Proyecto de Computación I: Informe

Análisis armónico de señales musicales mediante Matlab

Teresa Pelinski

Abstract

Este programa analiza muestras de sonido mediante la transformada de Fourier. Por una parte extrae la transformada general, de toda la muestra, a partir de la cual saca la frecuencia fundamental del sonido que está sonando (la nota). Por otro lado, aplica la transformada a conjuntos de *samples* para ver cómo evolucionan los parciales a lo largo del tiempo. Finalmente, reconstruye los sonidos aplicando la transformación inversa. También recopila algunos ejemplos de qué ocurre si modificamos artificialmente la distribución de los armónicos.

1. Introducción

fourier.m tiene como objetivo estudiar la distribución de armónicos[1] en muestras de audio de notas de diferentes instrumentos.
Los armónicos son primordiales en el estudio del sonido, pues su distribución en el espectro de frecuencias determina si nuestro oído escucha un tono definido o ruido. El ruido se compone de armónicos distribuidos a lo largo de todo el espectro de frecuencias, sin orden aparente, mientras que un tono se genera con armónicos distribuidos a múltiplos quasienteros de una frecuencia fundamental. Partiendo de esta idea, el programa $fourier.m$ utiliza la transformada de Fourier para descomponer señales de notas ('discretas', no está pensado para señales polifónicas) [2] e identificar cuál es la nota sonante en cuestión. Además, muestra gráficos de cómo se distribuyen dichos armónicos en el espectro de frecuencias y cómo varía su intensidad en el tiempo. Finalmente, propone 'jugar', editando las intensidades de estos armónicos para ver cómo varía el sonido.

1.1. Sobre la idea inicial

La primera idea del programa fue que fuera capaz de reconocer el instrumento del que procedía la nota o al menos la familia del mismo (viento, cuerda, etc.). Esto resultó ser un problema mucho más complicado que un análisis espectral básico. Comentaré en el apartado 5 algunas ideas sobre cómo podría hacerse.

2. Teoría

> Cualquier pulso periódico puede representarse matemáticamente por una combinación de senos y/o cosenos por el proceso conocido como descomposición de Fourier.[1]

Si tenemos una muestra de sonido, podemos aplicar el proceso inverso: la transformada de Fourier. De esta manera encontramos que su energía espectral está concentrada en frecuencias discretas f_i. Si el sonido es quasiperiódico (como ocurre en los instrumentos reales), los armónicos serán aproximadamente múltiplos de la frecuencia fundamental f_i, i.e. $f_i(t) \simeq i f_0(t)$. La amplitud de cada parcial no es constante y su variación en el tiempo es de capital importancia para caracterizar su timbre. [2] (El timbre es la propiedad del sonido que permite, para una misma nota, distinguir entre diferentes fuentes de sonido.)

Cabe preguntarse si las amplitudes de los parciales siguen algún tipo de patrón. Lo cierto es que las propiedades físicas y geométricas de la fuente de sonido (del instrumento), actúan como un filtro, amoldando las amplitudes de las componentes individuales de la frecuencia. Dado que el resonador (como

[1] Armónicos y parciales son términos equivalentes.

[2] He comprobado que el programa también funciona para reconocer acordes en las llamadas 'posiciones cerradas' en la guitarra. Las posiciones cerradas tienen como nota más grave la nota fundamental del acorde, luego es probable que el reconocimiento se haga por este motivo, pero tampoco está claro.

puede ser la caja de una guitarra española) está acoplado a la fuente, el cuerpo del instrumento vibra de acuerdo a los diferentes modos de vibración, pero con zonas del espectro de freceuencia activadas por la vibración de la fuente. Qué frecuencias y en qué medida son cuestiones que vienen determinadas por las propiedades físicas y geométricas del resonador. Las zonas de frecuencia ampliadas son llamadas *formants* y no dependen de la frecuencia (fundamental) a la que vibra la fuente[3]. La función que 'amolda' las amplitudes de cada frecuencia y modela los *formants* suele llamarse *spectral envelope*. [3]

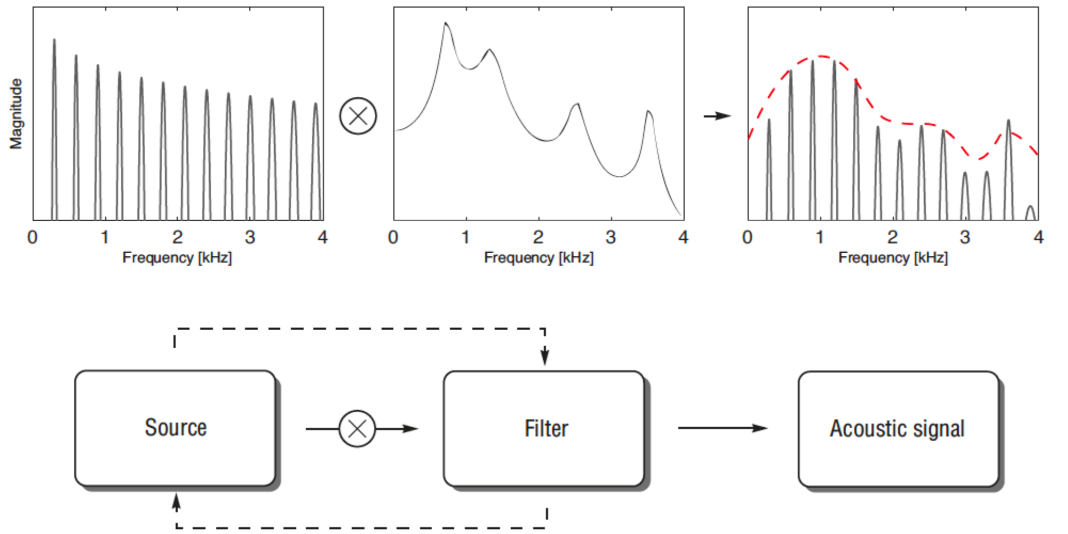

Figure 3.1: Source-filter representation of instrumental sound production. The process can be regarded as a multiplication of the excitation spectrum with the resonator's transfer function. The coupling of source and filter causes an interaction of the two components, depicted as forward and feedback loops. The dashed line in the upper right plot denotes the resulting spectral envelope of the sound.

Figura 1: Figura extraída de [3]

El objetivo de la representación en tiempos cortos de la transformada de Fourier y su plot temporal es precisamente apreciar visualmente en qué medida varían los parciales en el tiempo. Hay que tener en cuenta que en una muestra real, influyen el ataque, el sustain y el release. En el apartado **??** puede verse un gráfico de esto.

2.1. Sobre el orden de los armónicos

Si particularizamos para una cuerda sujeta por ambos extremos, el hecho de que los armónicos estén ordenados, sin tener en cuenta las amplitudes, (como puede verse al hacer funcionar el programa) de tal manera que el segundo es la octava del primero, el tercero es la quinta del segundo, el cuarto es la cuarta del tercero, etc., se corresponde directamente con cómo varía la longitud de onda, y con ella la frecuencia cuando la cuerda vibra en sus diferentes modos armónicos.
En el segundo armónico, la longitud de onda, como puede verse en la figura[4], se reduce a la mitad (1/2), con lo que su frecuencia se dobla. A esto es lo que llamamos octava y se denota con 1:2. El segundo armónico tiene una longitud de onda 2/3 veces la longitud de onda fundamental, por ello se denota con 2:3. El siguiente armónico, corresponde al cuarta (3:4), etc. [5]. Las frecuencias pueden calcularse

[3]Esto hace que el timbre también dependa en cierta medida del tono. [3]

[4]Imagen extraída del siguiente sitio web.

[5]Este es el motivo de que percibamos la octava, la quinta y la cuarta como consonantes, pues siempre que tocamos un Do en un piano, está sonando, al menos, su segundo (Sol) y tercer armónico (Fa). Luego si tocamos un Do y un Sol en el teclado necesariamente sonará bien (si el piano está afinado), pues al picar la cuerda de Do, se está haciendo sonar también, aunque quizás más débilmente, un Sol.

sistemáticamente usando:

$$v = \sqrt{\frac{T}{\mu}} \qquad\qquad f_n = \frac{n}{2L}\sqrt{\frac{T}{\mu}}$$

donde v es la velocidad de propagación de la onda en la cuerda, T la tensión de ésta, μ su densidad lineal y L su longitud. Los índices $n = 1, 2, 3, ...$ corresponden al modo de vibración (o armónico) de la cuerda.
Este razonamiento para una cuerda sujeta por dos puntos fijos puede extenderse a todos los instrumentos, teniendo en cuenta, claro está, las particularidades de aquello que sea que genera la onda de sonido. En ciertos instrumentos se da el fenómeno de que solo suenan ciertos conjuntos de armónicos (por ejemplo, los pares).

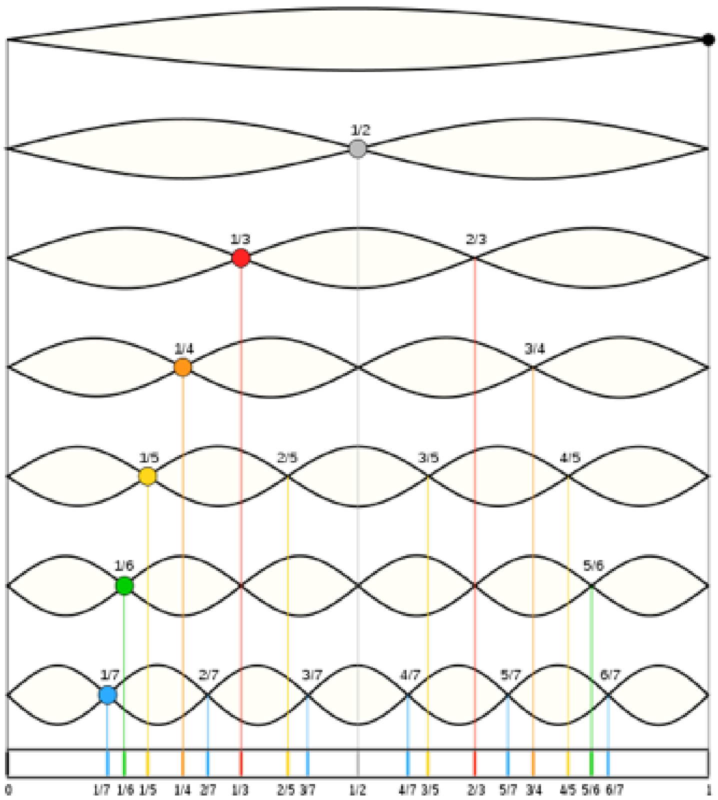

Figura 2: Modos de vibración de una cuerda sujeta por ambos extremos.

El propósito de esta explicación es justificar cómo sé que el funcionamiento de mi programa es el correcto. Hablaré de esto en el apartado 4.2.

3. Funcionamiento

3.1. Diagrama de flujo

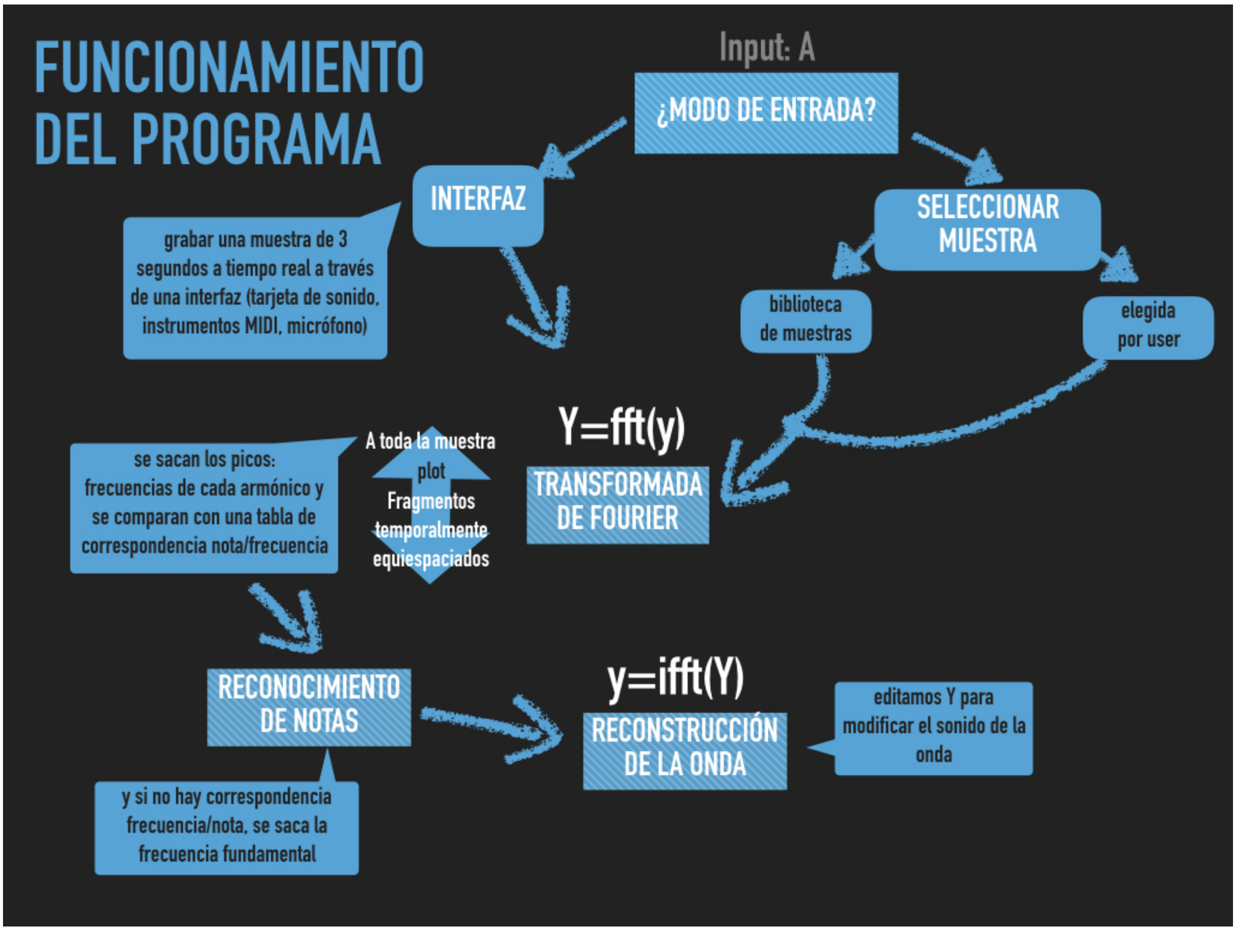

Figura 3: Funcionamiento del programa

3.2. Resumen del funcionamiento

El programa recibe una muestra de sonido, que puede recogerse o bien por una interfaz, de manera instantánea, o por un archivo previamente grabado preferiblemente en formato .wav. El programa también contiene una biblioteca de muestras[6] para que el usuario no tenga que aportar su propio archivo para hacer funcionar el programa.

Una vez leída la muestra, se aplica la transformada de Fourier tanto a la muestra entera como a secciones de la misma equiespaciadas en el tiempo. Esto permite sacar una transformada de Fourier general para obtener los armónicos, y otra en tres dimensiones en la cual se puede ver su evolución a través del tiempo. Dado que en el sonido influyen características tales como el ataque, el sustain, y el decaimiento, este gráfico es muy útil para ver cómo se comportan los armónicos en cada etapa.

De la transformada de Fourier se obtiene la frecuencia fundamental y el programa compara esta frecuencia con una base de datos que recoge las equivalencias frecuencia–notas. Si hay correspondencia, saca la nota por pantalla, si no la hay, saca la frecuencia fundamental.

[6]Todas las muestras utilizadas por el programa han sido descargadas de freesound.org bajo la licencia de Creative Commons.

Podemos preguntarnos qué ocurre acústicamente si modificamos las amplitudes relativas entre armónicos. La parte final del programa recoge un par de ejemplos de aplicar funciones a las amplitudes de los armónicos y reproducir el sonido resultante al aplicar la función inversa a la transformada de Fourier.[7]

3.3. Explicación del código

El código está explicado en los comentarios del script. Tan solo quería señalar algunos detalles:

- La versión que he usado es MATLAB2016b. Encontré que tenía un bug con el comando `play`. El uso de la función no es demasiado importante, simplemente reproduce la muestra de sonido si no ha sido grabada a través la interfaz. Esto puede solucionarse descargando el siguiente paquete o simplemente haciendo el 'arreglo' de cambiar la línea 119:

  ```
  p=audioplayer(y,fs);
  ```

  por

  ```
  p=audioplayer(y(1:0.99*end),fs);
  ```

- No he marcado ningún tiempo máximo para las muestras pero lo ideal es no duren más de 7 segundos para que el programa sea eficiente.
- La correspondencia entre notas y frecuencias es relativa a la afinación 440[8]. La tabla de correspondencias fue obtenida del siguiente sitio web: MTU: Physics of Music.
- Las amplitudes entre armónicos han sido normalizadas (es decir, son relativas a la amplitud del armónico máximo). El propósito de esto es que factores como el volumen de la muestra no influyan en el estudio de los parciales y poder observar mejor las relaciones entre armónicos.
- El algoritmo para encontrar el armónico fundamental consiste en buscar aquellas frecuencias cuyo pico sea mayor de 0.3 (normalizado). No se ha hecho encontrando aquel que tiene la frecuencia máxima pues no tiene por qué ser el primero.
- Los parámetros de tolerancia para la correspondencia de notas (véase la línea 156) fueron ajustados experimentalmente. Es decir, fui probando con diferentes notas en la guitarra hasta que el programa acertó todos los ensayos.

[7] Existen muchos métodos para sintetizar sonido a partir de datos numéricos. Comentaré algunos brevemente en la sección 5. Quería añadir que es interesante tratar de reproducir la muestra de sonido quitándole el primer o el segundo armónico. Lo esperable es que nuestro oído oiga un sonido más bajo, pero al parecer esto no ocurre así. En [4], se mencionan estudios sobre la *altura residual*, la altura percibida correctamente en ausencia de la frecuencia fundamental. Se ha demostrado que *percibimos la altura corresponidente a la diferencia de frecuncias entre armónicos sucesivos, aún cuando falten el fundamental y varios de los armónicos graves.* Al parecer esto ocurre en las radios de transistores (antiguas), pues tienen altavoces tan pequeños que las notas graves son demasiado déiles para ser audibles, y sin embargo se oyen a alturas correctas. Por esto es, cuanto menos, curioso, escuchar cómo varía el sonido al mutilar el primer armónico.

[8] Esto significa que al la central del piano, La_4 se le asocia la frecuencia 440Hz.

4. Resultados

4.1. Ejemplo de representaciones gráficas

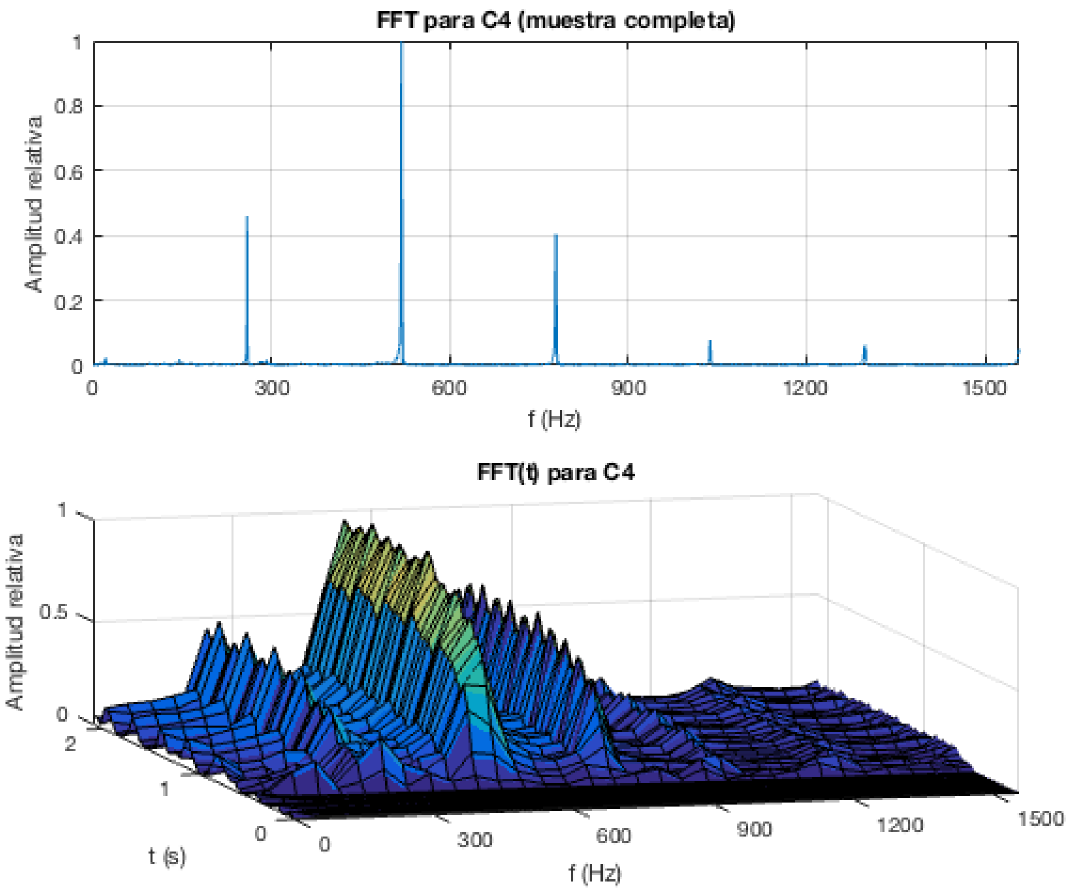

Figura 4: Gráficos generados por *foruier.m* para una muestra de un cello (incluída en la biblioteca de muestras). Pueden diferenciarse las partes de ataque y sustain.

4.2. Hitos

Transformada de Fourier Para comprobar que la transformada de Fourier es correcta, hay que asegurarse de que, conocida la nota a la que suena la muestra, los parciales se correspondan con múltiplos quasienteros de la frecuencia fundamental. En la práctica, basta con cerciorarse de que el segundo armónico se corresponde con la octava (por el motivo que he explicado en el apartado 2.1) y el tercero con la quinta. Se ha obserado que se cumple para la guitarra y para las muestras de la biblioteca de muestras.

Identifiación de notas Las muestras utilizadas venían identificadas 'de serie' con el nombre de la nota, luego fue fácil comprobar que el programa diera un resultado correcto. En cuanto a las notas de la guitarra, los parámetros se ajustaron especialmente para que el programa acertara todos los intentos (véase 3.3) para notas distribuidas a lo largo de todo el mástil.

Plots Mientras que el plot para la transformada de Fourier de toda la muestra fue sencillo, para la transformada en el tiempo no fue tan simple. Ajustar las dimensiones fue problemático: según la frecuencia fundamental salían gráficas con puntos bien distribuidos (número de puntos en t – eje X similar al número de puntos en f – eje Y) o mal distribuidos (rectángulos muy estirados en la vista tiempo frente a frecuencia – plano XY). Para arreglar esto busqué un parámetro que cambiara el número de transformadas de Fourier según la frecuencia fundamental (véase las líneas 203–207). Es funcional pero es cierto que con este tipo de 'apaños' el programa pierde generalidad.

Multiopción y ventanas emergentes La primera verisón del programa interactuaba con user por la consola de Matlab. Me parecía poco dinámico, dado que como se muestra en el diagrama de flujo hay que elegir entre opciones o incluso entre una lista (en el caso de que se elija la opción de acceder a la biblioteca de muestras). Por eso, edité el programa utlizando funciones como `questdlg`, `listdlg` o `msgbox` y como resultado user no tiene que interactuar en ningún momento con la consola, puede hacerlo perfectamente a través de las ventanas emergentes. Además están programadas para que si se cierra la ventana esta vuelva a abrirse automáticamente y el programa no muestre error.

5. Ampliación: Síntesis de sonido

fourier.m está centrado principalmente en la forma del espectro de frecuencias de un sonido. En la última parte, se deja entreveer las posiblidades que ofrece la síntesis de sonido, 'montar' un sonido matemáticamente mediante métodos computacionales.
Algo interesante sería obtener funciones que aproximaran los formants de cada instrumento. Para ello harían falta muchas notas del mismo instrumento (por no decir que todas, para ubicar bien las zonas 'afectadas' por los formants). Haciéndolo sobre diversos instrumentos podría obtenerse una base de datos de *spectral envelopes* para esos diversos instrumentos. Así, seríamos capaces de modificar matemáticamente ondas para que sonaran como uno u otro instrumento. Cabría investigar cómo funcionan los instrumentos MIDI y cómo se simulan los sonidos para que sean tan parecidos a los instrumentos reales en programas del estilo ProTools, LogicPro, Garageband, etc.

El modelo de análisis de sonido que hemos utilizado en este trabajo es un modelo espectral, pues está centrado en el espectro de frecuencias que se obtiene de la transformada de Fourier para una muestra. Otros modelos espectrales trabajan con ruido blanco (*white noise*) o tratan los resonadores como filtros. Estos últimos, de hecho, trabajan con las zonas de resonancia del filtro, lo que sería equivalente a los *formants* mencionados en la sección 2 (la figura 3 propone, de hecho, este enfoque). [2]
Existen otros muchos modelos. El *Spectral Modeling Synthesis (SMS)* es un modelo híbrido que considera el sonido formado por una parte sinusoidal y una parte residual. La primera corresponde a los modos de vibración del sistema, y la segunda, la parte residual, es modelada como "la convolución del ruido blanco con un *time-varying shaping filter*.[2] Este programa podría ser un punto de partida para la síntesis de sonido utilizando el método *SMS*.

Referencias

[1] Thomas D. Rossing (Ed.): *Springer Handbook of Acoustics.*, 2nd ed. Standford: Springer, 2014.

[2] G. Assayag, H.G. Feichtinger, J.F. Rodrigues (Eds.): *Mathematics and Music. A Diderot Mathematical Forum.*, 1st ed. Berlín: Springer, 2002.

[3] Ferdinand Fuhrmann: *Automatic musical instrument recognition from polyphonic music audio signals* Barcelona: Tesis doctoral. Universidad Pompeu Fabra, 2012.

[4] John R. Pierce: *Los sonidos de la música.*, 1ªed. Barcelona: Prensa Científica. Editorial Labor, 1985