Virtualización de un VCF analógico

## Descripción general

**Objetivo:** modelar el módulo de VCF (Voltatge Control Filter) de una de las unidades del sintetizador analógico [RSF Kobol Expander](https://www.vintagesynth.com/misc/rsf_kobol.php) que disponen en la Fundación Phonos.

Un dispositivo analógico genera una señal continua, no discretizada y es imperfecto sónicamente, 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. El caso es que son esas imperfecciones las que hacen único el sonido analógico y que hoy en día fabricantes de plugins buscan recuperar y modelar desesperadamente inundando el mercado de emulaciones analógicas. Hoy en día los dispositivos analógicos siguen consumiéndose y van a la alza en un entorno profesional y una prueba de ello es que en la mayoría de estudios apuestan por un sistema híbrido donde conviven el mundo analógico y digital.

Por esos motivos, el objetivo de este proyecto es muestrear y modelar el procesador analógico del VCF. Teóricamente un VCF se basa en el siguiente diagrama de bloques:

* Existe una señal de entrada que excita una envolvente temporal.
* 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.

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

* Virtual Analog Modelling [[1]](https://dangelo.audio/docs/lightweightva.pdf)

Consiste en la simulación de circuitos eléctricos para aplicaciones relacionadas con el sonido en tiempo real.

* Impulse Response Modelling [[2]](https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5446595&casa_token=RTH_bU7fB5kAAAAA:VZLjJPJO91sBzQeeHwU5BlofUD95HiLXusGRocxi4KPtVqNaGFPKxON8nit3csV0K2Q19hjICDg&tag=1) (pel [2] cal registrar-se) [[3]](http://160.78.24.2/Public/AES-110/154-aes110.PDF)

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

* Deep Neural Network Modelling [[4]](https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8851950&casa_token=6DAFIBtHAXsAAAAA:B3rVoKCq_sBRYsXOEz8VFsRQ22dgnIV4PK8TWRUI_OGQ7cGYizG1DwILDVD1rThxF2R4JxJIcxk) (pel [4] cal registrar-se)

En la mayoría de los casos (excepto en el primero), se precisa una base de datos con sonidos o señales de test para extraer resultados/conclusiones y poder modelar modelar. La calidad de la 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 test y registrar la salida del VCF. De esta manera tenemos la opción de analizar la respuesta en frecuencia del LPF 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 [magenta] [christian steinmetz].

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 un cambio en análisis de las muestras de 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/respuesta (lineal, logarítmico, exponencial).



El VCF dispone de dos parámetros manuales:

* VCF FREQUENCY: define la frecuencia de cortes
* RESONANCE: define la campana de resonancia que se genera en la frecuencia de corte para dar un cierto carácter al sonido generado.

Y de dos controles que modifican el comportamiento del VCF:

* KEYB. CONTROL
* ADS CONTROL: no es una completa ADSR

Para modelar de forma precisa, es necesario conocer el rango en frecuencia del parámetro VCF FREQUENCY, si su escala de forma logarítmica o lineal en frecuencia. En la serigrafía se indica de 16Hz a 16KHz, pero hay que comprobarlo.

Con el parámetro de resonancia el objetivo es un poco el mismo, conocer el ancho de banda de la campana para valores discretos de este parámetro.

Por otro lado el filtro se puede controlar via 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 test a estas entradas y ver como responde el filtro.

Por otro lado también es importante conocer las especificaciones técnicas de la unidad proporcionadas por el fabricante (si las hay) y analizar aspectos técnicos como la THD, la SNR 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.

Finalmente, la grabación de muestras se hará controlando el tiempo desde el encendido de la unidad, ya que este seguramente afectará el sonido. Podríamos hacer: inmediato, pasada 1 hora, tras 2 horas, tras 4, tras 8, tras 24…

## Secciones del proyecto

Este proyecto se puede descomponer en cuatro secciones:

1. Revisión de la documentación relacionada con la unidad (si está disponible).
   * Es necesario conocer detalles técnicos de cada control y su implementación. El fabricante suele dar alguna pista en el [manual](http://www.univertron.com/SAS/manuals/RSF/Kobol%20Expander%20Schematics/) o en el [manual de servicio](https://elektrotanya.com/rsf_kobol_expander-ii_sch.pdf/download.html).
   * Anuncios, catálogos o flyers de la época pueden aportar alguna información.
   * Emulaciones en versión plugin
     + Native Instruments - Kontakt
   * Artículos relacionados.
2. Grabación de muestras
   * 48k, 24 bit en formato WAV
   * Para analizar el parámetro manual y los controles por CV podemos utilizar *sweeps logarítmicas* a la entrada.
   * En el control manual, tendremos que registrar la señal una a una tras modificar el control. En el control por CV podemos enviar una señal de amplitud estable que conozcamos para ver cómo modula la frecuencia para cada nivel de amplitud.
     + En este sentido se puede crear una señal sinusoidal con una rampa lineal en amplitud para ver cómo evoluciona la frecuencia del filtro para cada nivel de amplitud.
   * Con estos registros podremos estimar la función de transferencia en distintas configuraciones de cada control.
3. Análisis de las muestras
   * Analísis Global
     + Total Harmonic Distortion
     + Signal-To-Noise Ratio
       - Análisis de la envolvente frecuencial de ruido de fondo.
     + Rango Dinámico
   * Análisis de cada parámetro
     + VCF Frecuencia
       - Manual
       - CV
     + Resonancia
       - Manual
       - CV
     + ADS control?
4. Implementación
   * Implementación del prototipo
     + Implementación del prototipo de cada control
     + Integración de los prototipos de cada módulo
   * Implementación de un VST en JUCE

Preguntas que nos surgen:

* Podremos modelar con DNNs el VCF utilizando la base de datos generada?

## Fites - Milestones - Hitos

* Búsqueda de información técnica de la documentación del fabricante.
  + Puede ser interesante y útil instalarse la demo de la única virtualización que existe de un modelo “hermano” de nuestros módulos, y familiarizarse con el VCF y su control (desactivando todo lo que se pueda del resto de controles y dispositivos): <https://www.xils-lab.com/products/polykb-iii-&-ii-p-159.html>
  + Hay una buena reseña en Sound on Sound que también puede servir como introducción: <https://www.soundonsound.com/reviews/xils-lab-polykb>
* Extracción de información de valor para el proyecto.
* Diferenciar y listar controles de parámetros.
  + Tenemos 2 parámetros y 2 controles.
* Listar y definir el plan de cada análisis paso por paso.
  + THD del modulo
  + SNR del modulo
  + Ruido de fondo del módulo
  + Tipología del filtro
  + Rango en frecuencia de corte
  + Desviación en frecuencia de corte
  + Orden del filtro
  + Respuesta del control de frecuencia: lineal, logarítmico, exponencial.
  + Rango de resonancia en factor Q (fc / ( f1 - f2) = fc / BW)
  + Respuesta de resonancia
* Creación de una base de datos con muestras para cada análisis.
  + Importante definir una resolución (pasos o discretización) y un método de calibración
    - -18dBRMS = 0VU = +4dBu = 83dBC con ruido rosa
    - Así nos aseguramos que siempre estamos en las mismas condiciones de nivel.
    - Utilizar siempre la misma señal de calibración.
  + Definición de la estructura, taxonomía, nomenclatura y etiquetado de la base de datos.
* Analisis global
  + Estimación la THD con el sistema de muestreo calibrado
  + Estimación la SNR
  + Estimación del ruido de fondo
* Análisis del control frecuencia
  + Rango y respuesta: Se puede realizar con una sweep logarítmica.
  + Estimar la función de transferencia de cada sweep.
  + Previamente definir los pasos discretos del rango continuo del control.
* Análisis de control de resonancia
  + Factor Q o ancho de banda para cada frecuencia central
  + Amplitud y loudness para cada nivel de resonancia
* Análisis de los controles por Control Voltage (CV)
  + Frecuencia de corte para diferentes niveles de amplitud de la señal de control.
    - Onda sinusoidal a diferentes niveles.
  + Variaciones en función de la frecuencia?
    - Responde igual a cualquier frecuencia en la onda sinusoidal.
* Comparativa con la respuesta del plugin.
* Implementar prototipo en Python, MATLAB o Pure Data
  + Handcrafted
    - Implementación del control de VCF frecuencia
    - Implementación del control de resonancia
    - Implementación del control de VCF frecuencia (CV)
    - Implementación del control de resonancia (CV)
* Implementación
  + en un plugin VST
  + un modelo en con DNNs en magenta [[6]](https://github.com/magenta/ddsp)
  + Un efecto neural controlable [[5]](https://neuripscreativityworkshop.github.io/2021/accepted/ncw_82.pdf)

## Puntos importantes

* La calidad del muestreo y su resolución definirán la calidad del análisis y la implementación del modelo.
* Vale la pena documentarse sobre el estado del arte de sintetizadores analógicos. Puede ser un capítulo (o parte) del documento a entregar.
* El análisis de cada control puede ser una sección del capítulo de análisis en la tesis.
  + Y los resultados y la comparativa entre osciladores otro.
* En caso que se llegue a implementar el prototipo, sería otro capítulo más.

## Referencias bibliográficas

[1] D’Angelo, S. (2018, November). Lightweight virtual analog modeling. In Proceedings of the 22nd Colloquium on Music Informatics, Udine, Italy (pp. 20-23). [[PDF]](https://dangelo.audio/docs/lightweightva.pdf)

[2] Bank, B., Zambon, S., & Fontana, F. (2010). A modal-based real-time piano synthesizer. IEEE transactions on audio, speech, and language processing, 18(4), 809-821. [[PDF]](https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5446595&casa_token=RTH_bU7fB5kAAAAA:VZLjJPJO91sBzQeeHwU5BlofUD95HiLXusGRocxi4KPtVqNaGFPKxON8nit3csV0K2Q19hjICDg&tag=1)

[3] Armelloni, E., Bellini, A., & Farina, A. (2001, May). Not-linear convolution: A new approach for the auralization of distorting systems. In Audio Engineering Society Convention 110. Audio Engineering Society. [[PDF]](http://160.78.24.2/Public/AES-110/154-aes110.PDF)

[4] Sheng, D., & Fazekas, G. (2019, July). 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. [[PDF]](https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=8851950&casa_token=6DAFIBtHAXsAAAAA:B3rVoKCq_sBRYsXOEz8VFsRQ22dgnIV4PK8TWRUI_OGQ7cGYizG1DwILDVD1rThxF2R4JxJIcxk)

[5] Steinmetz, C. J., & Reiss, J. D. (2021). Steerable discovery of neural audio effects. arXiv preprint arXiv:2112.02926. [[PDF]](https://neuripscreativityworkshop.github.io/2021/accepted/ncw_82.pdf) [[repo]](https://github.com/csteinmetz1/steerable-nafx)

[6] Engel, J., Hantrakul, L., Gu, C., & Roberts, A. (2020). DDSP: Differentiable digital signal processing. arXiv preprint arXiv:2001.04643. [[PDF]](https://arxiv.org/pdf/2001.04643.pdf) [[repo]](https://github.com/magenta/ddsp)