultado que hemos visto es que el rango va desde los 0.5 hasta los 15 dB de magnitud máxima de resonancia.

Los demás parámetros (como el ancho de banda, la frecuencia de resonancia o el factor Q) no tienen un comportamiento tan estable como el máximo, por lo que no se han tenido en cuenta a la hora de conocer el rango o la evolución del mapeo entre los puntos del potenciómetro y los valores reales de la resonancia (que también es exponencial). Pero sí que podemos decir que dependen de la frecuencia de corte y del nivel máximo de la campana de resonancia.

Por último, podemos destacar que este comportamiento general se mantiene independientemente del nivel de la señal de entrada, ya que los valores son bastante similares, si bien hay algunos cambios. Y lo mismo sucede con el control por CV en vez de manual, el comportamiento es muy similar, si bien, como hemos dicho al principio de este apartado, los valores estimados no concuerdan con los asumidos inicialmente, sino que son más graves, lo que implica que nuestras asunciones de que el VCF recibe 1V/octava y que, como máximo, podemos enviar 10V, son incorrectas.

El análisis de un filtro controlado por tensión es un proceso complejo, con muchos detalles que hay que tener en cuenta y muchos procesos que hay que realizar al tratarse de equipamiento con un ajuste de parámetros manual. En este sentido, ha habido ciertos aspectos que no han quedado demasiado claros (como, por ejemplo, los factores que hacen que el filtro incremente el nivel de la banda pasante o como el mapeo entre los voltajes que enviamos con CV y el valor de frecuencia de corte elegido). Y otros detalles que podrían haberse mejorado o haberse explicado mejor (como las desviaciones existentes o algunos detalles de implementación o de planteamiento de la metodología o realizar las medidas a una frecuencia de muestreo con mayor resolución). También nos ha faltado analizar el comportamiento temporal ante una señal conocida para conocer su envolvente o como se genera internamente cuando hay un cambio súbito de niveles o un transitorio. Quizás con pulsos de diferente amplitud hubiéramos visto la velocidad de la envolvente interna incluyendo el *attack* y el *decay*.

Asimismo, ha habido dos valores que se desvían respecto a los valores teóricos dados en el manual (el rango del potenciómetro de frecuencias y la pendiente del VCF). Recordemos que estos dos parámetros son los únicos que aparecen específicamente en el manual, los demás (como las posiciones de los potenciómetros de frecuencia de corte y resonancia) eran parámetros asumidos en un principio. Es posible que el ruido de fondo esté afectando alguna octava y haya falseado estos resultados. Como trabajo futuro, se podría realizar el cálculo de la pendiente en más octavas y ver la evolución de éste en lugar de directamente hacer la media del pendiente en 2 octavas. Por otro lado, en cuanto a futuros proyectos, el próximo paso podría ser la virtualización de este VCF, de manera lo más real posible, que implemente el comportamiento que hemos expuesto en el análisis. Para ello, hay diversas técnicas, como el modelado a partir de respuestas impulsionales, a través de Wave Digital Filters (WDF)<sup>16</sup>, o usando algoritmos de *deep learning*.

---

<sup>16</sup> Modo de simulación de circuitos eléctricos para el procesamiento de señales.

## Bibliografía

- [1] "Sintetizador", 2022. [online]. Disponible en: <https://es.wikipedia.org/wiki/Sintetizador> (acceso: 8 de Febrero de 2022)
- [2] "Proyecto Sintetizador Documentación Versión 1.0", 2009. [online]. Disponible en: <http://profesores.elo.utfsm.cl/~agv/elo329/1s09/projects/AcevedoBarros/SinteAnalogo.html> (acceso: 18 de Enero de 2022)
- [3] C. Meyer, "Learning Modular", 2016. [online]. Disponible en: <https://learningmodular.com/glossary/control-voltage/> (acceso: 7 de Febrero de 2022)
- [4] "RSF Kobol", 2021. [online] Disponible en: [https://en.wikipedia.org/wiki/RSF\\_Kobol](https://en.wikipedia.org/wiki/RSF_Kobol) (acceso: 11 de Setiembre de 2021)
- [5] J. Michel Jarre, "RSF Kobol Expander", 2018. [online]. Disponible en: [http://www.jarrography.free.fr/details\\_equipement\\_audio.php?id\\_equip=80](http://www.jarrography.free.fr/details_equipement_audio.php?id_equip=80) (acceso: 8 de Febrero de 2022)
- [6] "Reverb", 2018. [online]. Disponible en: <https://reverb.com/item/18757272-rsf-kobol-expander-i> (acceso: 8 de Febrero de 2022)
- [7] "RSF Kobol Expander", 2018. [online]. Disponible en: [https://www.vintagesynth.com/misc/rsf\\_kobol.php](https://www.vintagesynth.com/misc/rsf_kobol.php) (acceso: 23 de Diciembre de 2021)
- [8] G. Reid,, "XILS Lab PolyKB", 2010. [online]. Disponible en: <https://www.soundonsound.com/reviews/xils-lab-polykb> (acceso: 19 de Enero de 2022)
- [9] "RSF Kobol Manual" , 2017. [online]. Disponible en: [http://yusynth.net/gear/Kobol\\_RSF\\_en.html](http://yusynth.net/gear/Kobol_RSF_en.html) (acceso: 10 de Enero de 2022)
- [10] S. Howell, "RSF Kobol Expander" , 1983. [online]. Disponible en: <http://www.muzines.co.uk/articles/rsf-kobol-expander/5929> (acceso: 6 de Febrero de 2022)
- [11] "Edirol\_UA-101\_OM", 2005. [online]. Disponible en: [https://www.roland.com/global/support/by\\_product/ua-101/owners\\_manuals/39364883-4342-471f-9730-0bf7648914aa/](https://www.roland.com/global/support/by_product/ua-101/owners_manuals/39364883-4342-471f-9730-0bf7648914aa/) (acceso: 16 de Febrero de 2022)
- [12] A. Farina, "Aurora for Audacity", 2021. [online]. Disponible en: <http://www.angelifarina.it/Public/Aurora-for-Audacity/> (acceso: 11 de Setiembre de 2021)
- [13] "Let's clear up some things about sweeps". [online]. Disponible en: <https://www.nti-audio.com/en/support/know-how/lets-clear-up-some-things-about-sweeps> (acceso: 7 de Junio de 2022)
- [14] Armelloni, E., Bellini, A., & Farina, A. (2001, Mayo). Not-linear convolution: A new approach for the auralization of distorting systems. In Audio Engineering Society Convention 110. Audio Engineering Society [\[PDF\]](#)

- [15] Universidad Miguel Hernandez, "Circuitos electrónicos analógicos", [online]. Disponible en: <https://sites.google.com/a/goumh.umh.es/circuitos-electronicos-analogicos/transparencias/tema-2> (acceso: 7 de Marzo de 2022)
- [16] "Transfer Functions", 2022. [online]. Disponible en: [https://en.wikipedia.org/wiki/Transfer\\_function](https://en.wikipedia.org/wiki/Transfer_function) (acceso: 6 de Abril de 2022)
- [17] "Pathlib", 2022. [online]. Disponible en: <https://docs.python.org/3/library/pathlib.html> (acceso: 28 de Abril de 2022)
- [18] "Distorsión Armónica Total", 2017. [online]. Disponible en: <https://www.seencogroup.com/distorsion-armonica-total-que-paso-con-mi-onda-perfecta/> (acceso: 22 de Marzo de 2022)
- [19] "Distorsión Armónica", 2021. [online]. Disponible en: [https://es.wikipedia.org/wiki/Distorsi%C3%B3n\\_arm%C3%B3nica](https://es.wikipedia.org/wiki/Distorsi%C3%B3n_arm%C3%B3nica) (acceso: 22 de Marzo de 2022)
- [20] X. Lizarraga, "analog digital level calibration", 2022. [online]. Disponible en: <https://github.com/xaviliz/analog-digital-level-calibration> (acceso: 7 de Febrero de 2022)
- [21] "MTG: Essentia", 2006, [online]. Disponible en: <https://github.com/MTG/essentia> (acceso: 28 de Abril de 2020)
- [22] "Rango dinámico", 2017. [online]. Disponible en: <https://geekland.eu/que-es-el-rango-dinamico-audio-ventajas/> (acceso: 28 de Marzo de 2022)



## APÉNDICES

### 1. Figuras



Figura 2.1: RSF Kobol



Figura 2.2: RSF Kobol Expander I



Figura 2.3: VCF



Figura 3.1: Configuración inicial



Figura 3.2: Loopback



Figura 3.3: Montaje de grabación

$$H(s) = \frac{V_o(s)}{V_i(s)}$$

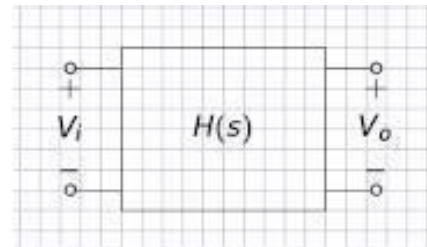


Figura 3.4: Función de transferencia de un filtro

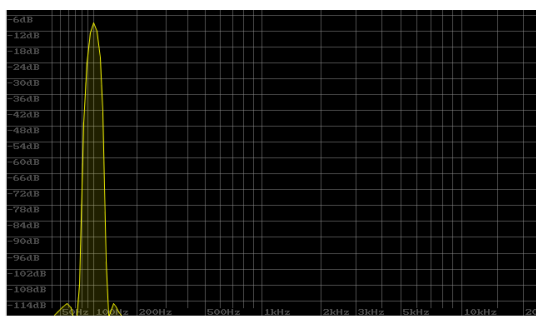


Figura 4.1: tone100 (input)

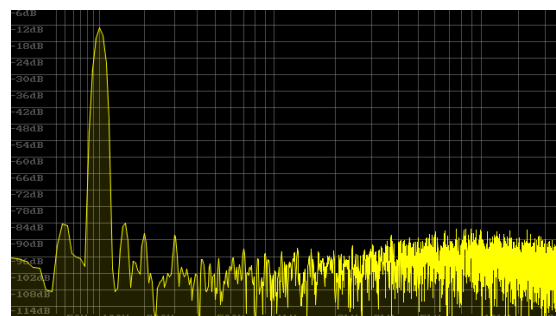


Figura 4.2: Bypass\_tone100 (output)



Figura 4.3: tone440 (input)

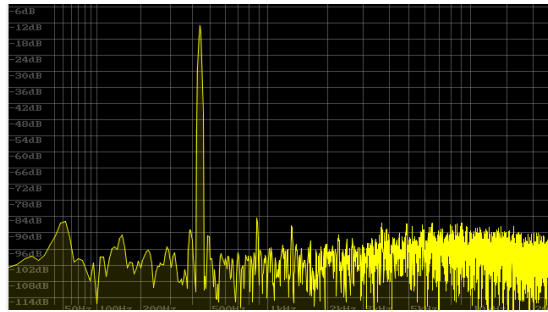


Figura 4.4: Bypass\_tone440 (output)

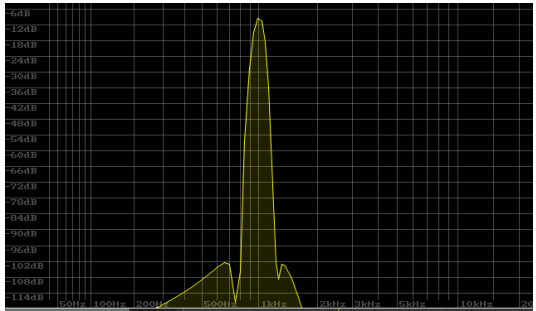


Figura 4.5: tone1k (input)

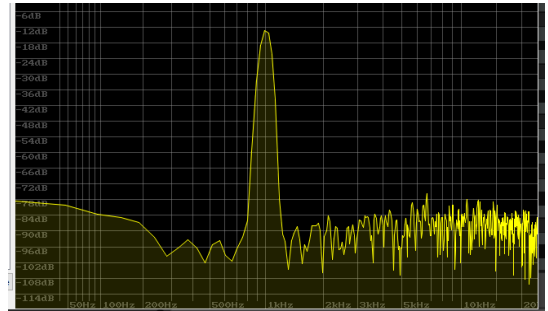


Figura 4.6: Bypass\_tone1k (output)



Figura 4.7: tone10k (input)

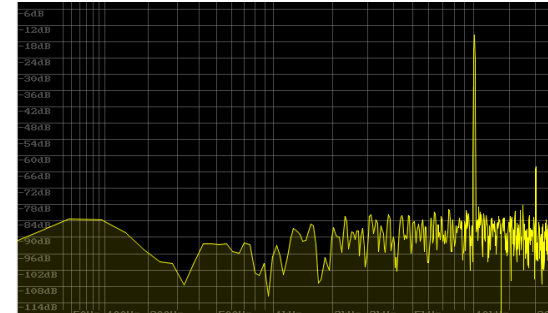


Figura 4.8: Bypass\_tone\_10k (output)

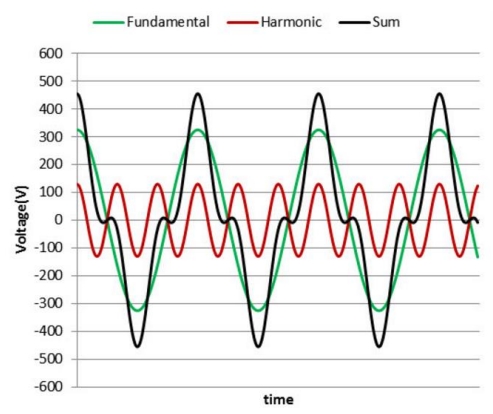


Figura 4.9: THD: Efecto de la distorsión armónica en una onda sinusoidal (línea verde: señal, línea roja: armónico, línea negra: suma)

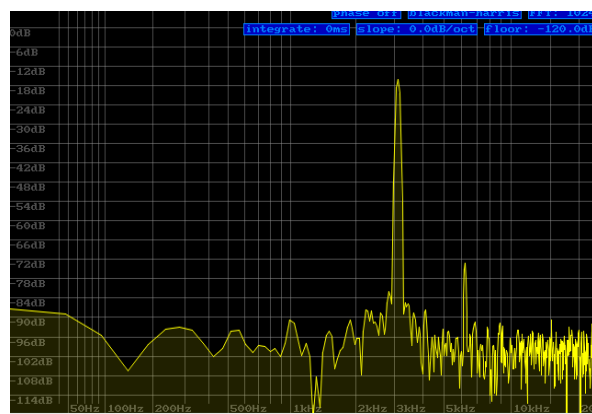
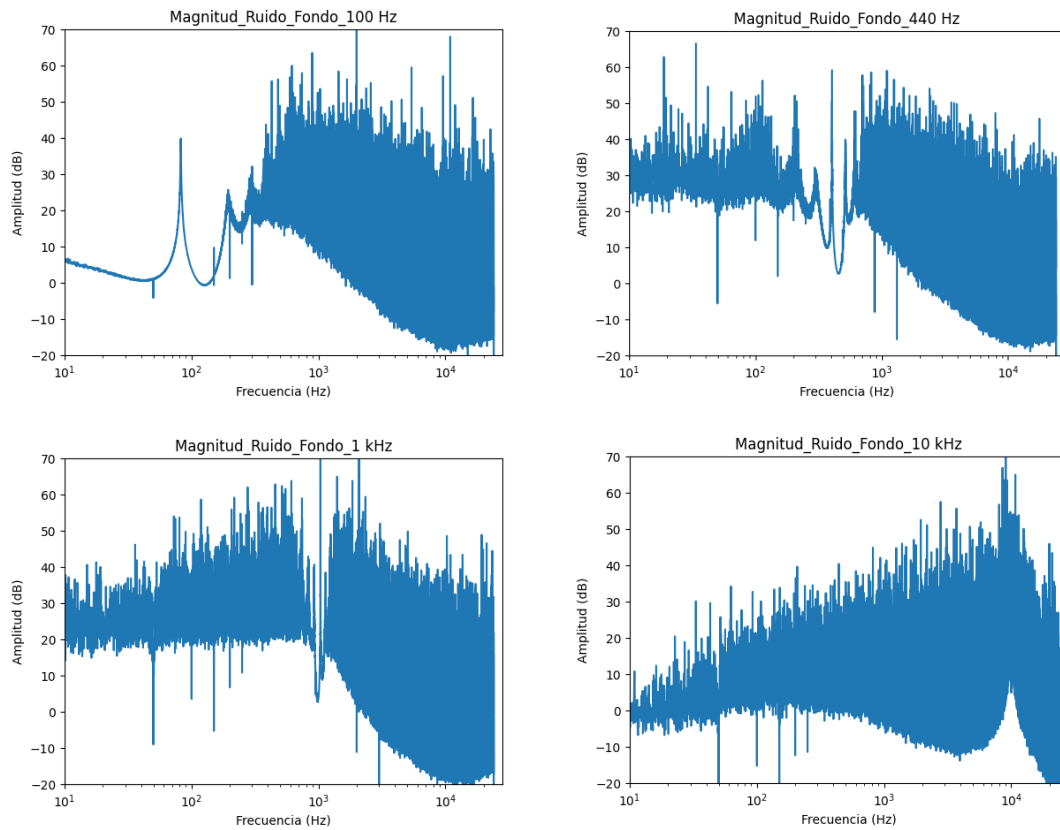
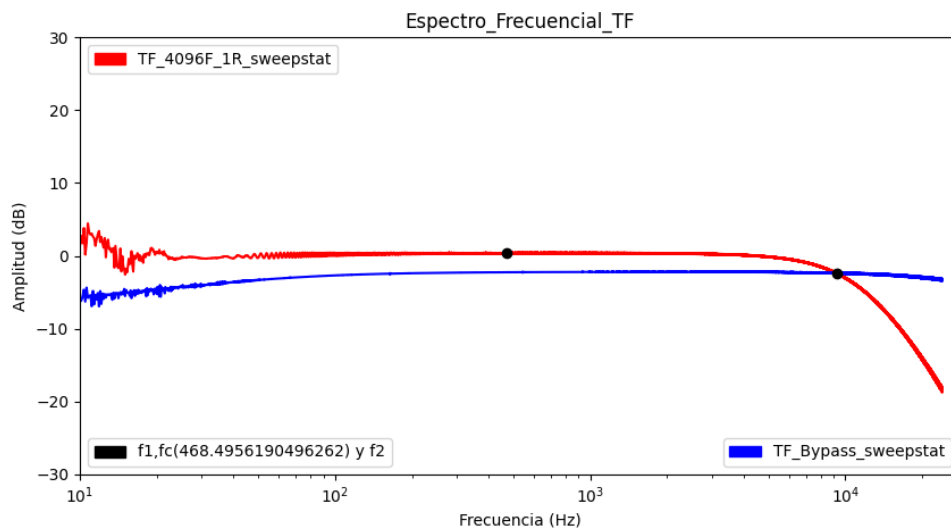


Figura 4.10: VCF 4096F allRes sweepstat: Espectro frecuencial de la salida obtenida con el potenciómetro de frecuencia en la antepenúltima posición y todas las posiciones del potenciómetro de resonancia.



**Figura 4.11: FFT Noise Tones:** Magnitud del espectro frecuencial del ruido de fondo según el tono de la señal de entrada.



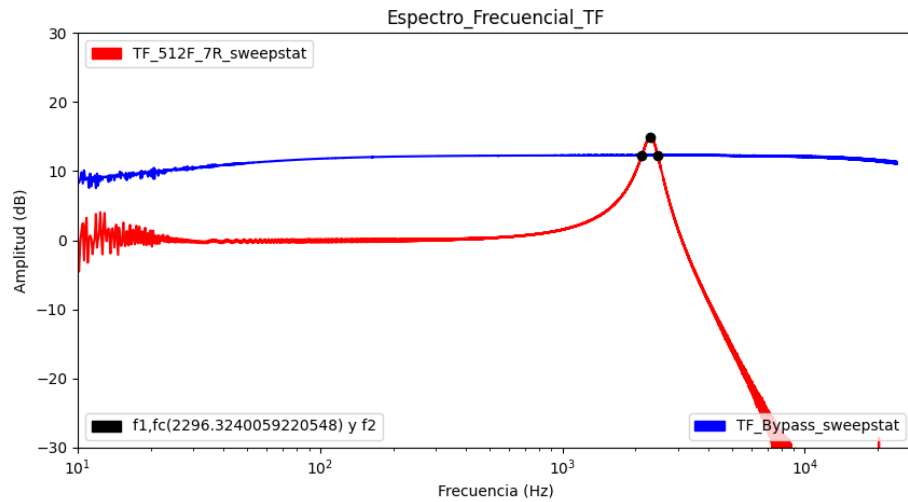
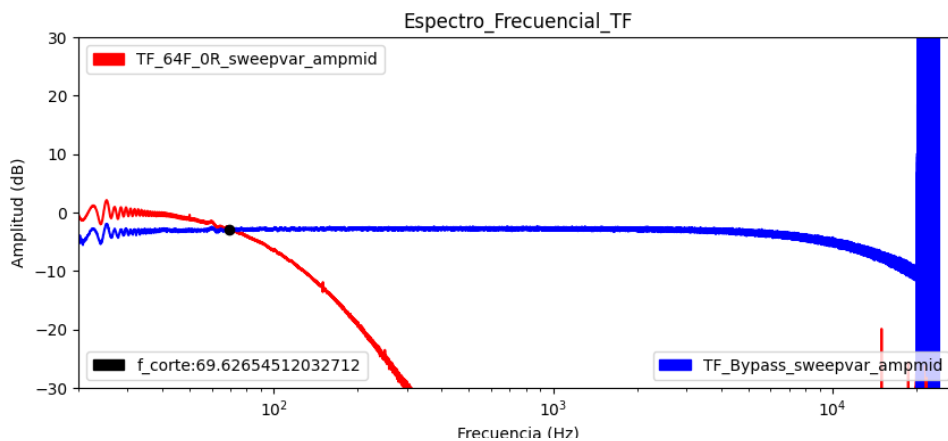
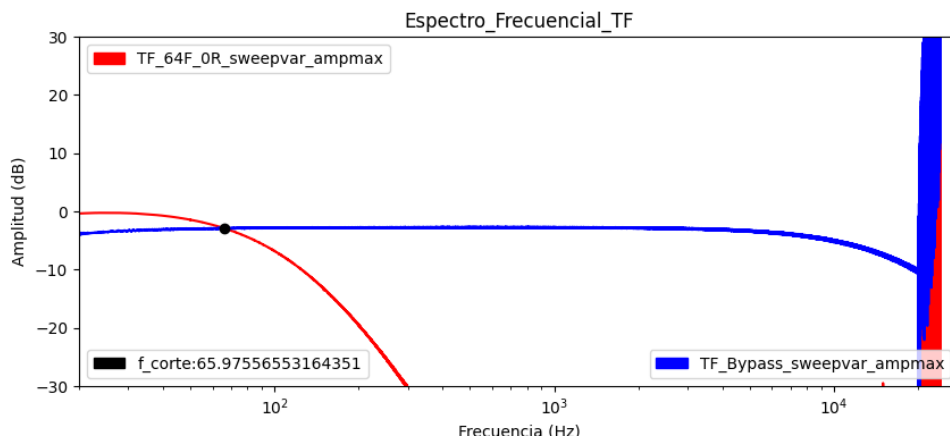


Figura 4.15: Rango Resonancia sweepstat: Vemos 2 TF de salidas obtenidas con dos posiciones del potenciómetro de frecuencia (9ª y 6ª, respectivamente) y dos posiciones del de resonancia (la mínima y la máxima previa a la oscilación, respectivamente)



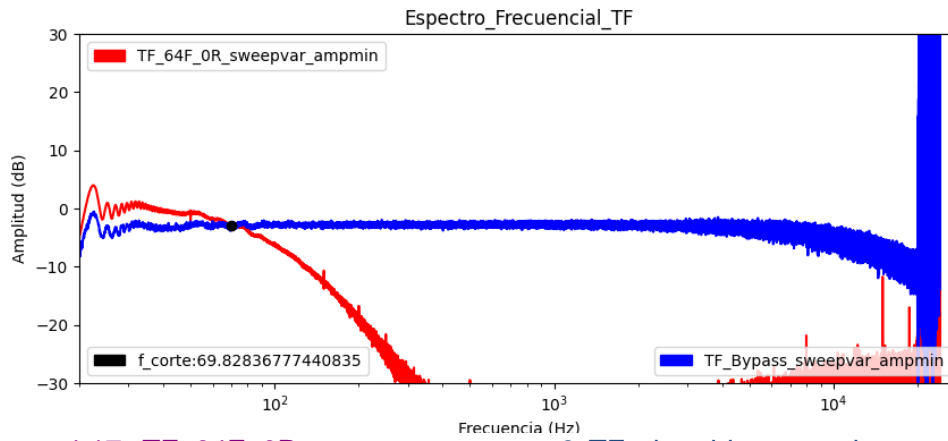
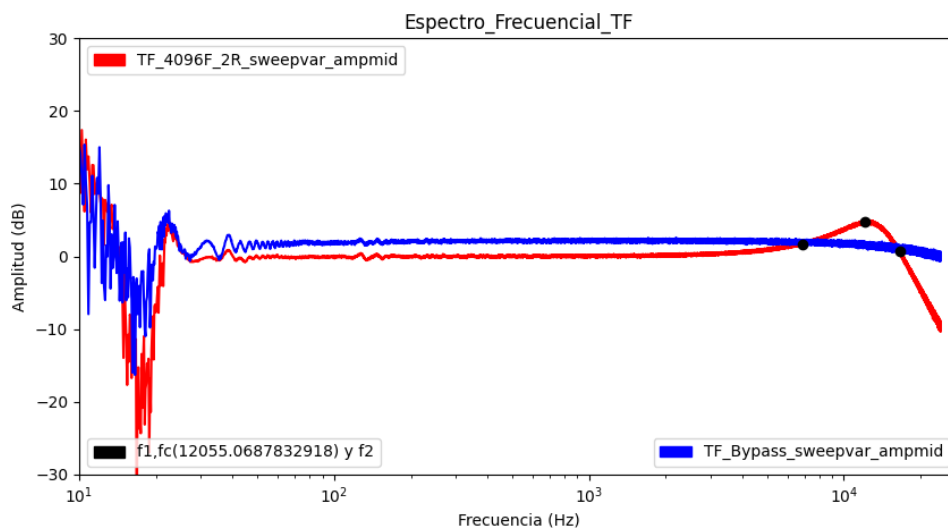
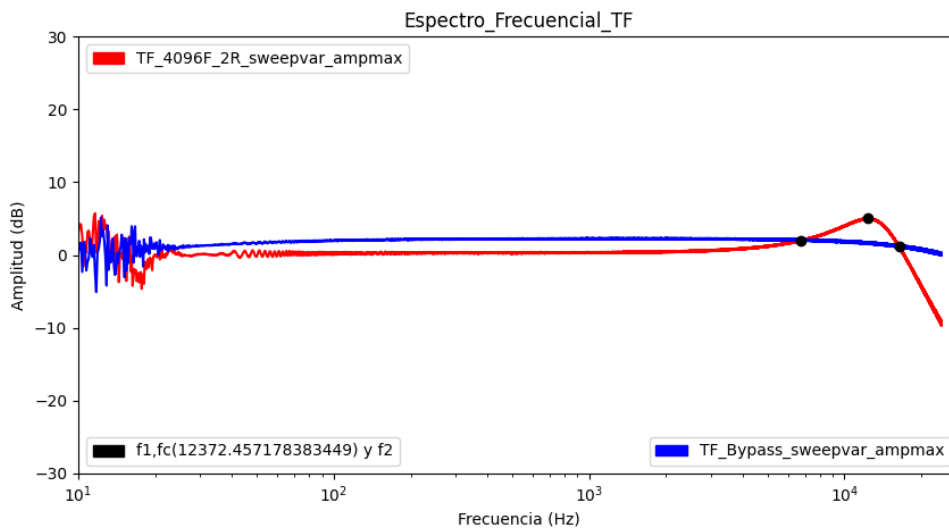
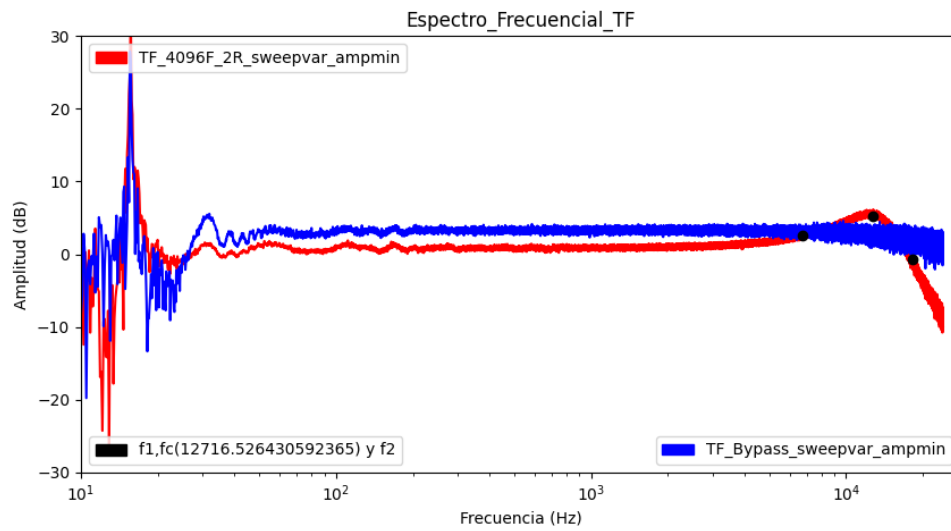


Figura 4.17: TF 64F 0R sweepvar: vemos 3 TF obtenidas con el potenciómetro de frecuencia en la 3ª posición, el de resonancia al mínimo y la sweep variable como señal de entrada.





[Figura 4.18: TF 512F 5R sweepvar](#): vemos 3 TF obtenidas con el potenciómetro de frecuencia en la 9ª posición, el de resonancia en la 3ª y la sweep variable como señal de entrada.



## 2. Nomenclatura

### Señales de prueba

<i>sweep_static</i>	<i>tone_100</i>	<i>tone_1k</i>
<i>sweep_variable</i>	<i>tone_440</i>	<i>tone_10k</i>

### Salidas del filtro

- *VCF\_16F\_0Res\_sweepvar* : salida con la frecuencia de corte a 16 Hz y la resonancia a 0, habiendo enviado la *sweep* variable como señal de entrada
- *Bypass\_sweepstat* : salida en *bypass* enviando la *sweep* estática como señal de entrada

### Respuestas Impulsionales (IR)

- *IR\_16F\_2R\_sweepstat* : IR de la salida obtenida con el potenciómetro de la frecuencia de corte a 16 Hz y el de la resonancia a 2, y habiendo enviado la *sweep* estática como señal de entrada.
- *IR\_CV\_res\_4V* : IR de la salida obtenida enviando 4V al potenciómetro de resonancia a través del CV.

### Funciones de transferencia (TF)

- *TF\_512F\_5R\_sweepvar\_ampmid* : función de transferencia de la IR obtenida a partir de la salida generada con el potenciómetro de la frecuencia de corte a 512 Hz y el de la resonancia en el punto 5 y con la *sweep* variable como señal de entrada.
- *TF\_32F\_6R\_sweepstat* : función de transferencia de la IR obtenida a partir de la salida generada con el potenciómetro de la frecuencia de corte a 32 Hz y el de resonancia en el punto 6 y con la *sweep* estática como señal de entrada.

### Resultados de los códigos

- *Results\_512F\_5R\_sweepvar\_ampmid* : valores de los parámetros de resonancia (*f1*, *f2*, *fres*, *fcentral*, *gain*, *Q*) de la función de transferencia de la IR obtenida a partir de la salida generada con el potenciómetro de la frecuencia de corte a 512 Hz y el de la resonancia en el punto 5 y con la *sweep* variable como señal de entrada.
- *Results\_32F\_0R\_sweepstat* : valores de frecuencia de corte y pendiente de la función de transferencia de la IR obtenida a partir de la salida generada con el potenciómetro de la frecuencia de corte a 32 Hz y el de resonancia en el punto 0 y con la *sweep* estática como señal de entrada.



### 3. Ecuaciones y Fórmulas

$$TF = \frac{IR \text{ output } fft}{IR \text{ input } fft} \quad (\text{Ecuación 3.1})$$

$$TF_{mag}(dB) = 20 * \log_{10}(TF_{mag}) \quad (\text{Ecuación 3.2})$$

$$THD_{tone10k} = \frac{\sum \text{ nivel armónicos}}{\text{ nivel frecuencia fundamental}} * 100 = \frac{-64 \text{ dBFS}}{-15 \text{ dBFS}} \quad (\text{Ecuación 4.1})$$

$$dBu = dBFS + 10 \quad (\text{Ecuación 4.2})$$

$$V = 10^{\frac{dBu}{20}} * 0.0775 \quad (\text{Ecuación 4.3})$$

$$SNR = \frac{\text{ nivel señal}}{\text{ nivel ruido}} \quad (\text{Ecuación 4.4})$$

$$SNR = 10 * \log_{10} \frac{\text{ nivel señal}}{\text{ nivel ruido}} \quad (\text{Ecuación 4.5})$$

$$SNR = 20 * \log_{10} \frac{\text{ nivel señal}}{\text{ nivel ruido}} \quad (\text{Ecuación 4.6})$$

$$stdev = \sqrt{|x_i - mean|} \quad (\text{Ecuación 4.7})$$

$$Q = \frac{f_{res}}{f_2 - f_1} \quad (\text{Ecuación 4.8})$$

$$f_{res} = \sqrt{f_1 * f_2} \quad (\text{Ecuación 4.9})$$

## 4. Tablas

Ajuste	Frecuencia de corte (Hz)	Pendiente (dB/octava)	Stdev (de la pendiente)
16F_0R	27.01	25.16	2.88
32F_0R	31.50	20.98	2.03
64F_0R	65.71	16.83	0.1
128F_0R	144.52	14.43	1.55
256F_0R	266.57	15.67	1.08
512F_0R	449.67	14.56	1.51
1024F_0R	830.47	15.46	1.17
2048F_0R	1412.62	15.35	1.22
4096F_0R	2918.07	15.14	1.30
8192F_0R	7802.20	14.84	1.41
<b>Media</b>		16.84	1.42

Tabla 4.3: Frecuencia sweepstat: Valores de los parámetros de la frecuencia de corte de las TF obtenidas con la sweep estática como señal de entrada y control manual.

Ajuste	f1	f2	f. central	f. res	Gain	Factor Q	stdev
16F:							
1R	15.68	22.51	0.38	18.71	2.16	2.75	0.47
2R	43.62	46.15	0.25	44.76	2.82	17.74	0.76
3R	57.03	57.41	41.73	57.16	2.59	150.83	1.61
4R	5.94	58.55	41.73	18.59	3.80	0.35	1.68
5R	0.38	59.43	45.27	4.68	4.69	0.08	1.76
6R	0.00	59.81	45.77	0.00	5.70	0.00	1.85
7R	0.00	60.19	48.43	0.00	7.02	0.00	1.96
8R	40.21	61.83	51.46	49.82	9.38	2.31	4.95
9R	54.25	61.96	61.58	57.91	18.58	7.52	5.11
10R	61.45	61.71	61.71	61.45	18.68	243.50	5.54
32F:							
1R	32.37	36.29	0.76	34.27	2.31	8.74	0.61
2R	81.56	83.08	0.76	82.19	2.54	54.24	0.93
3R	0.00	91.68	69.42	0.00	4.20	0.00	0.98
4R	42.23	94.20	72.96	62.97	5.47	1.21	1.08
5R	52.35	94.58	78.01	70.31	6.55	1.67	1.11
6R	59.94	95.72	79.54	75.74	7.69	2.12	1.19
7R	65.25	96.86	82.70	79.41	9.15	2.51	1.31
8R	75.87	98.25	87.25	86.24	11.07	3.86	4.78
9R	97.95	99.39	99.39	98.50	19.43	59.96	5.02
10R	99.39	99.39	99.39	99.39	35.60	inf	3.71

64F:							
1R	92.18	100.78	0.76	96.35	1.63	11.21	0.56
2R	52.86	268.07	203.33	118.99	2.97	0.55	0.65
3R	158.31	281.73	230.14	211.17	5.61	1.71	0.66
4R	190.31	286.53	244.30	233.43	7.06	2.43	0.65
5R	205.35	289.06	251.89	243.54	8.24	2.91	0.67
6R	224.07	292.73	259.98	256.07	9.98	3.73	0.93
7R	241.26	295.51	269.46	266.93	12.09	4.92	1.11
8R	299.68	299.94	299.81	299.69	20.05	1185.50	3.72
9R	299.68	299.94	299.94	299.68	36.65	1185.50	2.83
10R	307.78	307.90	307.90	307.78	37.90	2434.50	3.39
128F:							
1R	232.54	251.89	3.79	241.90	1.62	12.51	0.56
2R	248.85	714.06	497.96	421.46	3.81	0.91	0.64
3R	472.79	754.65	639.71	597.22	6.17	2.12	1.00
4R	532.86	765.90	662.85	638.82	7.67	2.74	1.74
5R	585.21	774.38	685.48	673.09	9.43	3.56	1.28
6R	620.99	780.32	707.36	696.10	10.90	4.37	1.34
7R	668.16	787.78	727.84	725.44	13.64	6.06	1.67
8R	744.03	796.38	796.25	769.70	19.93	14.70	3.74
9R	796.00	796.13	796.13	796.00	40.34	6295.50	2.08
10R	819.65	819.77	819.77	819.65	56.37	6482.50	2.64
256F:							
1R	353.80	377.45	2.78	365.31	2.33	15.45	0.62
2R	426.89	1233.26	909.04	725.57	3.81	0.90	0.64
3R	829.00	1304.45	1110.35	1039.79	6.01	2.19	0.92
4R	935.09	1324.69	1150.19	1112.88	7.85	2.86	1.10
5R	1010.08	1337.84	1193.94	1162.45	9.20	3.55	1.19
6R	1085.06	1350.48	1224.03	1210.50	11.06	4.56	1.40
7R	1153.98	1359.21	1259.44	1252.36	13.40	6.10	1.59
8R	1371.98	1372.99	1372.61	1372.36	52.36	1356.75	4.29
9R	1372.74	1372.86	1372.86	1372.74	72.56	10856.50	5.28
10R	1417.25	1417.37	1417.37	1417.25	76.71	11208.50	5.23
512F:							
1R	659.81	668.66	2.28	664.11	2.29	75.04	0.59
2R	892.35	2252.70	1691.77	1417.75	4.13	1.04	0.85
3R	1499.18	2365.24	1956.68	1882.96	7.08	2.17	1.38
4R	1701.25	2404.44	2092.23	2022.43	8.67	2.88	1.43
5R	1842.88	2430.23	2166.84	2116.26	10.28	3.60	1.58
6R	1981.72	2452.11	2234.11	2204.39	12.17	4.69	1.75
7R	2127.13	2473.27	2296.32	2293.54	14.94	6.63	2.02
8R	2504.08	2504.97	2504.59	2504.46	55.78	2829.50	4.68
9R	2504.08	2504.59	2504.59	2504.21	72.36	4951.25	5.26
10R	2572.36	2572.62	2572.49	2572.36	77.80	10172.00	5.33
1024F:							
1R	3.41	399.71	0.76	36.92	3.13	0.09	1.09
2R	1584.79	3951.29	2906.95	2502.31	4.16	1.06	0.87
3R	2607.77	4149.69	3403.52	3289.58	7.04	2.13	1.37
4R	2976.75	4214.31	3642.25	3541.85	8.75	2.86	1.46
5R	3254.68	4259.20	3777.55	3723.18	10.54	3.71	1.66

6R	3489.25	4303.34	3899.20	3875.30	12.46	4.75	1.83
7R	3711.93	4331.28	4034.37	4009.59	14.90	6.47	2.01
8R	4390.20	4390.96	4390.71	4390.45	61.99	5787.00	5.30
9R	4390.20	4391.09	4391.09	4390.58	73.07	4960.36	5.33
10R	4505.27	4506.28	4506.16	4505.65	77.78	4454.12	5.33
2048F:							
1R	1.64	36.54	1.14	7.71	4.18	0.22	1.50
2R	2555.17	6761.26	5071.76	4156.40	4.00	0.99	0.77
3R	4515.64	7115.06	5913.29	5668.23	7.13	2.18	1.40
4R	5135.24	7243.54	6190.21	6098.91	8.88	2.89	1.50
5R	5573.64	7317.00	6369.26	6386.08	10.52	3.66	1.65
6R	5951.09	7372.13	6675.40	6623.55	12.31	4.66	1.79
7R	6266.71	7419.68	6837.51	6818.79	14.29	5.91	1.85
8R	7518.44	7519.20	7518.94	7518.69	64.64	9910.17	5.54
9R	7519.95	7520.33	7520.21	7520.08	74.86	19823.83	5.49
10R	7713.30	7714.56	7714.56	7713.80	78.50	6100.40	5.39
4096F:							
1R	3.41	9287.85	468.49	178.04	0.47	0.02	1.21
2R	5212.76	14521.72	10886.04	8700.36	3.96	0.93	0.75
3R	9193.89	15214.66	12435.55	11827.08	6.88	1.96	1.31
4R	10694.09	15453.53	13204.87	12855.37	8.85	2.70	1.49
5R	11520.82	15566.95	13613.31	13391.89	10.26	3.31	1.57
6R	12405.84	15657.49	13960.54	13937.14	12.18	4.29	1.75
7R	13234.34	15779.13	14422.58	14450.78	14.55	5.68	1.92
8R	15848.55	15849.31	15849.19	15848.89	72.51	20889.67	6.21
9R	15848.43	15848.68	15848.55	15848.43	76.67	62667.50	5.65
10R	16273.80	16277.72	16277.47	16275.70	77.01	4152.05	5.25
8192F:							
1R	4.68	10.24	1048.90	6.83	0.52	1.24	1.19
2R	11.76	11137.93	23865.71	361.90	3.53	0.03	0.36
3R	10.37	14172.72	23726.62	383.27	4.10	0.03	1.03
4R	7.59	14650.95	23864.95	333.32	4.15	0.03	1.57
5R	10.49	15834.39	23726.49	407.55	4.33	0.03	1.86
6R	6.20	16026.98	23865.08	315.11	4.25	0.02	2.20
7R	10.49	14993.12	23674.90	396.67	4.06	0.03	2.61
8R	14.54	15472.37	23702.08	474.31	3.96	0.03	5.47
9R	16.19	23623.43	23731.80	618.34	4.50	0.03	6.34
10R	20.61	23880.50	23763.54	701.54	4.69	0.03	6.68
16384F:							
1R	2.28	13.53	1522.71	5.44	0.66	0.49	1.13
2R	4.42	17.20	23379.76	8.60	1.72	0.68	1.30
3R	9.61	18.08	23483.58	13.15	1.69	1.55	1.86
4R	1.90	18.59	23602.44	5.82	1.75	0.36	2.21
5R	13.78	18.84	23645.56	16.06	1.66	3.19	2.48
6R	14.03	17.32	23645.18	15.55	1.56	4.74	2.75
7R	9.23	17.32	23161.64	12.64	1.48	1.56	3.06
8R	8.72	15.81	23863.43	11.63	1.41	1.66	5.70
9R	10.87	21101.27	23149.50	478.99	2.41	0.02	6.50
10R	10.11	4766.01	23838.90	219.52	2.28	0.05	6.86

Pos	1R	2R	3R	4R	5R	6R	7R	8R	9R	10R
<b>Media (dB)</b>	1.94	3.40	5.17	6.63	7.79	9.11	10.86	33.92	44.67	49.39

Tabla 4.4: Resonancia\_sweepstat: Valores de los parámetros de la resonancia de las TF obtenidas con la *sweep* estática como señal de entrada y control manual.

Ajuste	Frecuencia de corte (Hz)	Pendiente (dB/octava)	Stdev (de la pendiente)
16F_0R: Ampmax Ampmid Ampmin	26.87 28.17 32.57	24.51 25.69 23.91	2.63 2.76 2.62
64F_0R Ampmax Ampmid Ampmin	65.97 69.63 69.83	15.93 16.75 16.82	1.29 1.15 0.45
512F_0R Ampmax Ampmid Ampmin	459.44 439.69 446.44	14.58 13.95 13.38	1.73 2.03 1.91
4096F_0R Ampmax Ampmid Ampmin	3350.71 3331.20 2934.55	15.34 15.93 13.95	1.50 1.47 1.75
<b>Media</b> Ampmax Ampmid Ampmin		17.59 18.08 17.02	1.79 1.85 1.68

Tabla 4.5: Frecuencia\_sweepvar: Valores de los parámetros de la frecuencia de corte de las TF obtenidas con la *sweep* variable como señal de entrada y control manual.

Ajuste	f1	f2	f. central	f. res	Gain	Factor Q	Stdev
16F: 2R							
Ampmax	0.25	52.22	29.21	3.54	1.05	0.07	1.53
Ampmid	4.55	23.01	9.36	10.11	5.55	0.55	0.87
Ampmin	26.30	34.39	6.07	29.97	5.54	3.72	0.68
5R							
Ampmax	19.98	58.04	44.26	34.01	4.62	0.89	2.03
Ampmid	4.08	53.11	9.36	15.93	4.93	0.33	2.04
Ampmin	34.90	52.73	6.20	42.87	3.43	2.41	2.37
8R							
Ampmax	40.08	60.44	51.84	49.19	9.32	2.42	2.73
Ampmid	40.19	53.11	51.32	46.20	9.54	3.57	2.84
Ampmin	42.23	58.29	50.31	48.57	10.33	3.09	2.84

10R							
Ampmax	60.57	60.82	60.82	60.57	22.03	240.00	5.56
Ampmid	60.70	60.82	60.82	60.70	70.26	480.50	3.29
Ampmin	60.57	60.82	60.82	60.57	53.49	240.00	5.17
64F:							
2R							
Ampmax	0.25	288.30	215.34	8.47	3.62	0.03	0.47
Ampmid	73.85	288.43	179.30	145.92	4.21	0.68	0.76
Ampmin	7.71	290.07	205.48	47.29	3.66	0.17	1.19
5R							
Ampmax	217.87	312.08	270.22	260.74	9.32	2.77	0.76
Ampmid	221.79	311.95	257.96	263.01	9.99	2.92	0.94
Ampmin	225.96	312.96	253.66	265.92	9.42	3.06	0.59
8R							
Ampmax	283.37	320.55	301.45	301.33	17.65	8.11	0.92
Ampmid	285.40	320.93	303.35	302.59	18.40	8.52	0.90
Ampmin	286.28	322.19	304.23	303.61	17.79	8.46	0.77
10R							
Ampmax	330.03	330.16	330.16	330.03	33.80	2610.50	4.38
Ampmid	330.03	330.16	330.16	330.03	74.58	2610.50	2.55
Ampmin	330.03	330.16	330.16	330.03	73.20	2610.50	2.65
512F:							
2R							
Ampmax	715.07	2293.29	1736.78	1280.55	3.88	0.81	0.69
Ampmid	4.17	2295.82	1596.30	97.87	4.44	0.04	0.59
Ampmin	3.16	2325.15	1686.96	85.73	4.98	0.04	0.32
5R							
Ampmax	1862.22	2482.46	2200.47	2150.02	10.46	3.47	1.31
Ampmid	1890.04	2489.41	2075.67	2169.12	11.04	3.62	1.39
Ampmin	1904.08	2488.53	2137.12	2176.70	11.67	3.72	1.61
8R							
Ampmax	2338.81	2538.73	2425.55	2436.68	20.86	12.19	2.01
Ampmid	2364.10	2543.15	2449.58	2451.98	22.02	13.69	2.10
Ampmin	2380.54	2544.42	2413.04	2461.09	22.66	15.02	2.07
10R							
Ampmax	1558.87	1559.37	1559.25	1558.99	72.57	3082.50	4.42
Ampmid	1558.74	1558.87	1558.87	1558.74	87.71	12327.50	2.58
Ampmin	1366.16	1366.29	1366.29	1366.16	96.16	10804.50	3.99
4096F:							
2R							
Ampmax	6776.05	16448.94	12372.46	10557.40	5.05	1.09	1.28
Ampmid	6887.83	16525.44	12055.07	10668.80	4.97	1.11	0.42
Ampmin	6741.15	18232.51	12716.53	11086.34	6.14	0.96	1.03
5R							
Ampmax	11984.76	17082.58	14622.25	14308.40	10.56	2.81	1.35
Ampmid	12058.36	17124.56	14429.66	14369.85	10.43	2.84	1.15
Ampmin	12082.00	17407.05	14803.32	14502.12	11.78	2.72	1.65
8R							
Ampmax	15246.02	17178.42	16063.39	16183.39	19.39	8.37	1.61
Ampmid	15380.82	17132.65	16080.46	16233.09	20.42	9.27	1.68
Ampmin	15465.54	17192.97	16130.54	16306.30	22.79	9.44	2.10

10R							
Ampmax	17492.02	17494.93	17429.05	17493.41	83.57	6014.93	5.53
Ampmid	17486.96	17489.87	17373.54	17488.35	91.75	6013.19	3.27
Ampmin	17483.42	17486.08	17485.44	17484.69	98.08	6584.50	4.22

Pos	2R	5R	8R	10R
<b>Media</b>				
Ampmax	3.40	8.74	16.80	52.99
Ampmid	4.79	9.10	17.59	81.07
Ampmin	5.08	9.07	18.39	80.23

Tabla 4.6: Resonancia\_sweepvar: Valores de los parámetros de la resonancia de las TF obtenidas con la *sweep* variable como señal de entrada y control manual.

Ajuste	Frecuencia de corte (Hz)	Pendiente (dB/octava)	stdev
freq_1V (32 Hz)	36.30	20.07	2.50
freq_4V (256 Hz)	162.35	15.46	1.28
freq_7V (2048 Hz)	897.10	14.54	0.84
freq_10V (16 kHz)	5980.54	5.25	2.93
<b>Media</b>		13.83	1.89

Tabla 4.7: Frecuencia\_CV\_sweepstat: Valores de los parámetros de la frecuencia de corte de las TF obtenidas con la *sweep* estática como señal de entrada y control con CV.

Ajuste	f1	f2	f. central	f. res	Gain	Factor Q
res_1V (1R)	1588.21	1590.61	164.26	1589.40	2.78	661.55
res_4V (4R)	115.95	3348.38	2493.84	623.09	4.01	0.19
res_7V (7R)	3475.21	3802.46	3634.29	3635.15	19.92	11.11
res_10V (10R)	4046.13	4046.26	4046.26	4046.20	78.17	31998.50

Tabla 4.8: Resonancia\_CV\_sweepstat: Valores de los parámetros de la resonancia de las TF obtenidas con la *sweep* estática como señal de entrada y control con CV.

Ajustes	Frecuencia de corte (Hz)	Pendiente (dB/octava)	stdev
<i>freq_1V (32 Hz)</i> <i>Ampmax</i> <i>Ampmid</i> <i>Ampmin</i>	36.42 38.57 43.38	19.70 20.62 21.13	1.79 2.16 2.41
<i>freq_4V (256 Hz)</i> <i>Ampmax</i> <i>Ampmid</i> <i>Ampmin</i>	160.38 165.23 141.03	15.59 16.12 13.80	0.96 0.42 1.24
<i>freq_7V (2048 Hz)</i> <i>Ampmax</i> <i>Ampmid</i> <i>Ampmin</i>	968.47 878.07 889.42	15.45 13.72 13.93	1.03 1.49 1.18
<i>freq_10V (16 kHz)</i> <i>Ampmax</i> <i>Ampmid</i> <i>Ampmin</i>	5943.10 5461.19 5930.60	15.30 13.29 12.47	1.10 1.63 1.69
<b>Media</b> <b>Ampmax</b> <b>Ampmid</b> <b>Ampmin</b>		16.51 15.94 15.33	1.22 2.17 1.63

Tabla 4.9: Frecuencia\_CV\_sweepvar : Valores de los parámetros de la frecuencia de corte de las TF obtenidas con la sweep variable como señal de entrada y control con CV.

Ajustes	f1	f2	f. central	f. res	Gain	Factor Q
<i>res_1V(1R)</i> <i>Ampmax</i> <i>Ampmid</i> <i>Ampmin</i>	0.25 3.29 3.16	1033.09 976.32 925.23	171.46 75.87 63.22	127.83 448.05 427.70	1.38 0.50 1.19	0.02 0.06 0.06
<i>res_4V(4R)</i> <i>Ampmax</i> <i>Ampmid</i> <i>Ampmin</i>	0.25 3.29 4.93	2121.32 2128.52 2153.06	1580.37 1441.27 1536.61	183.17 661.56 814.89	3.51 3.40 3.23	0.01 0.04 0.05
<i>res_7V(7R)</i> <i>Ampmax</i> <i>Ampmid</i> <i>Ampmin</i>	2164.31 2183.66 2184.04	2378.52 2384.08 2383.07	2268.51 2283.93 2283.55	2268.89 2281.67 2281.38	19.60 19.97 20.64	10.59 11.38 11.46
<i>res_10V (10R)</i> <i>Ampmax</i> <i>Ampmid</i> <i>Ampmin</i>	2078.32 2078.20 1887.89	2078.58 2078.32 1888.02	2078.45 2078.32 1888.02	2078.45 2078.26 1887.95	75.71 90.51 99.48	8218.50 16435.50 14930.50

Tabla 4.10: Resonancia\_CV\_sweepvar: Valores de los parámetros de la resonancia de las TF obtenidas con la sweep variable como señal de entrada y control con CV.



## 5. Explicaciones adicionales

### Separación de IRs

El método que usaremos para separar las respuestas impulsionales se ha implementado en el *script Separacion\_IRs.py* y se basa en los siguientes pasos:

1. Encontrar los picos de la señal con el algoritmo *Peak Detection*<sup>17</sup> de la librería *Essentia*. Es necesario dibujar el tren de IRs para conocer los valores de los parámetros que nos interesa pasar al algoritmo, que son los siguientes:
  - a. *maxPeaks*: determina el número máximo de picos que queremos detectar. Corresponden a cada IR, por tanto, este valor será 11 o 3.
  - b. *minPeakDistance*: define la distancia mínima entre cada pico. Sabemos que la distancia entre las IRs es uniforme, por tanto, es suficiente con calcular una y ponerla. Hay que tener en cuenta que el algoritmo trabaja con distancias de 0 a 1, con lo cual hay que aplicar una sencilla regla de tres para convertir la distancia real (medida en muestras) en esta escala de 0 a 1.
  - c. *threshold*: define el umbral por debajo del cual no se detectarán los picos. Tiene que estar por debajo de la IR más pequeña
2. Una vez conocidos los picos, conocemos el instante de tiempo en el que se encuentra cada IR. Esta puede contener cola o un pre-eco con lo que tenemos que asegurarnos que no cortamos demasiado. La diferencia entre cada pico nos da un intervalo temporal  $T$  que podemos utilizar para seccionar, cogiendo la señal entre el instante del pico  $p - (T/2)$  y  $p + (T/2)$ .
3. Los archivos resultantes de este proceso serán los audios con nombres como *IR\_16F\_2R\_sweepvar\_ampmax* o *IR\_128F\_3R\_sweepstat* en los cuales únicamente hay una IR, correspondiente a la salida con los potenciómetros y la señal de control indicados en el nombre.

### Comparativa de magnitudes de *outputs* i *loopbacks*:

**Pregunta 1:** Vemos que, a mayor frecuencia de corte, menos diferencia hay y que ni el nivel (si la *sweep* de entrada es de amplitud máxima, media o mínima) ni la resonancia afectan.

- *IR\_16F\_0Res\_sweepvar\_ampmax*: > 50 dB
- *IR\_64F\_0Res\_sweepvar\_ampmax*: > 40 dB
- *IR\_512F\_2Res\_sweepvar\_ampmax*: 20 dB
- *IR\_4096F\_5Res\_sweepvar\_ampmax*: 0 dB

---

<sup>17</sup> [https://essentia.upf.edu/reference/std\\_PeakDetection.html](https://essentia.upf.edu/reference/std_PeakDetection.html)

Pregunta 9: Igual que antes, a mayor frecuencia, menor diferencia. Pero a mayor resonancia, también hay menos diferencia. El patrón es el mismo tanto si enviamos *sweep* estática como si enviamos *sweep* variable (como antes, el nivel no afecta a la diferencia).

- *IR\_CV\_freq\_1V\_sweepvar\_ampmax*: 50 dB
- *IR\_CV\_freq\_4V\_sweepvar\_ampmax*: > 30 dB
- *IR\_CV\_freq\_7V\_sweepvar\_ampmax*: < 20 dB
- *IR\_CV\_freq\_10V\_sweepvar\_ampmax*: 5 dB
- *IR\_CV\_res\_1V\_sweepvar\_ampmax*: > 20 dB
- *IR\_CV\_res\_4V\_sweepvar\_ampmax*: 20 dB
- *IR\_CV\_res\_7V\_sweepvar\_ampmax*: < 20 dB
- *IR\_CV\_freq\_1V\_sweepstat*: > 50 dB
- *IR\_CV\_freq\_4V\_sweepstat*: > 30 dB
- *IR\_CV\_freq\_7V\_sweepstat*: 20 dB
- *IR\_CV\_freq\_10V\_sweepstat*: 5 dB
- *IR\_CV\_res\_1V\_sweepstat*: > 20 dB
- *IR\_CV\_res\_4V\_sweepstat*: 20 dB
- *IR\_CV\_res\_7V\_sweepstat*: < 20dB

Preguntas 2, 3 y 4: El patrón es el mismo que en los casos anteriores, a mayor frecuencia de corte, hay menos diferencia.

- *IR\_16F\_0R\_sweepstat*: > 50 dB
- *IR\_32F\_0R\_sweepstat*: 50 dB
- *IR\_64F\_0R\_sweepstat*: < 40 dB
- *IR\_128F\_0R\_sweepstat*: < 40-30 dB
- *IR\_256F\_0R\_sweepstat*: < 30-20 dB
- *IR\_512F\_0R\_sweepstat*: < 20-15 dB
- *IR\_1024F\_0R\_sweepstat*: < 20-0 dB
- *IR\_2048F\_0R\_sweepstat*: < 15-5 dB
- *IR\_4096F\_0R\_sweepstat*: < 10-0 dB
- *IR\_8192F\_0R\_sweepstat*: < 5 dB
- *IR\_16kF\_0R\_sweepstat*: < 10 dB

## **6. Listado de material de grabación**

- Ordenador portátil (con *Reaper*<sup>18</sup> y *Audacity*<sup>19</sup>)
- Tarjeta de sonido *Edirol UA-101*<sup>20</sup>
- Sintetizador analógico *RSF Kobol Expander* <sup>21</sup>
- Fuente de voltaje externa *PROMAX FAC-363B*<sup>22</sup>
- 5 cables jack-jack desbalanceados

---

<sup>18</sup> <https://www.reaper.fm/>

<sup>19</sup> <https://audacity.es/>

<sup>20</sup> <https://www.roland.com/global/products/ua-101/>

<sup>21</sup> [https://www.vintagesynth.com/misc/rsf\\_kobol.php](https://www.vintagesynth.com/misc/rsf_kobol.php)

<sup>22</sup> <https://www.manualslib.es/manual/368829/Promax-Fac-363B.html>

## 7. Resumen Extendido

En este trabajo me propongo analizar el funcionamiento del filtro pasa bajos controlado por tensión de una unidad del sintetizador analógico RSF Kobol Expander. El RSF Kobol es un sintetizador monofónico (sólo puede producir una nota a la vez) francés lanzado en 1978 por los hermanos Ruben y Serge Fernandez. El RSF Kobol Expander fue una versión de este modelo original lanzada tres años más tarde y era, básicamente, una versión en rack del RSF Kobol.

Un sintetizador es un instrumento musical de tipo electrónico que, a través de circuitos, genera señales eléctricas que son convertidas en sonidos. Una propiedad que diferencia al sintetizador de otros instrumentos electrónicos es que sus sonidos pueden ser creados y modificados. Además, pueden imitar otros instrumentos o generar nuevos. En el caso del sintetizador analógico, este usa circuitos analógicos para generar estos sonidos. Este instrumento tiene la característica de ser imperfecto sónicamente, ya que padece de desviaciones, ruidos y otros aspectos técnicos que normalmente no se consideran en el audio digital, y eso es lo que nos interesa analizar, ya que las especificaciones técnicas no siempre son las que, realmente, tiene el módulo. Uno de los elementos más característicos y diferenciadores de un sintetizador es su filtro, a menudo de tipo pasa-bajos. Por ello, este trabajo se centra en él.

Un filtro pasa-bajos es, esencialmente, un proceso que se aplica a una señal para mantener los sonidos graves y eliminar los agudos. El hecho de que se pueda controlar por tensión significa que se pueden regular sus parámetros mediante un aparato externo, por ello nos referimos a él como VCF (Voltage Controlled Filter). La idea general es que los niveles de tensión analógicos producidos por otro instrumento o por una fuente externa de voltaje se pueden utilizar para controlar las funciones y los parámetros de un módulo, en este caso, del filtro. Para lograr este objetivo, hemos dividido el proyecto en una serie de pasos, todos ellos imprescindibles para llegar al último. Primero revisamos la documentación relacionada con la unidad para conocer las especificaciones técnicas de cada control, su historia y su funcionamiento.

A continuación, creamos una base de datos que podemos analizar para conocer el comportamiento real del filtro. Esta base consiste en una serie de grabaciones, que se han obtenido enviando señales de prueba, previamente creados y preparados, desde un ordenador hasta el filtro. Estas señales son sweep tones (señales que contienen un barrido de frecuencias, es decir, que recorren todo el espectro frecuencial) de amplitud fija y de amplitud variable y diversos tonos puros (señales con una sola frecuencia) y se graba la salida del filtro habiendo manipulado los potenciómetros de éste con diversas combinaciones, tanto manualmente como con una fuente de voltaje externa.

Aparte de las grabaciones manipulando los potenciómetros, hace falta una grabación con el filtro abierto, es decir, con el potenciómetro de la frecuencia de corte al máximo y el de la resonancia al mínimo, para tener una referencia de la respuesta sin que el filtro actúe (también denominada bypass). Nos sirve para comparar con los niveles de las tomas procesadas y obtener geoméricamente los valores de los parámetros.

También hace falta una grabación del paso de la señal por la tarjeta de sonido utilizada para conectar el ordenador con el sintetizador, sin llegar al filtro, para tener la señal que le llegará al VCF. Éste contiene tanto la señal de control como el ruido y coloración aportados por la tarjeta. Estas grabaciones sirven para conocer los valores exactos de los parámetros, así como el comportamiento del filtro con diferentes señales de entrada. Estos parámetros hacen referencia a los 2 potenciómetros con los que hemos trabajado, uno que determina la frecuencia de corte del filtro (a partir de que frecuencia se empieza a eliminar sonido) y otro que determina el tamaño de la resonancia (la cual consiste, básicamente, en un aumento del nivel de la señal de salida, en forma de pequeña campana de Gauss, justo antes de empezar a disminuir).

En la tercera y última parte del proyecto, analizamos detalladamente estas salidas, usando diversos códigos escritos en Python. Para este análisis, nos basamos en ciertas preguntas sobre su comportamiento con las diferentes señales y sus respectivas salidas. Por ejemplo, hay preguntas como:

¿Hay diferencias en la respuesta en frecuencia del filtro para diferentes niveles de señal de audio de entrada? ¿Cuál es el rango de la frecuencia de corte? ¿Cuál es el rango de la resonancia? ¿Cuál es la pendiente real? ¿Responden igual los parámetros controlados por una fuente de voltaje externa?

Como hemos mencionado, el VCF tiene dos parámetros principales, la frecuencia de corte del filtro y la resonancia. No obstante, hay diversos parámetros secundarios, pero no por eso menos importantes. Estos son la pendiente del filtro (cuando se llega a la frecuencia de corte, el sonido no es repentinamente eliminado, sino que es un proceso gradual), el ancho de banda de la resonancia (que frecuencias ocupa la campana), la ganancia (la diferencia entre el máximo nivel de esta resonancia y el nivel previo del sonido), la frecuencia central o de resonancia (en que frecuencia se encuentra el nivel máximo de la resonancia) y, finalmente, el factor Q (indica la forma de la señal en la resonancia, y cuanto más alto es este valor, más estrecha es la forma). Todos estos valores se pueden conocer geométricamente usando diversas librerías de Python (como matplotlib, numpy, scipy o Essentia) para hacer cálculos con los datos de las grabaciones obtenidas.

Otro dato importante es que para el análisis usamos las respuestas impulsionales de las salidas. Estas consisten en señales donde el nivel es 0 en todo el dominio, excepto en el final, cuando hay un pico. Básicamente nos dicen cómo reaccionan las señales a un impulso (un sonido repentino y fuerte). Al pasarlo a dominio frecuencial y aplicar algunas operaciones (explicadas con detalle en el trabajo) estos impulsos nos dan la respuesta frecuencial de las salidas con respecto a la señal de entrada (llamada función de transferencia). Y eso es lo que podemos usar para calcular los parámetros que hemos mencionado antes. Con este análisis, podemos comparar los valores que dan en las especificaciones técnicas con los valores encontrados empíricamente y conocer el comportamiento real del filtro, de manera que dicho comportamiento real pueda quedar caracterizado paramétricamente.