VIRTUALIZACIÓN DE UN FILTRO CONTROLADO POR TENSIÓN (VCF) EN UN SINTETIZADOR ANALÓGICO

Treball Fi de Grau de

Ricard Martín Fernández

Directors: Perfecto Herrera

Xavier Lizarraga

Grau en Enginyeria en Sistemes Audiovisuals

Curs 2021-2022

*[Pàgina en blanc]*

*[S’han marcat els apartats del treball que haurien d’anar en pàgina senar o dreta de la publicació (situació de pàgines que ajuda a estructurar i a donar prioritat formal als diferents apartats).*

*Hi ha apartats on no s’indica res. En aquests cas l’estudiant pot triar la situació, en pàgina parell o senar, segons li convingui per al compaginat final.*

*Es recomana que les pàgines blanques no portin número de foli (es compten, però el número no s’imprimeix)]*

*Si s’imprimeix a simple cara no s’han de deixar pàgines en blanc*

A mi família y mis amigos músicos

Dedicatòria [opcional] [12 punts]

*[Pàgina en blanc]*

**Agraïments**

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

Text dels agraïments [12 punts]

*[Pàgina en blanc]*

**Resumen**

El objetivo de este trabajo es modelar e implementar un plugin que emule el funcionamiento del módulo de VCF de una unidad del sintetizador analógico RSF Kobol Expander, disponible en la fundación Phonos. Para ello, primero revisaremos la documentación relacionada con la unidad para conocer las especificaciones técnicas de cada control. A continuación, grabaremos una serie de datos, los cuales analizaremos para ver el comportamiento real de cada parámetro y usaremos para la posterior implementación, la cual se hará en un plugin VST.

Resum del treball en la llengua que es presenti [12 punts]

Extensió màxima recomanada: 150 paraules

**Resum [en una 2a llengua. Ex. Resumen]**

L'objectiu d'aquest treball és modelar i implementar un plugin que emuli el funcionament del mòdul de VCF d'una unitat del sintetitzador analògic RSF Kobol Expander, disponible a la fundació Phonos. Per això, primer revisarem la documentació relacionada amb la unitat per tal de conèixer les especificacions tècniques de cada control. A continuació, enregistrarem una sèrie de dades, les quals analitzarem per veure el comportament real de cada paràmetre i farem servir per a la posterior implementació, la qual es farà en un plugin VST.

****

Resum del treball en una llengua diferent a la utilitzada [12 punts]

Extensió màxima recomanada: 150 paraules

**Abstract [en una 3a llengua. Ex. Abstract]**

The objective of this work is to model and implement a plug-in that emulates the functioning of the VCF module of an RSF Kobol Expander analog synthesizer unit, available in the Phonos foundation. To do this, we will first review the documentation related to the unit to know the technical specifications of each control. Next, we will record a series of data, which we will analyze to see the real behavior of each parameter and we will use for the subsequent implementation, which will be done in a VST plugin.

Resum del treball en una llengua diferent a la utilitzada [12 punts]

Extensió màxima recomanada: 150 paraules

*[Aquest apartat,* ***sempre*** *hauria de començar en pàgina senar, pàgina dreta de la publicació]*

*[Pàgina en blanc]*

**Índice**

*[Taula de contingut del treball, parts en què està dividida] [12 punts]*

1. Introducción
2. Síntesis de sonido controlada por tensión
3. RSF Kobol Expander
   1. Historia y funcionamiento general
   2. Parámetros y controles
   3. Emulaciones existentes

1. Metodología

4.1. Grabación

4.2. Procesamiento de señales

4.3. Función de transferencia

4.4. Modelaje a partir de la respuesta impulsional

1. Análisis de muestras
   1. Análisis global
   2. Análisis de cada parámetro
2. Implementación
   1. Implementación del prototipo
   2. Implementación del VST
3. Conclusiones
4. Bibliografía
5. Apéndices

*[Aquest apartat també ha de començar en pàgina senar]*

**Llista de figures [opcional]**

Figura 1: RSF Kobol Expander (completo)

Figura 2: RSF Kobol Expander (sin teclado)

Figura 3: VCF

Figura 4: PolyKB II

Figura 5: Configuración inicial

Figura 6: Loopback

Figura 7: Balanceado vs No Balanceado

Figura 8: Montaje de grabación

Figura 9: Función de transferencia de un filtro

**Llista de taules [opcional]**

**1. Introducción [arial 14 punts]**

En este primer apartado del trabajo expondré la motivación de realizar dicho proyecto y sus objetivos y ambiciones.

Para empezar, vamos a definir que es un sintetizador analógico. Un sintetizadores 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 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. 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 géneros más duros como el industrial metal o el nu metal.

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 no se consideran en el audio digital. 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 esa similitud. Son esas imperfecciones las que hacen que el sonido analógico sea único y que hoy en día fabricantes de plugins busquen recuperar y modelar estos sonidos, inundando el mercado de emulaciones analógicas. 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 estudios de grabación apuestan por un sistema híbrido donde conviven el mundo analógico y el digital.

Personalmente, el mundo de los sintetizadores es un mundo que, como músico, me intriga mucho y en el que nunca he participado activamente. Por eso, me interesa modular un plugin propio que pueda emular el funcionamiento de una parte del sintetizador (hacerlo de todo el sinte sería un proyecto mucho más largo y complejo) para poder usarlo en la creación de música.

Otra motivación es que al ser un sintetizador muy antiguo (lo veremos en el capítulo 3) es posible que haya componentes que ya no se fabriquen y que estén en mal estado, por lo que el sintetizador original puede quedar obsoleto. Además, como también veremos en el apartado 3, el sintetizador con el que quiero trabajar es muy raro, cosa que le aporta un valor especial.

Por esos motivos, el objetivo de este proyecto es samplear y modelar el procesador analógico del VCF. Un VCF (Voltage Control Filter) es un filtro cuya frecuencia de corte depende del valor del voltaje en su puerto de control [1]. En otras palabras, lo que éste hace es modificar la envolvente de la señal en torno a la frecuencia de corte del filtro usando una señal externa.

Hoy en día existen distintos tipos de modelado basados en distintas técnicas:

* Virtual Analog Modelling

Consiste en la simulación de circuitos eléctricos para aplicaciones relacionadas con el sonido en tiempo real. Algunas técnicas de VA modelling son los métodos State-Space (SS), las Wave Digital Filters (WDF’s) o los sistemas Port-Hamilton (PHS) [2].

* Impulse Response Modelling

Consiste en hacer una convolución lineal entre la señal original y la respuesta impulsional (IR) del sistema. Este método permite emular sistemas no lineales, caracterizados por distorsión armónica de varios órdenes. [3]

* Deep Neural Network Modelling

Se pueden usar DNN’s para aprender las características de un módulo, en este caso del VCF. De este modo, el algoritmo es capaz de predecir la configuración de los parámetros y emular dicho módulo.[4]

Nosotros usaremos el segundo tipo de modelling que hemos mencionado, dado que es el que obtiene unos resultados más reales. En el capítulo 4 explicaremos con más detalle qué metodología seguiremos para obtener unos resultados con los que poder implementar el plugin.

En los dos últimos tipos de modelado, es necesario tener una base de datos con sonidos o señales de test para extraer resultados y conclusiones y poder modelar. La calidad de esta base de datos (resolución y calidad del audio) definirá la calidad del resultado final del modelado. Cuantas más muestras tengamos, más preciso será nuestro modelo, siempre que hagamos un buen análisis.

En este proyecto, lo interesante es definir y almacenar señales de prueba y registrar la salida del VCF. De esta manera tenemos la opción de analizar la respuesta en frecuencia del filtro ante una señal que conocemos y, por otro lado, teniendo la entrada y la salida tendremos la opción de hacer algún experimento modelando con redes neuronales.

Para poder modelar un dispositivo analógico, lo primordial es conocer sus posibilidades y limitaciones. Analizando las muestras registradas podremos conocer el rango de la frecuencia de corte del LPF, y su velocidad máxima y mínima (con la velocidad nos referimos al tiempo que tarda en responder a un cambio en los controles y parámetros implicados en el VCF).

Lo primero de todo es capturar el efecto de cada parámetro por separado con el objetivo de conocer su rango, desviación y comportamiento o respuesta (si es lineal, logarítmico 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 THD (Total Harmonic Distortion), la SNR (Signal to Noise Ratio)  y la envolvente en frecuencia del ruido de fondo de la salida de cada módulo, puesto que también forman parte de la señal de salida.

Veremos cómo funcionan los parámetros del filtro en la sección 3, además de explicar brevemente la historia y el funcionamiento general del sintetizador que contiene el módulo que queremos trabajar, el RSF Kobol Expanded, y daremos más detalles sobre la grabación y el posterior análisis de estas en las secciones 4 y 5.

Finalmente, en la sección 6, veremos cómo hemos implementado el plugin a partir de los pasos anteriores.

*[Cada capítol ha de començar en pàgina senar]*

**2. Síntesis de sonido controlada por tensión**

Antes de entrar en materia, vamos a explicar brevemente cómo funciona un VCF y el concepto de control por tensión eléctrica.

El VCF se basa en los siguientes pasos:

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

Es decir, tenemos un LPF variable según la envolvente de la señal de entrada. Aunque es posible desconectar esa modulación, y hacer que los parámetros del filtro sólo se modifiquen manualmente.

El concepto de control por tensión (CV) está en la raíz misma del sintetizador modular. 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. Por ejemplo, un CV puede determinar el tono de un oscilador mientras que un segundo CV puede determinar la potencia de una señal después de que pase a través de un VCA (amplificador controlado por tensión). Por ejemplo, cuando un módulo de sintetizador dice que presenta "CV sobre la resonancia del filtro", eso significa que hay una entrada de CV para controlar la cantidad de resonancia que tiene el filtro usando una fuente de señal externa [5]. Un CV externo podría ser, por ejemplo, un pedal, un secuenciador o una fuente de voltaje.

Como veremos en el siguiente capítulo, casi todos los controles de la parte delantera del Kobol pueden ser regulados con un CV externo, lo cual es otra particularidad importante de este sintetizador.

**3. RSF Kobol Expander [arial 14 punts]**

3.1 Historia y funcionamiento general [14 punts]

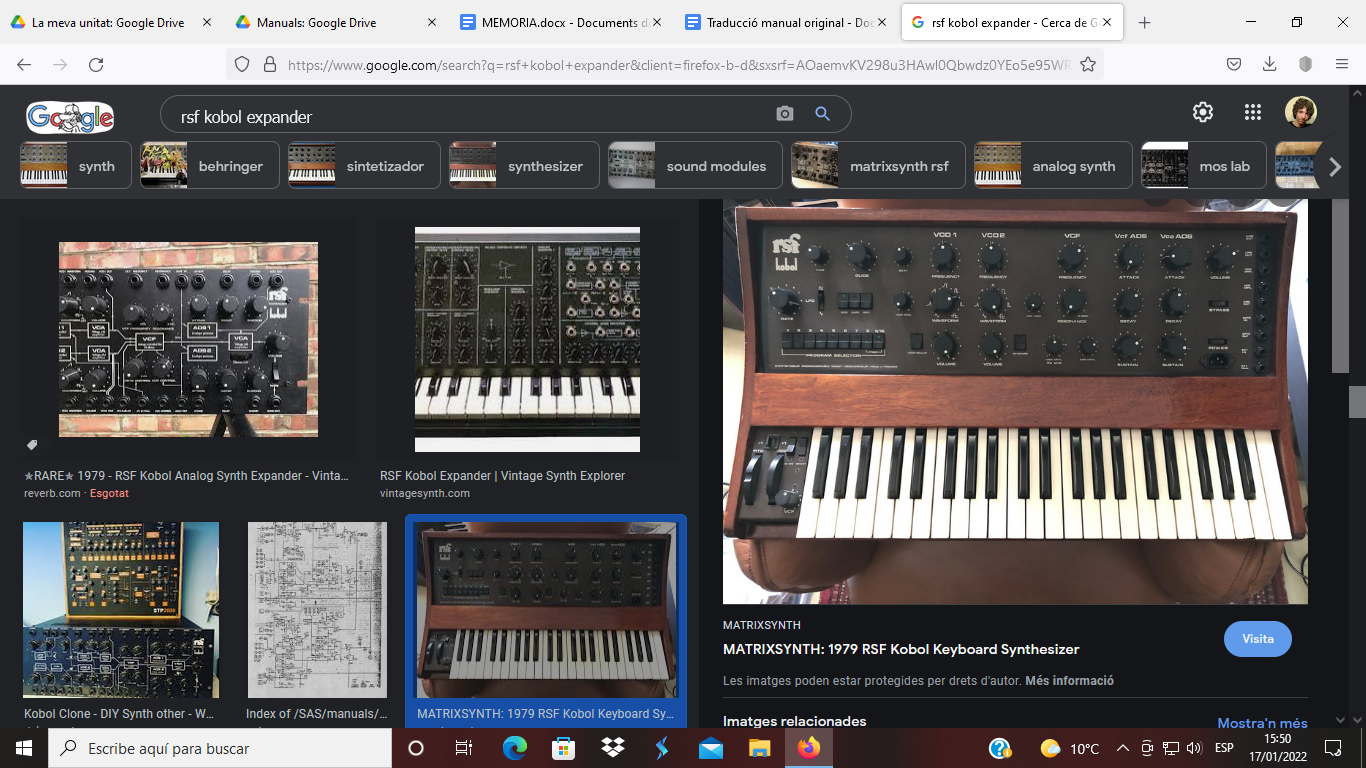


Figura 1: RSF Kobol Expander (completo)



Figura 2: RSF Kobol Expander (sin teclado)

El RSF Kobol es un sintetizador monofónico (sólo puede producir una nota a la vez) francés lanzado en 1978. Fue creado por los hermanos Ruben y Serge Fernandez. Es un instrumento muy versátil que ofrece muchas características buscadas de otros sintetizadores analógicos de su época.[6]

Más tarde, en 1981, se lanzó el RSF Kobol Expander (Figuras 1 y 2), el cual era una versión en rack del RSF Kobol original. Entre otras cosas, contenía 2 VCO (Voltage Control Oscillator), 2 controles de ADS (Attack, Decay y Sustain), 1 LFO (Low Frequency Oscillator), 1 VCF, 1 VCA (Voltage Controlled Amplifier), 1 generador de ruido y 1 procesador de voltaje.[7]

Aunque es principalmente un sintetizador analógico tradicional con osciladores estables y muchos controles que le permiten controlar la envolvente ADS, la sección de filtro y el LFO, ofrecía algunas características únicas para su época. En particular, las formas de onda del oscilador se pueden barrer continuamente para obtener una transición muy interesante entre los diferentes tipos de onda que puede generar (sinusoide, sierra, pulso, etc). Esto, así como la activación y desactivación de notas y otros parámetros, se pueden controlar a través de CV.

La versión de rack venía en cuatro componentes diferentes. El Kobol Rack (Expander 1) contenía, esencialmente, las secciones VCO/VCF/VCA/LFO del Kobol en un módulo de rack (tal como hemos mencionado antes). El Expander 2 fue un complemento del Kobol que agregó algunos módulos de procesamiento nuevos, como modulación en anillo, retención de muestras y seguidores de envolvente, así como módulos VCA y LFO adicionales. El Programmer contenía los circuitos de memoria para almacenar *patches* y 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 Kobol se veía y sonaba muy bien, y rápidamente fue absorbido por las grandes empresas de sintetizadores de la época. El problema era que, al ser una pequeña compañía francesa, RSF nunca pudo producirlos en masa en el mismo tipo de escala que Moog, Arp, Roland o Korg. Se fabricaron menos de 200 teclados Kobol, alrededor de 800 RSF Kobol Expander, 200 Expander 2, 200 Programmer y solo unas decenas de KM8. Esto hace que sean sintetizadores raros y muy caros (alrededor de 5000 euros [8]). Han sido utilizados por artistas y grandes grupos de música como Vince Clarke, Vangelis, Jean-Michel Jarre, Depeche Mode, Peter Gabriel y Hans Zimmer.[9]

Como introducción al siguiente subapartado, vamos a definir brevemente los módulos que contiene el RSF:

**- Keyboard:** Es el módulo que contiene las teclas del teclado. Genera voltajes continuos al tocar las teclas.

**- VCA (Voltage Controlled Amplifier):** Es un amplificador cuya ganancia depende del valor del voltaje en su puerto de control.

**- VCF (Voltage Controlled Filter):** Es un filtro pasa bajos cuya frecuencia de corte depende del valor del voltaje en su puerto de control.

**- VCO (Voltage Controlled Oscillator):** Es un oscilador que genera una onda con una frecuencia que depende del valor del voltaje en su entrada.

**- LFO (Low Frequency Oscillator):** Es un oscilador que genera una onda de baja frecuencia en su salida (es un tipo particular de VCO).

**- Envolvente ADS:** Consiste en una serie de parámetros (Attack, Decay, Sustain) que modelan la envolvente de una onda. [1]

Ahora si, en el siguiente subapartado, veremos en detalle el funcionamiento de los diversos módulos del Kobol Expander y como se pueden aislar para estudiar y modelar únicamente el comportamiento del VCF.

3.2. Parámetros y controles

El RSF Kobol Expander tiene tres modos de funcionamiento: manual, de memoria y secuenciador. El manual permite utilizar el sintetizador como un dispositivo convencional. El modo de memoria sirve para grabar y almacenar lo que toquemos en modo manual o secuenciador. Y por último, el modo secuenciador sirve para reproducir una secuencia de notas [10]. Dado que la unidad con la que trabajaremos no contiene el teclado, solo el rack de los módulos, únicamente podremos trabajar en modo manual.

En modo manual encontramos los siguientes controles y parámetros:

* Voltage Controlled Oscillators (VCO)

El RSF tiene dos VCO que van desde 10Hz a 10kHz [11], en un rango de 4 octavas cuantizadas por semitonos [10] (ampliable por CV, que también puede variar su tono y su nivel) con una salida para cada uno. Lo que hace inusual a este VCO es el control de selección de forma de onda; en lugar del selector de forma de onda conmutada habitual que se encuentra en la mayoría de los sintetizadores, los controles del RSF son totalmente variables entre las formas de onda. Éstas pueden ser: triángulo, diente de sierra ascendente, cuadrado y pulso. Al configurar el control a medio camino entre dos formas de onda que se muestran en el panel, se pueden obtener mezclas interesantes y hay muchos sonidos inusuales disponibles. Esto también puede ser controlado por CV, lo que significa que puede barrer automáticamente las formas de onda aplicando la salida de uno de los envolventes generadores o el oscilador de baja frecuencia a la entrada de control de forma de onda.[11]

Los dos VCO son idénticos y proporcionan una amplia gama de sonidos. El volumen de cada oscilador está controlado de manera separada, es decir, hay un potenciómetro de volumen independiente para cada VCO. También hay una función de sincronización, por lo que, como se puede ver, hay infinitas posibilidades para la creación de formas de onda únicas.

* Voltage Controll Filter (VCF)

En la figura 3 podemos ver con claridad el módulo de VCF, que para este proyecto es el más importante de los 4 que el RSF Kobol Expander contiene.

 Figura 3: VCF

Este es un filtro pasa bajos (LPF) de 4o orden, con una pendiente de 24 dB/octava. Hay cuatro parámetros asociados con el filtro: frecuencia de corte (VCF Frequency), resonancia, cantidad de ADS (Attack, Decay y Sustain) y control de teclado (KeyB Controll). También hay entradas para el CV externo de cada uno de estos controles y también una entrada de audio para procesar una señal externa [11].

De estos 4 parámetros que hemos nombrado, dos son manuales:

* VCF FREQUENCY: define la frecuencia de corte. Se duplica con el botón VCF ubicado a la izquierda del teclado. [10]
* RESONANCE: define la campana de resonancia que se genera en la frecuencia de corte para dar un cierto carácter al sonido generado. Si la rueda está girada al extremo izquierdo, el filtro eliminará gradualmente las oscilaciones de los armónicos altos. Si está girada a la derecha, el filtro entrará en resonancia propia (oscila de manera automática), creando una onda sinusoidal audible, cuyo tono estará determinado por el control de frecuencia.[10]

Y dos modifican el comportamiento del VCF:

* KEYB. CONTROL: Realiza un seguimiento de la frecuencia que le transmite el teclado y mueve la frecuencia de corte en consonancia con esa frecuencia que le llega.

En otras palabras, permite que el VCF se sintonice con el teclado. Si está en la posición 0, el VCF será independiente de la nota tocada. Si está en la posición 1, se sintonizará con la nota tocada, eso hará que el sonido sea homogéneo en todo el teclado. Finalmente, colocado en la posición 2, obtendremos un sonido muy rico en armónicos en las notas altas del teclado [10].

* ADS CONTROL: controla el ADS del VCF. La frecuencia de corte del filtro evolucionará según la envolvente creada con los potenciómetros de Attack, Decay y Sustain.

Attack es el tiempo que tarda el sonido en escucharse desde que se aprieta la tecla; Decay es el tiempo que tarda el sonido en bajar hasta el volumen de *sustain* y Sustain es el tiempo que una nota se mantiene en un volumen constante. Más adelante veremos que el otro parámetro que normalmente se ve en ADSR, Release(R), no se tiene en cuenta porque está vinculado al decay.

Para modelar de forma precisa, es necesario conocer el rango exacto del parámetro VCF FREQUENCY, y si su escala es logarítmica o lineal en frecuencia. En la serigrafía se indica un rango de 16Hz a 16KHz, pero en la sección de análisis lo comprobaremos, ya que, en la práctica, pueden haber desviaciones.

Con el parámetro de resonancia el objetivo es el mismo, conocer el ancho de banda y l’altura de la campana para valores discretos de este parámetro.

Por otro lado, el filtro se puede controlar con señales de control. Es interesante conocer la respuesta del VCF con señales de control para aislarlo de las señales que genera el VCO, que pueden tener sus propias desviaciones. Para ello podemos rutear señales de prueba a estas entradas y ver cómo responde el filtro. Eso es lo que haremos en este proyecto (lo veremos en la sección 4), usar una señal de control externa.

* Envelope Generator

Hay dos generadores de envolvente en el RSF, uno para el VCA y otro para el VCF (VCA ADS y VCF ADS respectivamente).[11] El VCA controla la envolvente del sonido final, después de haber pasado por los otros módulos, antes de salir del sintetizador.

Los generadores de envolvente funcionan de manera que el tiempo de release final está vinculado al control de decay inicial y se activa o desactiva. El rango de control con los tiempos de attack y decay es muy amplio, desde 1 ms hasta más de 20 segundos. Una vez más, hay entradas que permiten variar estos tiempos y el nivel de sostenido con un CV externo.

Al aplicar el CV del teclado directamente a la entrada de decay, las notas bajas tendrán un decay más corto que las notas altas. En cambio, al enrutar el CV del teclado a través del inversor en el procesador de voltaje, las notas bajas tendrán una caída prolongada que se acortará progresivamente cuanto más alto toque el teclado, simulando así instrumentos acústicos más de cerca [11].

* Low Frequency Oscillator (LFO)

El LFO genera señales cuadradas o triangulares para modular los VCO y el VCF [10]. La tasa de modulación es muy amplia, desde 0,01 Hz hasta 100 Hz y esta tasa también es controlable por CV. El generador de ruido emite ruido blanco o rosa que se puede seleccionar mediante un interruptor interno. La salida es muy alta y también se puede utilizar como fuente de control [11].

Existen otros parámetros en el sintetizador, aunque son menos importantes para este trabajo, así que los definiremos brevemente, sin entrar en detalles técnicos: [10]

* BYPASS: Cambia para mantener un sonido indefinidamente. Equivalente a mantener presionada una tecla en el teclado.
* TUNE: Afinación del sintetizador en comparación con otros instrumentos de una orquesta. Si está en 0, el dispositivo es calibrado en la nota LA de 440Hz.
* BEAT: Ajuste de ritmo entre los 2 VCO. El 2 está sintonizado con el 1.
* VCO-MOD, VCF MOD: Modulación permanente de VCO y VCF a través del LFO.
* VCO 1 MOD KEY off: Elimina la modulación en VCO 1 cuando se presiona el botón.
* SYNCHRO KEY: sincronización con VCO 1 cuando la tecla está apretada.

Finalmente, llegamos al procesador de voltaje. Este es un dispositivo realmente útil que es capaz de aumentar los voltajes hasta 4 veces más, y hay salidas normales e invertidas [11].

En la Figura 1 podemos ver que a la izquierda del teclado, debajo de todos los parámetros que acabamos de describir, hay una pequeña placa con más controles. Estos son los siguientes: [10]

* RUEDA DE TONO: Controla el tono de los osciladores en un rango de más o menos un quinto.
* RUEDA MOD: Modula los osciladores a través del LFO.
* INTERRUPTOR DECAY: en posición OFF elimina el tiempo de retorno del sonido (si dejamos de apretar la tecla, el sonido se apaga instantáneamente).
* INTERRUPTOR DE DESLIZAMIENTO (GLIDE): en la posición ON, se realiza un enlace entre las diferentes notas tocadas en el teclado a una velocidad variable determinada por el potenciómetro GLIDE, situado arriba a la izquierda.
* INTERRUPTOR DE OCTAVA : Modifica el teclado aumentando o reduciendo en 1 octava su rango.
* BOTÓN VCF: Modifica la frecuencia de sintonía del filtro. Su acción se combina con el potenciómetro de FRECUENCIA del VCF.

3.3. Emulaciones existentes



Figura 4: PolyKB II

Para familiarizarnos con la unidad antes de ir al estudio a grabar, hemos usado una versión bastante parecida de un plugin en VST llamado PolyKB II (Figura 4), el cual es bastante parecido al Kobol original, con algunas modificaciones però con un sonido y una estètica muy parecidos. En este contexto, sería interesante comparar el sonido de este plugin con el que consiga implementar al final del proyecto.

Los osciladores del Polykobol II se han implementado basándose en el RSF Kobol original, incluida su capacidad para cambiar de una forma de onda a la siguiente. Junto a estos, el filtro de paso bajo de 24dB/oct es suave y potente. En resonancia máxima, oscila suavemente y la onda resultante sigue al teclado. También tiene una función bastante ingeniosa (no derivada del sintetizador original) que le permite colocar overdrive (distorsión) en la entrada del filtro o en su salida; las dos opciones crean efectos diferentes e interesantes.

PolyKB cuenta con dos generadores de contorno ADSR, el ADSR2 (que controla el VCF) y el ADSR1 (que controla el VCA). El botón Loop2 recorre las etapas 'A' y 'R' del ADSR2.

Por otro lado, el key tracking (el equivalente del control del teclado en el modelo original) solo afecta las etapas de ataque de las envolventes, lo que significa que es inútil para imitar las respuestas de instrumentos punteados o martillados como guitarras, pianos y percusión.

Los 2 LFO permiten seleccionar cualquier número de formas de onda simultáneamente para crear formas de onda complejas.

En la sección de Modulations vemos cinco fuentes de modulación (LFO1, Ruido, ADSR2, VCO2 y velocidad) enrutadas (a través de la rueda de modulación) a los tonos y formas de onda de los osciladores, así como a la frecuencia de corte del filtro.

En la parte izquierda, hay un arpegiador simple (reproduce una secuencia de notas) con modos arriba, abajo y arriba/abajo. Si no se selecciona ninguno de estos, se arpegia aleatoriamente.

Debajo de este, el panel de OUTPUTS te permite seleccionar el número de voces disponibles (de 1 a 16), el rango de inflexión, la sensibilidad a la velocidad, la sensibilidad a la presión y la cantidad de desafinación. Encima se encuentran los menús que permiten seleccionar Unison (de dos a seis voces) y si queremos sonido polifónico o monofónico (esto es una mejora importante, ya que el modelo original era monofónico).

Debajo de la sección de modulación tradicional se encuentra un panel que ofrece dos buses que permiten seleccionar dos fuentes de modulación de un total de 12 y dirigir cada una hasta cuatro destinos seleccionados de un total de 29. Esto es muy útil, pero no se puede determinar la cantidad de modulación aplicada a cada destino.

Una característica crucial del plugin es que no tiene aliasing. Además, no hay saltos grandes entre frecuencias, solo una respuesta suave desde las frecuencias de corte más bajas hasta las más altas. [12]

En resumen, es un sintetizador bastante parecido al modelo original, con algunas mejoras y algunos fallos o cosas que se pueden mejorar, aunque es normal ya que es un prototipo de una implementación en la que la empresa XILS Labs sigue trabajando.

**4. Metodología**

En esta sección explicaremos con detalle qué procesos realizaremos con tal de obtener los resultados que analizaremos en la sección 5 y que, posteriormente, usaremos para implementar el plugin en VST (Virtual Studio Technology). Esencialmente, seguiremos el método de modelaje con respuestas impulsionales que vimos en [3] y que introducimos brevemente en la sección 1.

4.1. Grabaciones

El primer paso es realizar una série de grabaciones con tal de ver cómo responde el filtro a una señal externa. Pero antes de eso tenemos que preparar esas señales, que llamaremos señales de control. Las grabaciones se harán para responder una serie de preguntas, las cuales responderemos en el apartado 5 de este trabajo.

Estas preguntas nos ayudan a conocer el comportamiento del VCF con tal de implementar el plugin de la manera más real y fidedigna posible. Y 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 en frecuencia?

3. ¿Cuál es el rango de resonancia?

4. ¿Cuál es el comportamiento de los parámetros? Lineal, logarítmico, exponencial?

5. ¿Cuál es la SNR del filtro?

6. ¿Cuál es el nivel de 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? .

En la siguiente página veremos qué señales son las que hacen falta para responder cada pregunta.

Además de las salidas del filtro manipulando los parámetros, también necesitamos captar la salida en bypass (sin ningún proceso, con el potenciómetro de la frecuencia de corte al máximo y el de la resonancia al mínimo) para tener una referencia de la respuesta en FLAT. Nos servirá para comparar con los niveles de las tomas procesadas.

Un paso muy importante antes de realizar las grabaciones es la calibración del sintetizador. Es necesaria para asegurarnos de que siempre estamos en las mismas condiciones de nivel. Para este paso, es muy importante tener en cuenta el nivel de referencia que utilizaremos para la calibración, el cual depende de la tarjeta de sonido que usemos.

Para realizar las grabaciones con el ordenador portátil, usamos una tarjeta de sonido modelo Edirol UA-101, la cual, usando las entradas de jack, tiene un nivel de voltaje de salida máximo de 16 dBu y un voltaje nominal (de referencia) de 4dBu [13].

El nivel máximo se corresponde con los 0 dBFS del señal digital, al cual, para tener el señal en el nivel de referencia, hay que restarle 12 dBs FS. Pero estos son los dBs referencia de rms, si queremos el voltaje de pico, tenemos que sumarle 3 dB’s, con lo cual, finalmente tenemos que restarle 9dBs FS al señal original, con tal de referenciarlo y calibrarlo correctamente.

Realizando estos ajustes, los señales de entrada que tenemos para las diferentes preguntas son los siguientes:

Pregunta 1: Sweep variable con 3 amplitudes diferentes, empezando con el nivel máximo referenciado y una diferencia de 10 dBs y 5 segundos entre una y la siguiente (-9 dBFS, -19 dBFS, -29 dBFS)

Preguntas 2,3 y 4: Sweep estática al nivel máximo (-9 dBFS). Es un tren de 11 señales con 5 segundos de espacio entre dos señales consecutivas, para poder grabar la salida con 11 puntos diferentes de los potenciómetros de frecuencia de corte y resonancia.

Preguntas 5 y 6: 4 tonos puros de 100 Hz, 440 Hz, 1 kHz y 10 kHz. (todos al nivel de referencia de -9dBFS).

Preguntas 7 y 8: No hace falta ninguna grabación, ya que son preguntas que responderemos después del análisis de las tomas para las preguntas 2, 3 y 4

Pregunta 9: Usaremos las mismas señales que para las preguntas 1,2,3 y 4 pero en vez de cambiar los parámetros manualmente, lo haremos usando una fuente de voltaje externa.

Además del nivel, todos los señales tienen en común una série de parámetros:

frame rate: 48 kHz

formato: wav y 24 bits de encoding

canales: mono (1 canal)

duración: 15 segundos

Estos señales los hemos creado usando los plugins de Aurora en Audacity [14].

Las salidas que grabemos tienen que tener un nombre fácilmente interpretable y único, ya que habrá muchas grabaciones. Las señales de entrada tendrán el nombre del tipo de señal que son, por ejemplo, *sweep\_static*. Por otro lado, las salidas tendrán el nombre de los parámetros que toquemos y del tipo de señal de entrada de origen, por ejemplo *VCF\_16F\_0Res\_sweepvar (*que seria la salida con la frecuencia de corte a 16 Hz y la resonancia a 0, habiendo enviado la sweep variable como señal de entrada) o *Bypass\_sweepstat* (que sería la salida en bypass enviando la sweep estática como señal de test).

Algo muy importante y útil para el proyecto es que en este sintetizador, se puede acceder a cada proceso por separado, lo que implica que el VCF acepta señales externas sin tener que pasar por los módulos anteriores (VCO y LFO), de modo que el señal le llega limpio, sin ninguna interferencia causada por los otros módulos, únicamente con las posibles interferencias provocadas por el cable).

Aún así, por si acaso, dado que queremos observar únicamente el filtro, todos los demás controladores (VCO, LFO, VCA…) deben estar al mínimo (a 0) de modo que no afecten a la salida del VCF.

No obstante, el primer día de grabación descubrimos que los potenciómetros del VCA (el siguiente módulo por el cual pasa el sonido después de pasar por el VCF) tienen que estar en una posición concreta para que el sonido no se vea afectado, concretamente el Sustain del ADS 1 debe estar al máximo. Sucede igual con el potenciometro del volumen final, antes de salir del sinte, tal como vemos en la Figura 5, la cual muestra también 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 (la salida en este estado sería el bypass).

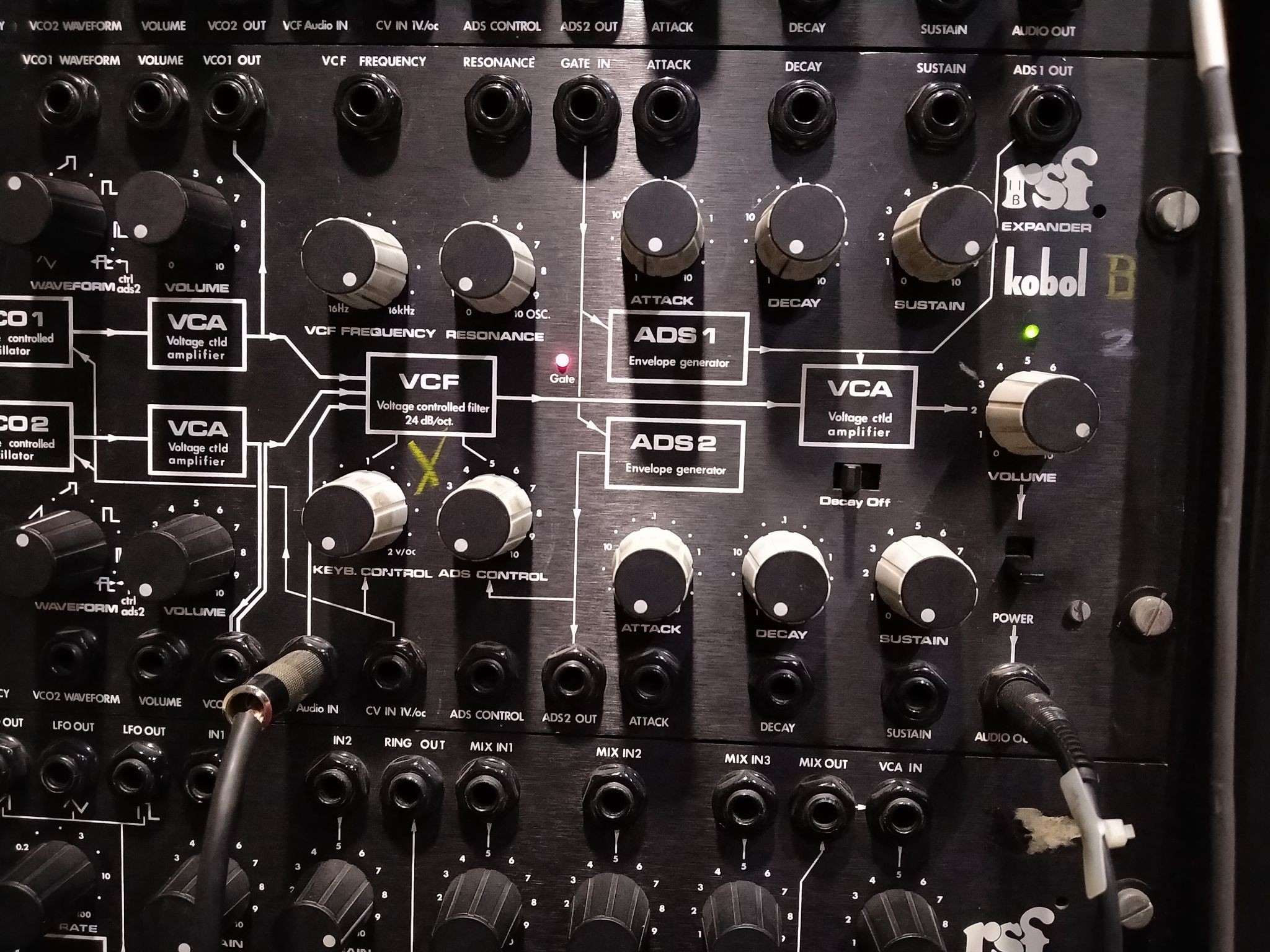


Figura 5: Configuración inicial

El método de grabación consistirá en enviar el señal al módulo desde un ordenador y grabar la salida del VCF, moviendo los potenciómetros de frecuencia y resonancia . Cada pregunta tendrá un método diferente, aunque en todas (excepto la 9) los cambios en los potenciómetros se harán manualmente:

Pregunta 1: (sweep variable)

La idea es trabajar con 4 puntos del potenciómetro de frecuencia (16, 64, 512 y 4096) y con 5 de resonancia (0, 2, 5, 8, 10) y grabar la salida con las 20 combinaciones posibles: 16 Hz y 0 resonancia, 16Hz y 2 resonancia, …, 4096 Hz y 10 resonancia. La frecuencia máxima (16 kHz) no la tenemos en cuenta porque equivale al bypass, el filtro no actúa.

Preguntas 2, 3 y 4: (sweep estática)

En este caso queremos ver cómo responde cada frecuencia al cambio de resonancia, para saber cómo se comporta el filtro con una señal de mismo nivel y cual es su rango. Para ello, la idea es coger cada frecuencia y grabar la salida con 11 valores de resonancia, que corresponden a los puntos marcados en el potenciómetro (por eso la señal de test es un tren de 11 señales, una para cada resonancia). Moveremos el potenciómetro de resonancia de menos a más, es decir, de izquierda a derecha. También habrá una pista (la *VCF\_allF\_0R\_sweepstat*) en la que queremos ver cómo cambia el comportamiento del filtro a medida que la frecuencia de corte va creciendo, sin ninguna resonancia.

Preguntas 5 y 6: (tonos puros)

Para conocer el SNR (Signal to Noise Ratio) y la THD (Total Harmonic Distorsion) solo nos hace falta conocer la salida del filtro sin que esté actue, es decir, el bypass, para poder compararlo con el señal original. Usamos un tono puro (4 tonos puros a diferentes frecuencias) ya que será más senzillo ver si hay distorsion o ruido que con una sweep.

Pregunta 9: (fuente de voltaje externa)

En este caso, usaremos una fuente de voltaje externa (la PROMAX FAC-363B) para modificar los parámetros. Concretamente, haremos dos grupos de grabaciones, en el primero, usaremos la fuente para cambiar la frecuencia de corte y en el segundo, la usaremos para cambiar la resonancia. En cada caso, haremos 4 grabaciones enviando 4 voltajes diferentes (1V, 4V, 7V y 10V) ya que el sintetizador puede recibir 1 Voltio/octava [10], es decir, un total de 11V.

Hay que tener en cuenta que es necesario enviar una señal de audio igualmente, ya que esta señal de voltaje únicamente sirve para modificar los valores de frecuencia y resonancia. Por si sola, no envia ninguna señal de audio. Haremos los dos tipos de grabaciones con la sweep estática y con la sweep variable, por lo que al final tendremos 16 grabaciones.

En total, contando todas las preguntas y todas las grabaciones que acabamos de explicar más los loopbacks (que explicaremos más adelante), el ruido producida por la electrónica del sintetizador y los cables (que también lo usaremos para el análisis) y las señales de test, tendremos un total de 67 pistas.

Hemos hablado de dos parámetros del VCF, eso es porque los otros dos (KB y ADS) son parámetros que añaden detalles al filtro pero que no son parte de su funcionamiento básico, sinó que sirven para controlar el teclado. Por tanto, ambos parámetros tendrán que estar a 0, para que no afecten al comportamiento del VCF. En otras palabras, el filtro solo responde a la frecuencia de corte y a la resonancia en esta frecuencia.

Como hemos mencionado, además de las salidas del filtro con los potenciómetros manipulados, previamente habrá que grabar como se comporta el VCF con un señal que le entra sin modificar los parámetros (con la configuración de la Figura 5) y otra señal al final de todo (el loopback), que contendrá la señal original más lo que aporta la tarjeta de sonido, para restarla al señal original y obtener el ruido o procesado que pueda añadir la tarjeta de sonido. Es esencial que el loopback se grabe cada día después de la calibración del equipo. También es importante grabar el loopback para cada señal de control, ya que puede variar en función de qué tipo de audio le entre a la tarjeta.

El loopback se graba usando un cable jack-jack que salga de la targeta y entre en ella de nuevo, tal como se ve en la Figura 6, en la cual el loopback es el cable negro. En esta misma imagen, el cable blanco es la señal de entrada del filtro, que sale de la tarjeta y el gris es la salida del filtro, que entra en la tarjeta (hay que tener en cuenta que las entradas de jack de la fila superior son las salidas y las entradas inferiores son las entradas de la tarjeta). Más adelante mostraremos un diagrama completo del sistema de grabación usado.



Figura 6: Loopback

Cabe destacar que hay una diferencia entre usar un cable balanceado o no balanceado a la hora de grabar el loopback. El primer dia de grabación descubrimos que con un cable no balanceado, había una pérdida de, aproximadamente, 6 dB 's entre la señal y el loopback, es decir entre el señal entrante y la salida. En cambio, con un cable balanceado, no había ninguna diferencia de niveles.

Dicho esto, hay que enfatizar que es necesario realizar TODAS las grabaciones con cable desbalanceado, simplemente teniendo en cuenta esta diferencia en dB 's, ya que el sinte es desbalanceado y todos los cables que usaremos lo son.

Para contextualizar, vamos a explicar brevemente la diferencia entre cables balanceados y no balanceados. En los primeros se envía el mismo señal pero duplicado, uno en fase y el otro en contrafase, de manera que en el punto de destino el señal se ve reforzado y el ruido se elimina (invirtiendo la fase de uno de los dos señales y sumándolos). Por eso no hay diferencia de niveles. El ruido se elimina ya que, si enviamos el señal A + ruido, y el -A + ruido y en el punto final invertimos una fase, tenemos (A + ruido) + [-(-A + ruido)] = A + ruido + A - ruido = 2A

En el segundo caso, se envía únicamente una señal, con lo cual el ruido no se elimina.[15] En la Figura 7, podemos ver la diferencia entre ambos cables (en estas grabaciones usamos cables jack). En la izquierda está el cable balanceado y a la derecha, el no balanceado.



Figura 7: Balanceado vs No Balanceado

Antes de entrar en el siguiente apartado, en la Figura 8 tenemos una imagen del sistema de grabación que hemos usado para todas las grabaciones excepto las correspondientes a la pregunta 9. Enviamos la señal de test desde el ordenador a la tarjeta de audio y desde ahí, la enviamos directamente al VCF. A continuación, la salida hace el recorrido inverso, vuelve a la tarjeta y de ahí al ordenador.



Figura 8: Montaje de grabación

4.2. Procesamiento de señales

Una vez tengamos todas las grabaciones hechas, habrá que realizar cierto procesamiento de señales para obtener la frecuencia de corte del filtro VCF y su resonancia. Esto es necesario porque queremos conocer el comportamiento real. El fabricante da unas pistas pero no siempre son ciertas y realizar este proceso es una forma de analizar el dispositivo y verificar estos valores (que corresponden a los valores por defecto sin calibrar del filtro). Es posible que el valor de la frecuencia sea cierto, pero que no conozcamos exactamente cómo se comporta la resonancia, sólo un valor que hay que mapear en magnitud y el ancho de banda de la resonancia. Con este procedimiento podemos conocer este comportamiento. Como explicamos en la sección 3, la resonancia añade una pequeña campana de gauss, en el valor de la frecuencia de corte, que ocupa un cierto ancho de banda y llega hasta un nivel máximo de magnitud (el factor de calidad Q).

Este procesamiento consiste en coger la inversa del filtro (que está generada por el plugin de Aurora y que es independiente del nivel, por lo que es igual tanto para la *sweep\_static* como para la *sweep\_variable*) y convolucionarla con el señal output (el que sale del VCF). El resultado de esta convolución es la respuesta impulsional (IR) de esa salida (esto implica que hay que hacerlo para todas las salidas y que será diferente para cada una de ellas). Calculando su transformada de Fourier, obtendremos una cierta magnitud con una cierta fase, en dominio frecuencial. Visualmente, a partir del gráfico de la magnitud podremos encontrar la frecuencia de corte del VCF, que será aquella en la que l’amplitud decaiga 3 dB. Con la resonancia, el proceso es muy parecido, ya que esta se ve, también, en el gráfico de la magnitud de la IR (ocurre en el valor de la frecuencia de corte).

El mencionado filtro inverso es, simplemente, la inversión temporal de la señal de entrada, con una pendiente de 6dB/octava [3]. No tiene nada que ver con el filtro VCF.

4.3. Función de transferencia

Volviendo a la función de transferencia, esta describe la relación entre la señal de salida y la de entrada en términos de voltaje, tal como se muestra en la Figura 7. Se puede descomponer en módulo y fase [17].

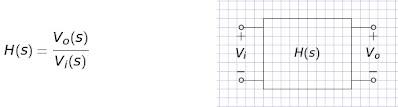


Figura 9: Función de transferencia de un filtro

En este caso, la función de transferencia de una frecuencia es la IR de la salida correspondiente dividida entre la IR del loopback, que es el señal de entrada puro.

4.4. Modelaje a partir de la respuesta impulsional

Como hemos mencionado en la sección 1, este tipo de modelaje consiste en hacer una convolución lineal entre la señal original y la respuesta impulsional (IR) del sistema. Esto permite emular sistemas no lineales, caracterizados por una distorsión armónica de varios órdenes.

Los efectos no lineales son una parte importante de la evaluación de la calidad del sonido percibido, y su eliminación provoca una degradación de la calidad sonora, en el sentido que pierde naturalidad por la falta de elementos ambientales, tales como reverb o ruido. Este método permite la caracterización completa de los comportamientos lineales y los no lineales de un sistema complejo (como el VCF). El resultado es un conjunto de respuestas impulsionales, siendo la primera la respuesta lineal tradicional, y las otras la respuesta en varios órdenes armónicos. Y este resultado es una buena aproximación al sonido real.[3]

La idea principal es emplear este conjunto de respuestas impulsionales en un proceso de convolución múltiple, capaz de reconstruir la modificación que le sucede a una señal que pasa a través del complejo sistema. En este caso, sería capaz de reconstruir el paso de un señal a través del VCF.

En [3] se explica que la función de transferencia del sistema (del VCF) se calcula usando una expansión de las series de Volterra, las cuales son muy parecidas a las series de Taylor pero teniendo en cuenta la memoria del señal, es decir, que calcula el output en un instante según el valor del input en todos los instantes anteriores, en vez de calcular el output de un instante con el input de ese instante concreto [18]. Si conocemos los “Volterra Kernels”, se puede reconstruir la salida independientemente de la entrada.

Los Volterra Kernels se pueden calcular a partir de las IR del VCF, que en este caso ya son conocidas, las obtenemos con el procesamiento de las salidas explicado en el apartado 4.2. Como hemos visto, es posible obtener la respuesta impulsional convolucionando la salida con el filtro inverso de la señal de control. Las respuestas de impulso obtenidas no son directamente los Volterra Kernels, pero estos se pueden calcular resolviendo un sistema de ecuaciones lineales. La salida medida se puede representar como la suma de la convolución lineal de las IR con la señal de entrada original y la correspondiente versión de frecuencia desplazada.

*[Cada capítol ha de començar en pàgina senar]*

**8. Bibliografia**

[1] (2009, Julio). Proyecto Sintetizador Documentacion Version 1.0 <http://profesores.elo.utfsm.cl/~agv/elo329/1s09/projects/AcevedoBarros/SinteAnalogo.html>

[2] D’Angelo, S. (2018, Noviembre). Lightweight virtual analog modeling. In Proceedings of the 22nd Colloquium on Music Informatics, Udine, Italy (pp. 20-23).

[3] 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

[4] Sheng, D., & Fazekas, G. (2019, Julio). A feature learning siamese model for intelligent control of the dynamic range compressor. In 2019 International Joint Conference on Neural Networks (IJCNN) (pp. 1-8). IEEE.

[5] Chris Meyer, (2016, November), Learning Modular

[**https://learningmodular.com/glossary/control-voltage/**](https://learningmodular.com/glossary/control-voltage/)

[6] (2021, Diciembre) RSF Kobol

<https://en.wikipedia.org/wiki/RSF_Kobol>

[7] (2018,Mayo), RSF Kobol Expander

<http://www.jarrography.free.fr/details_equipement_audio.php?id_equip=80>

[8] (2018), Reverb

<https://reverb.com/item/18757272-rsf-kobol-expander-i>

[9] (2018, Mayo), RSF Kobol Expander | Vintage Synth Explorer

<https://www.vintagesynth.com/misc/rsf_kobol.php>

[10] Yusynth, (February, 2017), RSF Kobol Manual

<http://yusynth.net/gear/Kobol_RSF_en.html>

[11] Steve Howell, (1983, Marzo), RSF Kobol Expander

<http://www.muzines.co.uk/articles/rsf-kobol-expander/5929>

[12] Gordon Reid, (October, 2010), XILS Lab PolyKB

<https://www.soundonsound.com/reviews/xils-lab-polykb>

[13] Roland Owner’s Manual, (2005), Edirol\_UA-101\_OM

<https://www.roland.com/global/support/by_product/ua-101/owners_manuals/39364883-4342-471f-9730-0bf7648914aa/>

[14] Angelo Farina, (Març, 2021), Aurora for Audacity

<http://www.angelofarina.it/Public/Aurora-for-Audacity/>

[15] Enric Giné, (Decembre, 2020), Apunts Equips i Sistemes d’Audio

[16] Wikipedia, (Octubre, 2016), Inverse Filter

<https://en.wikipedia.org/wiki/Inverse_filter>

[17] Universidad Miguel Hernandez, (...), Circuitos electrónicos analógicos

[**https://sites.google.com/a/goumh.umh.es/circuitos-electronicos-analogicos/transparencias/tema-2**](https://sites.google.com/a/goumh.umh.es/circuitos-electronicos-analogicos/transparencias/tema-2)

[18] Wikipedia, (February, 2022), Volterra Series

<https://en.wikipedia.org/wiki/Volterra_series>

*Cal documentar les fonts bibliogràfiques utilitzades amb un format de citació estàndard:*

[*http://guiesbibtic.upf.edu/tic/tfg*](http://guiesbibtic.upf.edu/tic/tfg) *(apartat* ***Presentació del treball*)**

*La Biblioteca de la UPF ofereix el gestor de bibliografies Mendeley, que us permet crear la vostra base de dades personal de referències bibliogràfiques en línia, importar referències automàticament des de diferents recursos d'informació, extreure llistes i generar bibliografies en diferents estils de citació, i incorporar les citacions i llistes de bibliografia als vostres documents de text*

[*Gestor de bibliografies Mendeley*](http://guiesbibtic.upf.edu/mendeley)

*Exemple de cita bibliogràfica [12 punts]*

Heery, M. “Organització de la biblioteca: Repàs d'estructures.” *Item: Revista de biblioteconomia i documentació 23.2 (1998): 8-15.*

Text de la cita bibliogràfica [12 punts]

*[Aquest apartat ha de començar en pàgina senar]*