



**Universidad de Jaén**  
Escuela Politécnica Superior de Linares

Trabajo Fin de Grado

**SEPARACIÓN DE FUENTES  
SONORAS APLICADA A LA  
EXTRACCIÓN DE LA PISTA  
RÍTMICA EN SEÑALES  
MUSICALES**

**Alumno:** David González Gallardo

**Tutor:** Francisco Jesús Cañadas Quesada  
Pedro Vera Candeas

**Depto.:** Ingeniería de Telecomunicación

**Septiembre, 2016**

# Índice

|                                                 |    |
|-------------------------------------------------|----|
| Índice.....                                     | 1  |
| Glosario de términos.....                       | 3  |
| 1. Introducción .....                           | 4  |
| 1.1. Motivación.....                            | 4  |
| 1.2. Introducción a la teoría musical .....     | 6  |
| 1.3. Señales sonoras.....                       | 10 |
| 1.3.1. Caracterización del sonido .....         | 10 |
| 1.3.2. Tipos de señales sonoras.....            | 14 |
| 1.4. Sonidos rítmicos .....                     | 18 |
| 1.5. Sonidos no rítmicos .....                  | 20 |
| 2. Objetivos .....                              | 22 |
| 3. Estado del arte .....                        | 24 |
| 3.1. Separación de fuentes acústicas .....      | 24 |
| 3.2. Acondicionamiento de la señal .....        | 27 |
| 3.3. Extracción de la pista rítmica .....       | 32 |
| 3.4. NMF .....                                  | 41 |
| 3.5. NMPCF .....                                | 55 |
| 3.6. Máscaras .....                             | 57 |
| 4. Implementación .....                         | 62 |
| 4.1. Hipótesis inicial.....                     | 62 |
| 4.2. NMF modificado.....                        | 69 |
| 4.2.1. REPET + NMF modificado .....             | 73 |
| 4.2.2. ELLIS + NMF modificado .....             | 75 |
| 4.3. Selección de bases con menor varianza..... | 79 |
| 5. Evaluación .....                             | 84 |
| 5.1. Bases de datos.....                        | 84 |
| 5.1.1. Base de datos de optimización.....       | 85 |
| 5.1.2. Base de datos de testeo .....            | 86 |
| 5.2. Métricas.....                              | 87 |
| 5.3. Optimización .....                         | 90 |
| 5.4. Resultados .....                           | 96 |
| 6. Conclusión y líneas futuras.....             | 98 |

|    |                        |     |
|----|------------------------|-----|
| 7. | Bibliografía .....     | 100 |
| 8. | Manual de usuario..... | 103 |

# Glosario de términos

|              |                                                                                                  |
|--------------|--------------------------------------------------------------------------------------------------|
| <b>NMF</b>   | Non-negative Matrix Factorization (Factorización de Matrices No negativas).                      |
| <b>BSS</b>   | Blind Source Separation (Separación de fuentes ciega).                                           |
| <b>SSS</b>   | Sound Source Separation (Separación de fuentes acústicas).                                       |
| <b>NMPCF</b> | Non-negative Matrix Partial Co-Factorization (Cofactorización Parcial de matrices No negativas). |
| <b>ICA</b>   | Independent Component Analysis (Análisis de Componentes Independientes).                         |
| <b>STFT</b>  | Short-Time Fourier Transform (Transformada de Fourier en Tiempo Corto).                          |
| <b>ISTFT</b> | Inverse STFT (STFT Inversa).                                                                     |
| <b>FFT</b>   | Fast Fourier Transform (Transformada Rápida de Fourier).                                         |
| <b>REPET</b> | REpeating Pattern Extraction Technique.                                                          |
| <b>SDR</b>   | Source to Distortion Ratio.                                                                      |
| <b>SIR</b>   | Source to Interference Ratio.                                                                    |
| <b>SAR</b>   | Source to Artifacts Ratio.                                                                       |
| <b>SNR</b>   | Source to Noise Ratio.                                                                           |
| <b>ISI</b>   | Inter-Symbol Interference (Interferencia Entre Símbolos).                                        |
| <b>MIR</b>   | Music Information Retrieval.                                                                     |

# Capítulo 1

## 1. Introducción

### 1.1. Motivación

El oído humano es, por sí solo, capaz de discernir sonidos concretos dentro de toda la escena auditiva que se encuentra a su alrededor de manera sencilla, a diferencia de las máquinas. Esta particular característica resulta de gran utilidad para seleccionar un determinado sonido y seguirlo mentalmente a lo largo del tiempo. Así, por ejemplo, durante un concierto, un espectador tiene la capacidad de identificar el sonido procedente de un determinado instrumento y concentrarse en una determinada parte de la obra.

A día de hoy se está tratando de automatizar este proceso mediante lo que se conoce como MIR (Music Information Retrieval) utilizando separación de fuentes acústicas o sonoras. Aunque se han conseguido enormes progresos, aún no se ha alcanzado el grado de calidad suficiente como para considerar solventado este problema.

La separación de fuentes acústicas nos permite, partiendo de una señal polifónica y multitímbrica, realizar el procedimiento que el oído humano es capaz de llevar a cabo pero mediante el procesado digital de señales. En los últimos años, este problema ha sido abordado desde el punto de vista de la descomposición de señales para su posterior clasificación de componentes y, finalmente, tratar de resintetizar aquellas señales que resulten de interés. Así, buena parte de los algoritmos que a día de hoy buscan una solución a este problema, se basan en modelos de señales consistentes en la descomposición de matrices que nos permiten obtener una serie de determinadas funciones.

El propósito de este trabajo es, como su propio nombre indica, la separación de fuentes sonoras aplicado a la extracción de la pista rítmica en señales musicales. En concreto el tipo de señales con el que se trabajará serán señales monoaurales, siendo este escenario el más complejo posible dado que el hecho de trabajar única y exclusivamente con un solo canal, conlleva la pérdida de cualquier tipo de posible información espacial aportada a la hora de realizar la grabación, información que sí se tiene en señales estéreo o multicanal. A pesar de trabajar sobre este tipo de señales, si nos encontrásemos en un escenario con señales multicanal, los resultados

obtenidos por este documento son completamente compatibles con otras técnicas y utilizadas en conjunto pueden dar lugar a una considerable mejoría.

En concreto, el método abordado en este documento se basa en la segmentación de nuestra señal en el dominio temporal y en el algoritmo conocido como *Factorización de matrices no negativas (Non-negative Matrix Factorization, NMF)*, aplicando una serie de modificaciones para aprovechar los beneficios obtenidos de dicha segmentación. Gracias a NMF se conseguirán unas funciones base que permitirán la reconstrucción de la pista rítmica de las señales analizadas. Es importante aclarar que este documento trata de implementar un sistema completamente ciego (*Blind Source Separation, BSS*), es decir, para la separación de fuentes no será necesario ningún procedimiento de entrenamiento ni tampoco utilizar otra señal que aquella que queramos descomponer. Esto conlleva una enorme ventaja con respecto a otros algoritmos que si se valen de la ayuda de ciertos patrones espectrales o de ciertas señales para realizar un entrenamiento previo a la separación.

### ***Hipótesis:***

*La pista rítmica de las señales sonoras consiste en aquella parte de la señal que se repite de forma regular a lo largo del tiempo. Trabajando en el ámbito de la música, ciertos instrumentos, como percusivos y línea de bajos suelen marcar el patrón rítmico seguido por la melodía aunque en la música más moderna también se introducen ciertos efectos y demás. Conocido esto, el algoritmo propuesto trata de obtener la pista rítmica a través de la obtención de las funciones base de dichos instrumentos.*

## 1.2. Introducción a la teoría musical

Las señales musicales que el ser humano percibe están compuestas por sonidos. La RAE define el sonido como la “*Sensación producida en el órgano del oído por el movimiento vibratorio de los cuerpos, transmitido por un medio elástico como el aire*”.

El modo más común de representar los diferentes sonidos es mediante el uso de partituras (figura 1.1). Estas se forman mediante unos determinados símbolos acompañados de otro tipo de información que nos indica, por ejemplo, la intensidad con la que debe interpretarse una determinada nota, la velocidad a la que debe ser tocada una obra o fragmento de la misma y muchos otros tipos de matices.

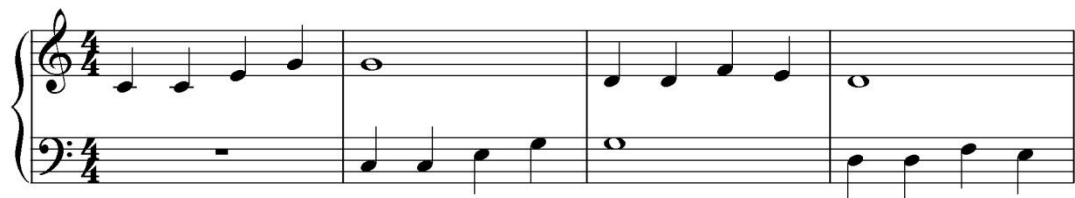


Fig. 1.1 Representación del sonido mediante el uso de un pentagrama. Puede apreciarse la clave utilizada, ritmo (4 por 4) y algunas notas de diferente duración y tono.

A continuación, describimos los principales elementos que componen la notación musical:

- **Evento:** es la unidad básica con la que se escribe la música. Representa un único sonido, el cual viene determinado por su frecuencia fundamental o pitch ( $f_0$ ). Cada evento tiene asociado un determinado intervalo de tiempo en el que se encuentra activo. Estas notas se ordenan por escalas musicales, siendo la escala musical occidental la que aparece en la figura 1.2. Esta escala está compuesta por siete notas sin alteración (teclas blancas) y cinco notas que presentan alteración (teclas negras), que se ordenan de grave a agudo en orden ascendente. Esta escala (escala occidental) queda caracterizada por presentar un intervalo constante entre notas consecutivas de un semitono (un tono queda formado por la suma de dos semitonos), por lo que a esta escala también la definen como escala cromática. Es importante diferenciar entre los sistemas de notación latino e inglés:
  - **Notación latina:** las notas que no presentan alteración son: *Do, Re, Mi, Fa, Sol, La* y *Si*, mientras que las notas con alteración se producen aumentando o disminuyendo un semitono las notas anteriores y se representan del siguiente modo: *Do#* ó *Re $\flat$* , *Re#* ó *Mi $\flat$* , *Fa#* ó *Sol $\flat$* , *Sol#* o *La $\flat$* , *La#* ó *Si $\flat$* .

- **Notación anglosajona:** las siete notas musicales sin alteración se representan mediante una letra del siguiente modo: A, B, C, D, E, F y G. La nota C corresponde con la nota Do. Este sistema también presenta cinco notas con alteración. Todo esto se aprecia en la figura 1.2.

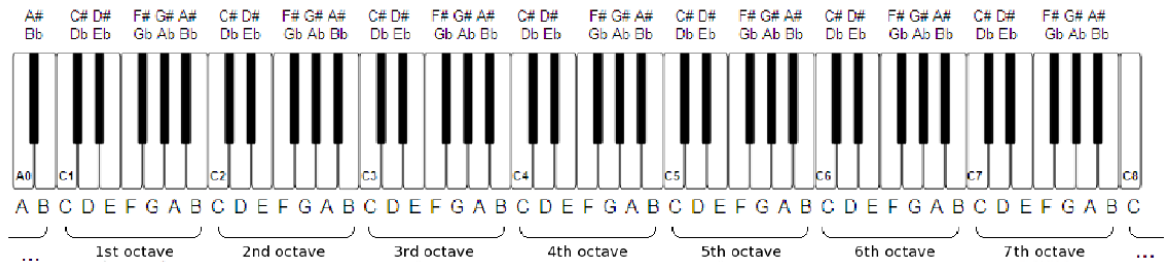


**Fig. 1.2 Representación de toda la escala musical en las teclas de un piano, puede apreciarse la notación latina y la notación anglosajona.**

- **Octava:** serie diatónica en la que se incluyen los siete sonidos consecutivos de una escala y la repetición del primero de ellos. En notación anglosajona se representa como un número que acompaña a la letra, este número indica la escala en la se encuentra dicha nota.
- **Escala diatónica:** escala en la que sus notas están separadas 2, 2, 1, 2, 2, 2 y 1 semitonos.
- **Clave:** símbolo utilizado para indicar la frecuencia fundamental de las notas escritas a continuación. Existen tres claves, las cuales se sitúan sobre diferentes líneas del pentagrama e indican el pitch de la nota que se escriba en dicha línea. Las claves son: Clave de Sol, Clave de Fa y Clave de Do.
- **Ritmo:** Se define el ritmo como “*el orden acompasado en la sucesión o acaecimiento de las cosas*”. Otra posible definición sería “*movimiento marcado por la sucesión regular de elementos débiles y fuertes, o bien de condiciones opuestas o diferentes*”.
- **Beat:** este término correspondería a la traducción de ritmo al inglés aunque también puede ser definido como pulso.
- **Compás:** consiste en una entidad métrica musical que se compone de varias unidades de tiempo (figuras con una determinada duración) organizadas en grupos. En un compás se produce una contraposición entre partes acentuadas (tónicas) y no acentuadas (átonas).



Fig. 1.3 Representación de los tres tipos de claves existentes.

- **Pentagrama:** es utilizado para escribir los símbolos musicales sobre él. Compuesto por cinco líneas y cuatro espacios entre ellas. El orden de lectura es de izquierda a derecha. El uso del pentagrama es diferente en función del instrumento que se utilice. El pentagrama se divide por líneas verticales, cada uno de los fragmentos se denomina compás. A su vez, líneas de mayor grosor y dobles indican el principio y fin de la obra. También se utilizan dos números al comienzo del pentagrama (tras situar la clave) que indican el ritmo que seguirá dicho compás.
- **Duración:** cada símbolo presenta una duración relativa con respecto a la figura negra (figura de referencia). Ver figura 1.3.

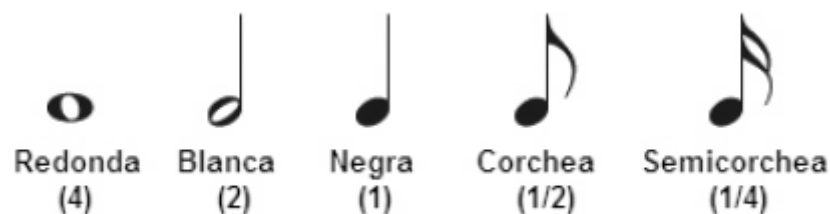


Fig. 1.4 Representación de los diferentes símbolos utilizados en notación musical para representar la duración de cada nota.

- **Acorde:** combinación de dos o más notas que suenan de manera simultánea. Distinguimos entre acordes consonante, produce un sonido agradable, y acordes disonantes, producen sonidos molestos.
- **Melodía:** sucesión temporal de notas musicales que se perciben como una única entidad. Suele estar formada por distintos pasajes que se van repitiendo a lo largo de la canción. La melodía podría ir asociada al ritmo de la canción ya que suele ser aquella parte de la misma que una persona recordaría y asociaría transcurrido un tiempo de la escucha de la canción.

- **Acompañamiento:** podemos definir el acompañamiento como todo aquello que no queda englobado dentro del concepto de melodía. Podríamos decir que los detalles musicales que un oyente olvida transcurrido un tiempo de la escucha de la canción consisten en el acompañamiento de la misma.

## 1.3. Señales sonoras

Se considera sonido a cualquier fenómeno que implique la propagación de ondas mecánicas a través de cualquier medio elástico (como un fluido) y cuyo origen sea el movimiento vibratorio de un cuerpo.

Definimos las señales sonoras como aquellas señales acústicas que se encuentran dentro del rango de audición humana, aunque existen otros tipos de sonidos, podemos diferenciar tres grandes grupos [1]:

- Infrasonidos: de 0 a 20 Hz.
- Sonidos audibles: de 20 Hz a 20 kHz.
- Ultrasonidos: frecuencias superiores a 20 kHz.

En este documento vamos a trabajar con sonidos audibles, ya que es este tipo de sonido el que se utiliza para la creación de música.

### 1.3.1. Caracterización del sonido

Existen una infinita variedad de sonidos en la vida real así como formas de clasificarlos. A continuación detallamos algunas de las características principales que nos ayudan a describir las señales sonoras.

Previamente hemos facilitado una definición de sonido, si nos centrásemos en el ámbito de las señales musicales, estas podrían ser caracterizadas mediante los siguientes tres parámetros:

- La **frecuencia fundamental** o **pitch**: nos permite ordenar los sonidos mediante una escala de frecuencia logarítmica ya que cada nota se asocia a una determinada frecuencia fundamental. De una forma algo más física podemos afirmar que un sonido representa un determinado pitch si la frecuencia de una senoide de amplitud arbitraria puede ser relacionada con el sonido percibido [2]. El concepto de frecuencia fundamental únicamente se asocia con señales periódicas o cuasi-periódicas [3].
- **Onset**: definimos el onset como el instante temporal en el que da comienzo una nota [4].
- **Duración**: intervalo temporal durante el cual una nota musical se encuentra activa en la señal de audio [5]. Su inicio viene marcado por el onset.

Siguiendo la nomenclatura de la musical occidental, las distintas notas se ordenan siguiendo una escala logarítmica de una forma parecida a la que presenta el oído humano [6]. Gracias a esto, podemos representar la frecuencia fundamental de cada nota siguiendo la ecuación (1.1) [3]:

$$f_0(Hz) = f_{0_{ref}} \cdot 2^{\frac{n}{12}}, n = -48, \dots, 0, \dots, 39 \quad (1.1)$$

Siendo  $n$  el número de semitonos entre la frecuencia de referencia ( $f_{0_{ref}}$ ), y la frecuencia deseada  $f_0$ . Se considera como frecuencia de referencia la nota La ( $f_{0_{La}} = 440 \text{ Hz}$ ).

Desde un punto de vista más físico podemos caracterizar los sonidos siguiendo otros parámetros: frecuencia fundamental, sonoridad, duración y timbre [7]:

- **Frecuencia fundamental ( $f_0$ ):**

Se considera periódica a una señal  $x(t)$  que se repite tras un determinado periodo de tiempo de forma continua. Se denomina periodo fundamental  $T_0$  al menor valor de  $T$  que satisfaga la ecuación  $x(t) = x(t + T_0), \forall t$

$$x(t) = x(t + nT_0), \quad n \in (-\infty, \infty), \quad n \in \mathbb{Z} \quad (1.2)$$

En el mundo real, las señales periódicas no pueden extenderse durante un tiempo infinito ya que son señales causales y reales. Esto implica la existencia de un periodo fundamental  $T_0$ . Definimos la frecuencia fundamental  $f_0$  en hercios de una señal periódica como la inversa del periodo fundamental  $T_0$ :

$$f_0(Hz) = \frac{1}{T_0(s)} \quad (1.3)$$

Podemos descomponer cualquier señal periódica en una suma de sinusoides (desarrollo en serie de Fourier). Estas sinusoides  $s_k(t)$  se definen como la parte real de una exponencial compleja. Por tanto, cada exponencial compleja se representa mediante una función coseno dependiente de tres parámetros: amplitud  $|a_k|$ , frecuencia  $f_0$  y fase  $\phi_k$ .

$$s_k(t) = \text{Re}(a_k \cdot e^{jk2\pi f_0 t} \text{Re}) \quad (1.4)$$

$$x(t) = \sum_{k=-\infty}^{\infty} s_k(t) = \sum_{k=-\infty}^{\infty} \text{Re}(a_k \cdot e^{jk2\pi f_0 t} \text{Re}) \quad (1.5)$$

Siguiendo el Teorema de Fourier [8], cualquier señal periódica podrá descomponerse en una suma de sinusoides armónicamente relacionadas entre sí. Si consideramos la señal  $x(t)$  real, de valor medio nulo y  $a_k = |a_k| \cdot e^{j\phi_k}$ , tendremos que  $x(t) = x^*(t) \rightarrow a_k = a_{-k}^*$

$$\begin{aligned} x(t) &= \sum_{k=-\infty}^{\infty} a_k \cdot e^{jk2\pi f_0 t} = \sum_{k=-\infty}^{-1} a_k \cdot e^{jk2\pi f_0 t} + \sum_{k=1}^{\infty} a_k \cdot e^{jk2\pi f_0 t} = \\ &= \sum_{k=1}^{\infty} a_{-k} \cdot e^{-jk2\pi f_0 t} + a_0 + \sum_{k=1}^{\infty} a_k \cdot e^{jk2\pi f_0 t} = \\ &= \sum_{k=1}^{\infty} (a_k \cdot e^{jk2\pi f_0 t} + a_{-k} \cdot e^{-jk2\pi f_0 t}) = \\ &= \sum_{k=1}^{\infty} (a_k \cdot e^{jk2\pi f_0 t} + a_k^* \cdot e^{-jk2\pi f_0 t}) = \\ &= \sum_{k=1}^{\infty} 2\text{Re}(a_k \cdot e^{jk2\pi f_0 t}) = \sum_{k=1}^{\infty} 2\text{Re}(|a_k| \cdot e^{jk2\pi f_0 t} \cdot e^{j\phi_k}) \\ &= 2 \sum_{k=1}^{\infty} |a_k| \cos(2\pi f_k t + \phi_k) \end{aligned} \quad (1.6)$$

A cada una de estas componentes  $f_k = k f_0$ ,  $k \in N$  se las conoce como armónicos y presentan una frecuencia múltiplo de la frecuencia fundamental o pitch. También se ha de tener en cuenta que las señales musicales no son señales estacionarias a largo plazo, esto implica una variación en sus propiedades estadísticas (media y autocorrelación) a lo largo del tiempo. Si queremos considerar una señal estacionaria, debemos trabajar con pequeñas porciones de nuestra señal original. A estos fragmentos (de decenas de milisegundos de duración) se les conoce como *tramas* o *frames*, y sí que presentan unas propiedades estadísticas constantes en dicho intervalo de tiempo.

- **Sonoridad:**

Consiste en un atributo de percepción auditiva. Gracias a él, podemos ordenar los sonidos basándonos en su intensidad. Aunque pueda parecer que la sonoridad de un determinado sonido dependa exclusivamente de la potencia con la que se percibe dicho sonido, existen otros factores que influyen en la sonoridad de una señal. Así, la frecuencia de dicho sonido es de vital importancia ya que el oído humano no percibe todas las frecuencias de igual forma. La estructura temporal así como la duración del sonido también influyen en la sonoridad del mismo.

Para poder clasificar los sonidos según su sonoridad, se han creado una serie de curvas denominadas *Curvas Isofónicas de Fletcher y Munson* (figura 1.5). Estas toman como referencia la frecuencia de 1 kHz, así un tono de 1 kHz presenta un nivel de sonoridad igual a su nivel de presión sonora (Sound Pressure Level, SPL). Al depender de la frecuencia, dos sonidos que se emitan con un mismo nivel de potencia pero cuya frecuencia se diferencie serán percibidos con un distinto nivel de sonoridad.

La unidad de medida de esta magnitud son los *fonos* o *fonios*.

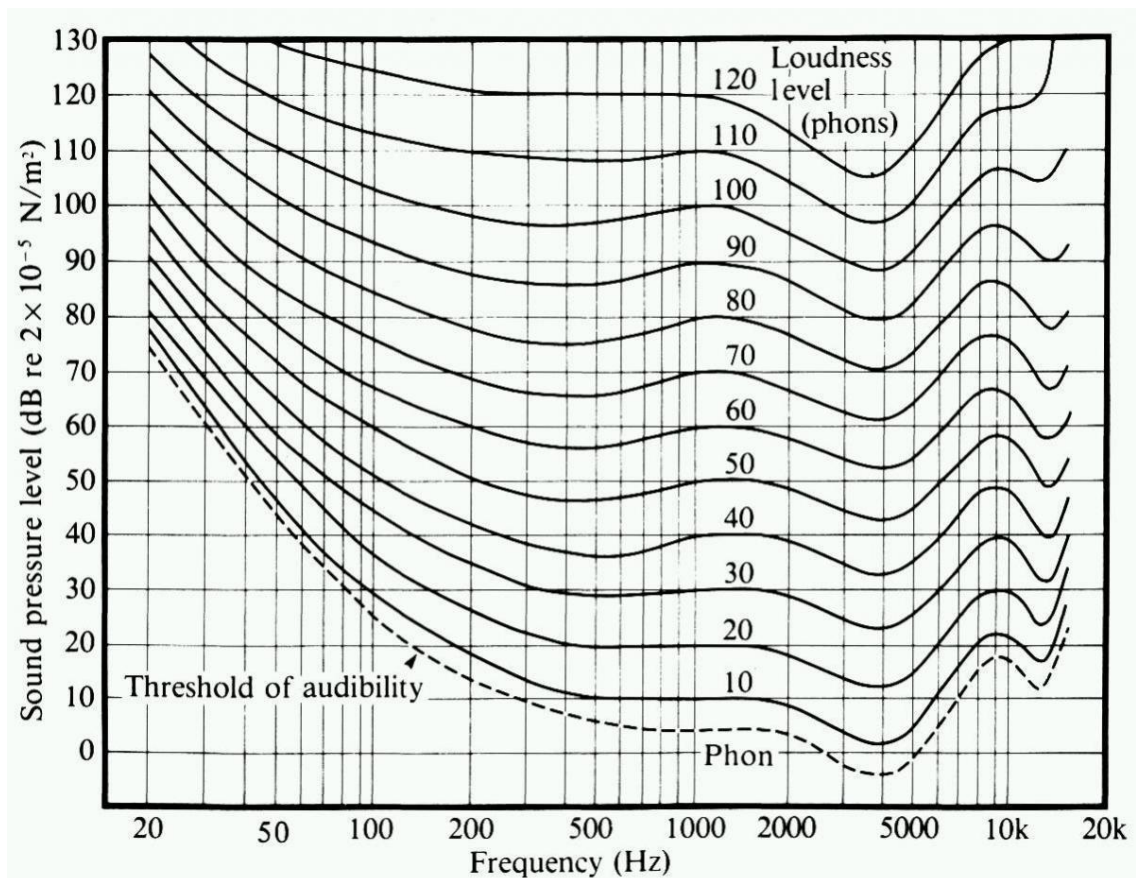


Fig. 1.5 Curvas isofónicas de Fletcher y Mudson que permiten conocer el nivel de sonoridad de un sonido en función de su frecuencia y nivel de presión sonora.

- **Duración:**

Este parámetro ya fue explicado en el apartado 1.2 y consiste en el intervalo de tiempo en el que una nota se encuentra activa. El momento de inicio de duración de un sonido se conoce como **onset** mientras que su fin viene se define **offset**.

- **Timbre:**

Este atributo no puede cuantificarse sino que consiste en un atributo perceptual asociado al concepto de color del sonido [9] y está fuertemente ligado a la diferenciación de instrumentos y al reconocimiento de fuentes sonoras [10]. Gracias al timbre podemos distinguir entre distintos tipos de notas que presenten el mismo pitch, duración e intensidad, por ejemplo, el timbre nos permite distinguir entre una misma nota tocada por un piano y por un violín. Es difícil caracterizar el timbre de cada instrumento ya que se necesitan varios parámetros que lo describen, algunos de ellos son el número de parciales, la envolvente espectral y temporal, tiempo de ataque o irregularidades espectrales [11].

Entre las características que afectan a la percepción auditiva se encuentran el material de construcción del instrumento en cuestión, la forma de producir el sonido (ej. viento o cuerda), excitación que genere el sonido (ej. punteada o frotada) o el grado de inarmonicidad entre otras.

### 1.3.2. Tipos de señales sonoras

Una vez descritas las características que nos ayudan a definir los sonidos podemos aglutinar estos según distintos puntos de vista, así podemos agrupar los sonidos atendiendo al número de canales, al número de instrumentos por el que están compuestas así como al uso de distintos instrumentos o instrumentos con el mismo timbre entre otras muchas clasificaciones. En concreto nos centramos en los siguientes tipos de clasificaciones:

- **Sonidos armónicos e inarmónicos.**

Definimos un **sonido armónico** como aquel sonido compuesto por sinusoides cuyas frecuencias fundamentales se relacionan entre sí de una forma armónica, esto quiere decir que sus componentes frecuenciales se encuentran situadas en múltiplos de su frecuencia fundamental ( $f_0$ ). A este tipo de sonido también se le conoce como sonido periódico o cuasi-periódico. Los instrumentos de viento madera, viento metal así como los de cuerda percutida, cuerda frotada y cuerda punteada producen este tipo de sonidos.

Por el contrario, los **sonidos inarmónicos** no son periódicos ni cuasi-periódicos. Estos sonidos son producidos por dos grandes grupos de instrumentos percusivos: sonidos producidos con maza (como el xilófono) y sonidos producidos con batería (como un tambor). Esta diferenciación se debe a que el primer grupo sí que presenta una ligera periodicidad temporal que permite asociarse a un determinado pitch (frecuencia fundamental) en el dominio frecuencial. Por el contrario, los instrumentos del segundo grupo no presentan absolutamente ninguna periodicidad temporal por lo que no pueden asociarse a ninguna frecuencia fundamental.

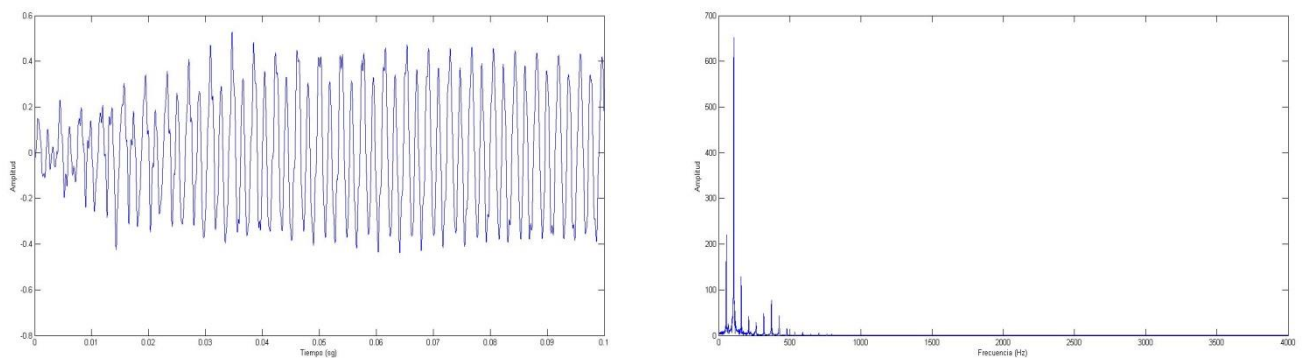


**Fig. 1.6** A la izquierda tenemos una señal en el dominio del tiempo de la nota Do interpretada por un piano con una duración de 100 ms a la derecha tenemos la transformada de Fourier donde puede apreciarse su frecuencia fundamental.

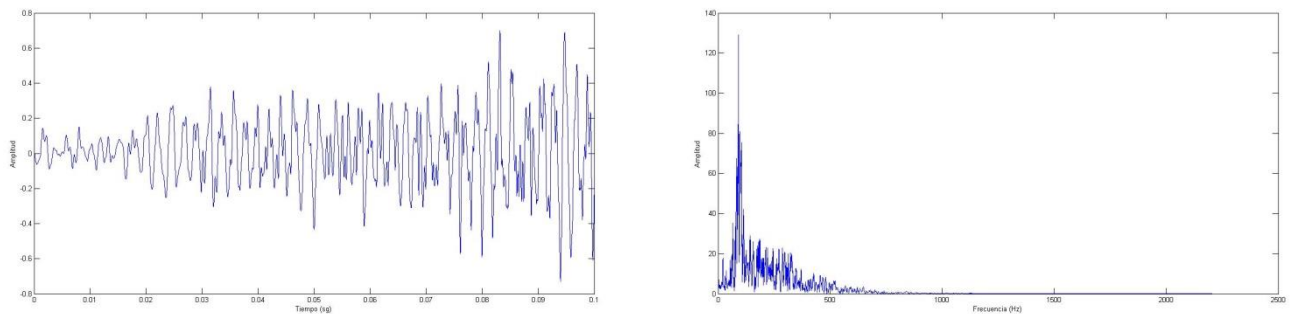


**Fig. 1.7** Sonido inarmónico interpretado por un gong. A la derecha se encuentra su transformada de Fourier donde puede apreciarse una mayor distribución en frecuencia de la energía de la señal.

- **Señales monoaural, estéreo y multicanal.**

Los **sonidos monoaurales** (comúnmente conocidos como mono) proceden de señales mono-canal que se componen por la suma directa de todas las fuentes de sonido que aparecen en la grabación. Esta característica implica una mayor dificultad a la hora de analizar y procesar las señales dado que el hecho de tener toda la información en un único canal nos impide obtener información espacial de las fuentes.

Sonidos **estereofónicos**: tienen su origen en señales de audio de dos canales ya que en el proceso de grabación se utilizan, al menos, dos micrófonos independientes. El hecho de utilizar dos micrófonos aporta una importante información espacial sobre la localización relativa de las fuentes de sonido (instrumentos).

Por último, definimos los **sonidos multicanal** como aquellos que se generan a partir de un mínimo de 4 canales independientes. Estos sonidos también se conocen como *surround* y necesitan equipos especiales para su reproducción (para aprovechar sus características). Así, para la reproducción de estas señales necesitaremos disponer de mayor cantidad de altavoces, ya que además del canal izquierdo y derecho (señales estéreo), tendremos la opción de reproducir un canal de voz y otro que contiene sonidos de muy baja frecuencia (*subwoofer*).

- **Sonidos monofónicos y polifónicos.**

El concepto de monofonía y polifonía se relaciona con el hecho de que en cada instante temporal de nuestra señal tengamos señales procedentes de uno o más instrumentos simultáneamente. Así por ejemplo una señal procedente de un solo de guitarra sería considerada monofónica mientras que una grabación de un grupo de rock consistiría en una señal polifónica.

En un **sonido monofónico**, para cada instante de tiempo, únicamente se encontrará activa una frecuencia fundamental  $f_0$ .

En un **sonido polifónico** encontramos para cada instante de tiempo dos o más frecuencias fundamentales  $f_0$  activas. Estas procederán de un único instrumento o de varios instrumentos que se encuentren tocando simultáneamente.

- **Sonidos monotímbricos y multitímbricos.**

Para esta caracterización de los sonidos nos basamos en el número de instrumentos que componga la señal analizada.

Así, el concepto de **monotímbrico** viene directamente relacionado con aquellas obras compuestas por señales que provienen de instrumentos que presentan el mismo timbre o envolvente espectral. Por el contrario, se definen los sonidos **multitímbricos** como aquellos que están formados por señales procedentes de dos o más instrumentos que presentan distinto timbre.

Es importante la distinción entre sonidos monofónicos y monotímbricos dado que no son excluyentes. Una señal compuesta exclusivamente por un sonido de piano sería una señal monotímbrica pero esta será monofónica cuando suene una única nota y

polifónica cuando suene un acorde formado por dos o más notas de este instrumento. Ocurre lo mismo a la hora de distinguir sonidos multitímbricos. Aunque se estén tocando diferentes instrumentos, si la nota interpretada por todos es la misma nos encontraríamos con una señal monofónica mientras que si se interpretan diferentes notas musicales la señal sería multitímica y polifónica.

## 1.4. Sonidos rítmicos

Para conocer en que consiste un sonido rítmico es necesario conocer previamente la definición de ritmo, la cual se facilita en la sección 1.2.

El ritmo es un atributo básico de la mayor parte de las artes, en especial de la música, la poesía y la danza [12]. En las partituras occidentales convencionales, podemos asociar el ritmo con el tiempo (dirección horizontal), por el contrario, la melodía suele relacionarse con la altura musical (dirección vertical) [13]. El ritmo viene marcado por una sucesión de elementos opuestos, como lo sería el sonido y el silencio, inaudible pero implícito. Por tanto, será necesaria la continua repetición de un patrón repetitivo de una corta duración. En la nomenclatura musical se utilizan los términos *beat*, *upbeat* y *downbeat*, siendo el *beat* comúnmente conocido como pulso mientras que *upbeat* y *downbeat* hacen referencia a los pulsos inicial y final de un compás. Cada ritmo es único y presenta una organización propia, de ahí la dificultad a la hora de extraer la pista rítmica.

En este documento asumiremos como sonidos rítmicos aquellos procedentes de instrumentos percusivos (ej. bombo, timbales, platillos) y de otros instrumentos como el contrabajo y el bajo eléctrico que también se utilizan como instrumentos de acompañamiento. A estos instrumentos, añadiremos en este documento otros posibles efectos o sonidos que puedan presentar un patrón rítmico a lo largo del tiempo y, por tanto, que puedan incluirse dentro de la definición de sonidos rítmicos.

Esta definición de sonidos rítmicos es algo subjetiva ya que, por ejemplo la voz de un cantante que repita una estrofa podría ser considerada como repetitiva. De ahí que recurramos a lo que en este documento definiremos como **repetitividad a corto y largo plazo**.

Definimos los sonidos rítmicos cuya repetitividad sea a corto plazo como aquel tipo de sonido que se produce varias veces en un segundo o en periodos de tiempo reducido. Por el contrario, los sonidos repetitivos a largo plazo serían aquellos que una vez se dan, no vuelven a producirse hasta que ha transcurrido un tiempo relativamente elevado (segundos o decenas de segundos). Como ejemplos de sonidos rítmicos a corto plazo tenemos las siguientes figuras, en ellas se aprecia un sonido de batería en el dominio temporal (obsérvese la repetitividad en un corto periodo de tiempo) y su espectrograma. Gracias a este espectrograma se aprecia como un mismo patrón espectral se repite a lo largo del tiempo.

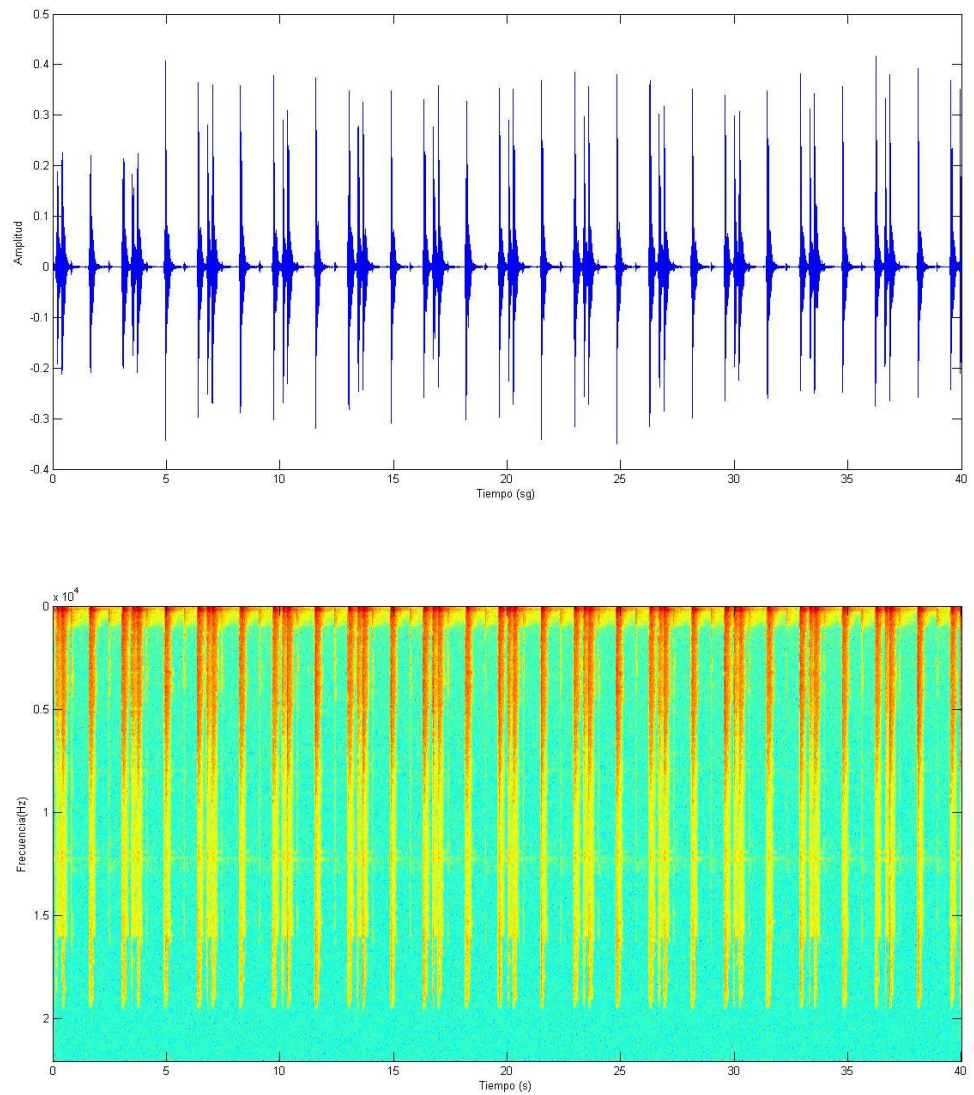


**Fig. 1.8 Representación temporal de una señal rítmica procedente de una batería. En la imagen inferior se representa el espectrograma de dicha señal donde se aprecia el reparto en frecuencia de los golpes de batería.**

## 1.5. Sonidos no rítmicos

Una vez conocida la definición de sonido rítmico, podemos definir como sonidos no rítmicos a todos aquellos que no se engloben dentro del conjunto anterior. Estos sonidos no presentan un patrón repetitivo a lo largo del tiempo y, por tanto, pueden aparecer en cualquier momento de la canción. Podemos incluir dentro de este tipo de sonidos aquel que producen instrumentos musicales como guitarra, violín, piano, flauta y muchos otros así como la voz o ciertos efectos musicales introducidos por ordenador. Aunque el propósito de este documento sea el de la obtención de la pista rítmica, al tratarse de señales aditivas, se podría obtener todo el resto de señal mediante una simple resta, es decir, una vez extraída nuestra señal rítmica, esta se resta a la señal original tras lo que únicamente quedaría todo aquello que no se haya considerado sonido rítmico.

Se ha de prestar cierta atención a lo que consideramos sonido rítmico y no puesto que un sonido no rítmico podría considerarse rítmico con un patrón de repetitividad cuyo periodo es más elevado. Las siguientes figuras muestran patrones no rítmicos.

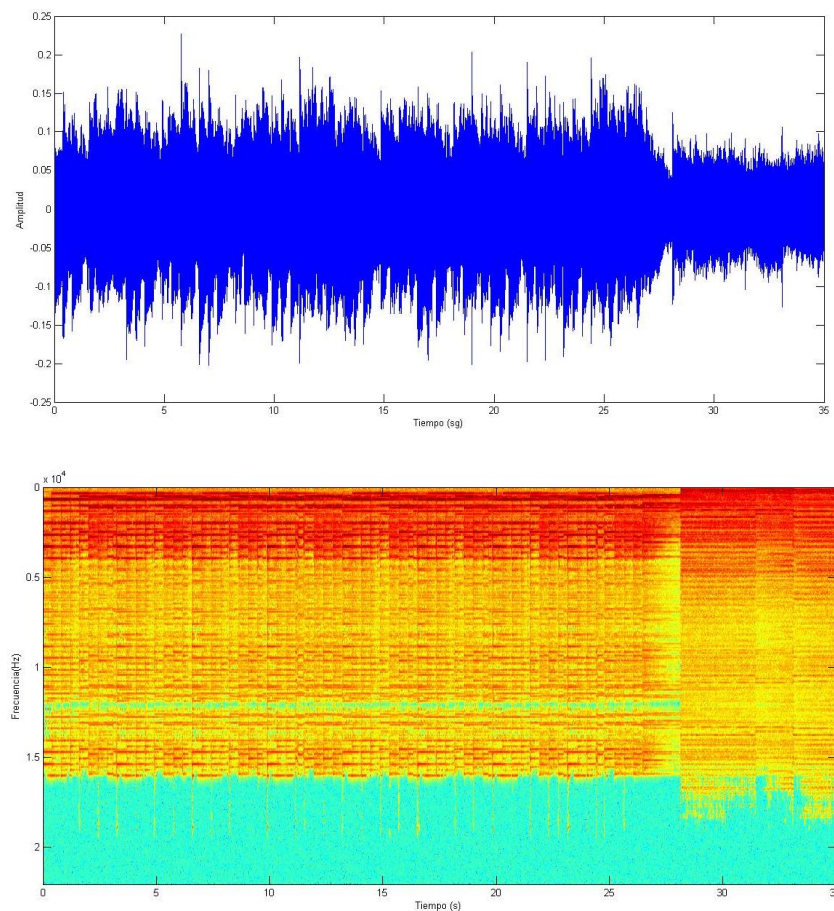


**Fig. 1.9** Representación temporal de una señal no rítmica procedente de las pistas de guitarra, voz y otros instrumentos. En la imagen inferior se aprecia su espectrograma y el reparto en frecuencia de la energía (centrado en torno a ciertas frecuencias).

A continuación se muestra como sería el espectrograma de una señal compuesta por ambos tipos de sonidos. A pesar de haber introducido en esta sección los espectrogramas, no será hasta la sección 4.1 donde se explique su interpretación.

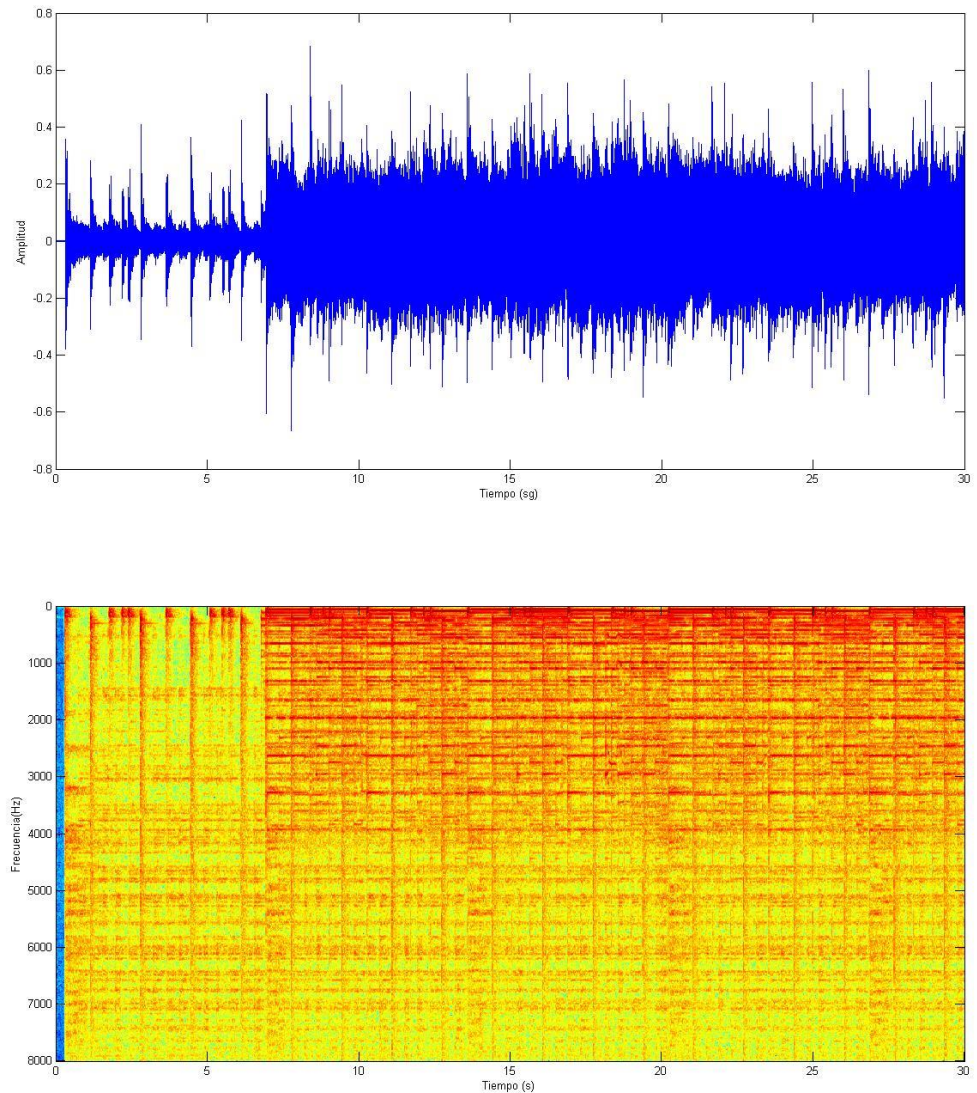


**Fig. 1.10** Representación temporal de la señal completa (pista rítmica + pista no rítmica). En la imagen inferior puede apreciarse el espectrograma de la suma de ambas pistas.

# Capítulo 2

## 2. Objetivos

La repetición es una parte fundamental a la hora de entender la estructura de una composición musical. Las composiciones musicales básicamente se componen de dos tipos de señales: rítmicas (aquellas que se repiten a corto plazo a lo largo de la composición como pueden ser baterías y línea de bajo) y no rítmica (aquellas que no se repiten a corto plazo a lo largo de la composición como pueden ser instrumentos armónicos y singing-voice). Este trabajo de fin de grado está centrado en la separación de fuentes sonoras. Concretamente se ha implementado un sistema capaz de separar una señal musical en dos pistas musicales diferentes: rítmica y no rítmica. Para ello, se utilizarán diferentes herramientas de procesamiento de señal asumiendo que los sonidos que componen la pista rítmica presentan alta similitud espectral y se repiten a corto plazo, lo cual se puede utilizar como información para discriminar este tipo de eventos musicales rítmicos. El proyecto se implementa utilizando el entorno MATLAB.

Para alcanzar esta meta, ha sido necesario un estudio del estado del arte (descrito en el capítulo 3 de este documento) así como adquirir ciertos conocimientos básicos sobre teoría musical y características de las señales sonoras (estos conocimientos ya han sido expuestos en el capítulo 1).

Una vez alcanzados estos objetivos iniciales, se realiza una revisión del estado del arte para conocer el estado actual de la técnica de extracción de la pista rítmica (capítulo 3).

En el capítulo 4 se recogen diversas herramientas que serán necesarias a la hora de implementar sistemas de separación de fuentes acústicas como pueden ser NMF o algoritmo de seguimiento del tiempo de beat.

A continuación se expone la implementación de diversos sistemas que realizan dicha separación. En concreto se estudian cuatro sistemas diferentes, todos ellos explicados en detalle a lo largo del capítulo 5.

A su vez, el capítulo 6 se centra en la elaboración de unas determinadas bases de datos que nos permitan comprobar la calidad de las señales obtenidas mediante la realización de una serie de medidas así como algunos detalles de los posibles resultados obtenidos.

Las conclusiones a las que se llega tras probar el algoritmo sobre diversas señales así como posibles líneas con las que continuar este trabajo, vienen expuestas a lo largo del capítulo 7.

Por último se ha desarrollado una sencilla interfaz de usuario que permite realizar todo el procedimiento de forma clara y simple para cualquier persona no versada en la programación ni en el lenguaje Matlab, con el que se ha trabajado, pueda realizar la separación que en este documento se explica. En el capítulo 9 se describe una breve guía sobre dicho interfaz.

Para la implementación de este trabajo de fin de grado se ha seguido la siguiente metodología:

- Recopilación y estudio de material bibliográfico relacionado con el tema.
- Diseño e implementación de la herramienta de extracción de la pista rítmica musical.
- Evaluación del sistema implementado.
- Desarrollo de una interfaz amigable para su uso por el usuario.
- Redacción de esta memoria.

# Capítulo 3

## 3. Estado del arte

### 3.1. Separación de fuentes acústicas

A día de hoy se ha trabajado en profundidad en lo que a separación de fuentes acústicas se refiere (*Sound Source Separation, SSS*). Para realizar esta separación se han propuesto distintos métodos, algunos de los cuales se describen a continuación. Desde el uso de filtros de mediana y matrices de similitud hasta algoritmos más complejos como NMF, NMPCF (Non-negative Matrix Partial Co-Factorization), Sparse Coding o ICA (*Independent Component Analysis*), los métodos seguidos tratan de realizar algún tipo de separación, generalmente la obtención de una señal que contenga la información percusiva de la señal y de otra que posea la información sobre los armónicos. También se ha intentado eliminar la voz de las señales con el propósito de crear karaokes y otras aplicaciones de ese estilo. En este documento el propósito de dicha separación será la obtención de la pista rítmica de la señal (Capítulo 2).

El **análisis de componentes independientes (ICA)** consiste en un método de descomposición de señales utilizado, por norma general, para la separación de fuentes en sistemas ciegos y con señales multicanal [14]. Este algoritmo suele utilizarse como solución al problema conocido como *cocktail party problem*, en el que se intenta separar las voces de distintas personas que hablan al mismo tiempo y que, por tanto, se encuentran mezcladas en las señales. Al situar varios micrófonos en una sala cada uno de ellos generará una señal diferente aunque en todas aparecerán todas las voces. Estas señales se diferenciarán entre sí dado que los micrófonos se sitúan en posiciones distintas así como los hablantes. Esto puede extrapolarse al problema de separación de instrumentos musicales en señales de audio.

Estos algoritmos suponen que cada trama de cada canal,  $x_c(t)$  consiste en una mezcla lineal de las señales que provienen de cada fuente,  $g(t)$ , de modo que:

$$x_c(t) = Bg(t)$$

Siendo  $B$  la matriz de mezcla cuyos coeficientes son desconocidos.

A partir de la información aportada por  $x(t)$ , un algoritmo ICA trata de invertir el procedimiento que se sigue para la mezcla mediante la búsqueda de una matriz de separación óptima  $W$ , quedando la estimación de las fuentes de la siguiente manera:

$$g_{est}(t) = Wx(t)$$

Idealmente,  $W$  y  $B$  serían matrices inversas y la reconstrucción sería perfecta, quedando las señales perfectamente separadas. ICA busca reconstruir fuentes que sean lo más independientes posibles entre ellas. Algunos algoritmos que siguen la filosofía ICA son:

- J Barry. Poplyphonic music transcription using independent component analysis. MSc, 2003 [28].
- M Plumbey. Adaptative lateral inhibition for non-negative ICA. *Proceedings of the International Conference on Independent Component Analysis and Blind Signal Separation (ICA)*, 2001 [29].
- M Plumbey. Algorithms for nonnegative independent component analysis. *IEEE Transactions on Neural Networks*, 14(3): 534-543, 2003 [30].

Los algoritmos de **Sparse Coding** representan la señal mezcla de entrada eligiendo, de entre un número elevado de elementos, un pequeño conjunto de estos [3]. Esto se utiliza para la separación de datos con distinto origen. Se aplican una serie de restricciones a las ganancias  $G$ , eso conlleva una elevada probabilidad de que un elemento de  $G$  sea cero. En consecuencia, solamente unas pocas componentes de  $G$  se activarán para cada instante de tiempo y esta activación será durante periodos muy cortos de tiempo. Este tipo de restricciones cuadran con lo que ocurre en la música real, ya que en esta, únicamente un pequeño grupo de notas suenan de forma simultánea [3].

En otros documentos, la separación de fuentes acústicas se realiza mediante otro tipo de algoritmos, algunos ejemplos son:

- **Separación armónico-percusiva utilizando filtrado de mediana** [15].  
En este documento se presenta un método sencillo y efectivo para realizar una separación armónica y percusiva de una señal de audio monoaural. Para ello se hace uso de filtros de mediana a lo largo de los distintos frames de tiempo en un espectrograma, elimina los eventos percusivos y realza los armónicos. Por el contrario, se realiza un filtrado de mediana a lo largo de los bins de frecuencia para realzar los eventos percusivos y suprimir los armónicos. Estos espectrogramas se utilizan en la construcción de máscaras que se aplican al espectrograma original para realizar la separación.
- **Separación de música y voz utilizando la matriz de similitud** [16].  
Este documento aprovecha el hecho de que la repetitividad sea fundamental a la hora de percibir la estructura de la música. Este algoritmo realiza la separación de fuentes aprovechando la repetitividad de la música, a pesar de que esta se pueda producir de forma intermitente sin ningún patrón reconocible. En lugar de buscar unas determinadas periodicidades, el método

propuesto utiliza una matriz de similitud para identificar los elementos que se repiten. A continuación, se calcula un espectrograma “modelo” de repetitividad y se extrae los patrones de repetitividad usando máscaras de tiempo-frecuencia.

- **Método de separación de música y voz basado en la extracción de la estructura de repetición musical** [17].

Este documento trata de aprovechar la repetición en música ya que esta es esencial en la música. Los autores proponen un método de separación basado en la extracción de esta propiedad. En primer lugar se busca lo que se conoce como periodo de repetición y a continuación, se segmenta el espectrograma. Se realiza un promediado de los segmentos para crear un modelo de espectrograma repetitivo. Por último, cada bin de los segmentos se compara con el modelo y se construyen máscaras con esta información.

- **Separación de sonidos de sistemas supervisados y semisupervisados en señales monoaurales** [18].

Este método describe la metodología seguida para la separación de sonidos en señales monoaurales. Se presenta un modelo que puede aprender sonidos en función de su distribución de energía en tiempo-frecuencia (espectrograma). Gracias a él, pueden extraerse sonidos de mezclas en dos tipos de escenarios. El primero es aquel en el que conocemos todos los sonidos que forman la mezcla y en el segundo, únicamente conocemos el sonido recuperado o las interferencias que le afectan.

Otros documentos estudiados para la realización de este proyecto han sido los siguientes:

- From Learning Music to Learning to Separate. Paris Smaragdis. Mitsubishi Electric Research Laboratories [31].
- On the Use of Masking Filters in Sound Source Separation. Derry Fitzgerald and Rajesh Jaiswal. Dublin Institute of Technology [32].
- Percussive/harmonic sound separation by non-negative matrix factorization with smoothness/sparseness constraints. Francisco Jesus Cañadas Quesada, Pedro Vera Candeas, Nicolas Ruiz Reyes, Julio Carabias Orti y Pablo Cabañas Molero [33].
- Separación de fuentes sonoras en señales musicales. Francisco José Rodríguez Serrano [34].

### 3.2. Acondicionamiento de la señal

Basándonos en los tipos de señales sonoras descritas en el capítulo 1, describimos las señales con las que trabajamos en este documento como señales **monoaurales**, **polifónicas** y **multitímbricas**. Es por ello que nuestro algoritmo transformará cualquier señal introducida que posea más de un canal en una señal monoaural mediante la suma de todos sus canales en una única señal.

En esta sección nos centramos en describir el procedimiento necesario para transformar nuestra señal de un dominio temporal al dominio frecuencial. Para ello debemos obtener el espectrograma de la señal ( $X(t, f)$ ) mediante la llamada *STFT* (*Short-Time Fourier Transform*). Explicamos en detalle en que consiste y como se obtiene un espectrograma.

Para la descripción de este apartado tomaremos como ejemplo el siguiente espectrograma:

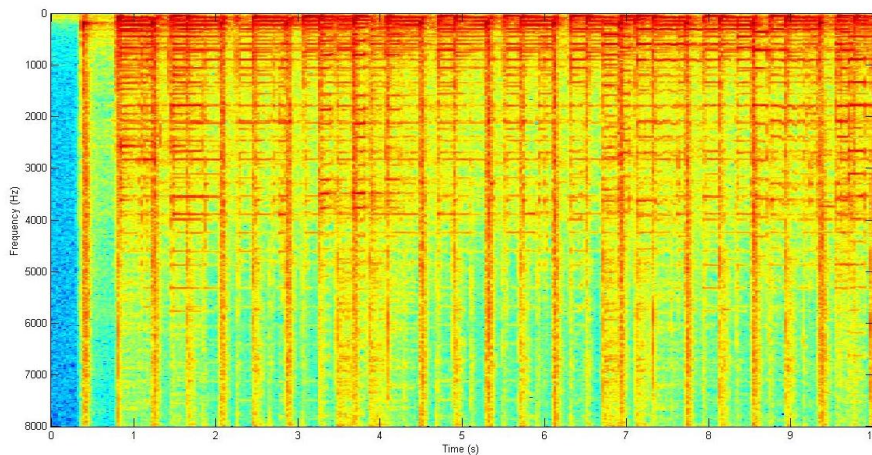


**Fig. 3.1** Espectrograma correspondiente a un fragmento de los diez primeros segundos de duración de la canción *Sultans of Swing* del grupo *Dire Straits*.

Un espectrograma consiste en la representación bidimensional de una señal en la que el eje de abscisas nos indica los tiempos y el eje de ordenadas las frecuencias. De este modo cada punto del espectrograma, denominado “Bin”, tendrá un determinado valor. Dicho valor consiste en la cantidad de energía que esta señal posee en esa frecuencia y en ese instante determinado. De este modo y de un simple vistazo, accedemos a gran cantidad de información sobre nuestra señal y, como vamos a explicar, pueden observarse ciertas características. En concreto buscamos encontrar de forma separada la parte armónica (que consideramos no repetitiva o no rítmica) de la señal de la parte percusiva (la cual posee características asociadas a señales repetitivas, también conocidas como rítmicas).

Partiendo de nuestro espectrograma original, es decir, de la señal mezclada, podemos realizar la siguiente descomposición:

$$X_m = X_r + X_{nr} \quad (3.1)$$

Siendo  $X_m$  el espectrograma de partida,  $X_r$  el espectrograma correspondiente a la parte rítmica de la señal y  $X_{nr}$  el espectrograma correspondiente a la parte no rítmica de la señal.

El principal propósito de este documento es la realización de dicha separación, lo cual no es tarea sencilla ya que nuestras señales se generan gracias a la suma de otras señales. Al trabajar con señales de audio, como por ejemplo una canción, debemos tener en cuenta que estas se componen de otras muchas señales correspondientes a las diferentes pistas. Si tomamos como ejemplo una canción de rock, podemos apreciar que las señales pertenecientes a este estilo musical se componen generalmente por pistas de guitarra, batería, bajos, voz e incluso otros instrumentos como piano o violín, además de los posibles efectos que se hayan podido añadir en el estudio de grabación. A la hora de realizar la grabación cada uno de estos instrumentos, esta se realiza por separado, generando una serie de pistas correspondientes a los diferentes instrumentos. Una vez están todas grabadas y mantienen el sincronismo entre sí se procede a su suma para obtener la señal multipista que contiene la información de todas las demás pistas. Esto es lo que se ha descrito previamente al indicar que nuestra señal proviene de la suma de otras muchas señales.

Conociendo la composición de nuestra señal, es fácil entender que para un determinado bin del espectrograma mezcla,  $X_m$ , su energía sería la suma de la energía de dicho bin en el espectrograma de ritmo,  $X_r$ , mas la energía de dicho bin en el espectrograma no rítmico,  $X_{nr}$ . Esto es algo lógico ya que si tocamos una nota de piano, por ejemplo la nota *La* a 440 Hz y simultáneamente se está dando un golpe de batería, que contiene energía en la mayor parte de frecuencias del espectro audible, tendremos una suma de energías en dicha frecuencia. Aunque el oído humano pueda diferenciar los sonidos por el timbre (apartado 1.3.1), desde el punto de vista del análisis frecuencial de señales no sabemos claramente que ha podido ocurrir ya que únicamente sabemos que en ese determinado instante tenemos energía en unas frecuencias sin saber si esa energía procede de una voz, de batería, de piano, etc.

De este modo, se llega a la conclusión de que un determinado porcentaje de la energía de dicho bin provendrá de la pista rítmica mientras que el resto proviene de la pista no rítmica.

Previamente se han descrito las señales armónicas y percusivas desde el punto de vista musical, si nos trasladamos de nuevo a nuestro espectrograma la interpretación sería la siguiente:

- Las señales percusivas como golpes de batería, platillos, bombos y demás poseen una gran cantidad de energía concentrada en un espacio de tiempo muy breve y repartida a lo largo de casi todo el espectro de frecuencias audibles. En un espectrograma esta característica implica unas estrechas columnas que se repiten a lo largo del tiempo ya que por norma general la señal de percusivos se utiliza para llevar el ritmo de la canción, de ahí su repetitividad.

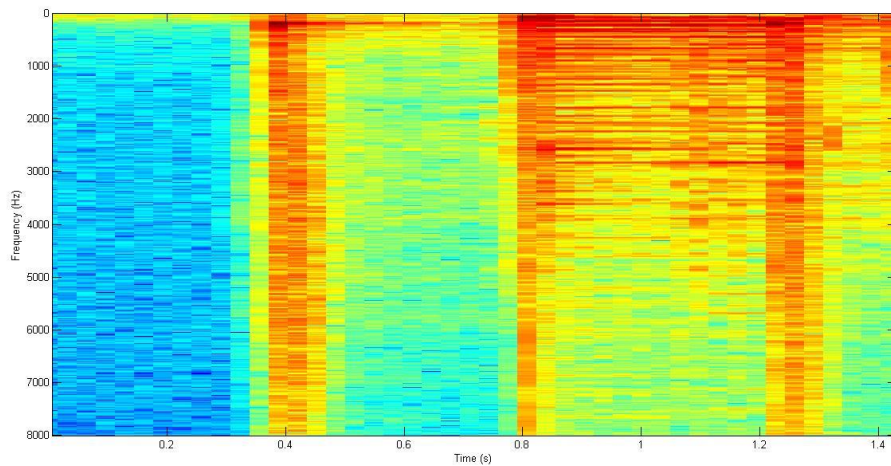


**Fig. 3.2** Fragmento de un espectrograma generado a partir de una señal percusiva.

- Las señales armónicas por el contrario concentran su energía en una determinada frecuencia (y los armónicos de esta) y suelen expandirse a lo largo del tiempo durante un periodo bastante mayor al de los percusivos. En un espectrograma esa característica se interpreta como una línea horizontal, colocada a la altura de la frecuencia fundamental de dicha nota, acompañada de otras líneas con menor energía situadas en múltiplos de la frecuencia fundamental.

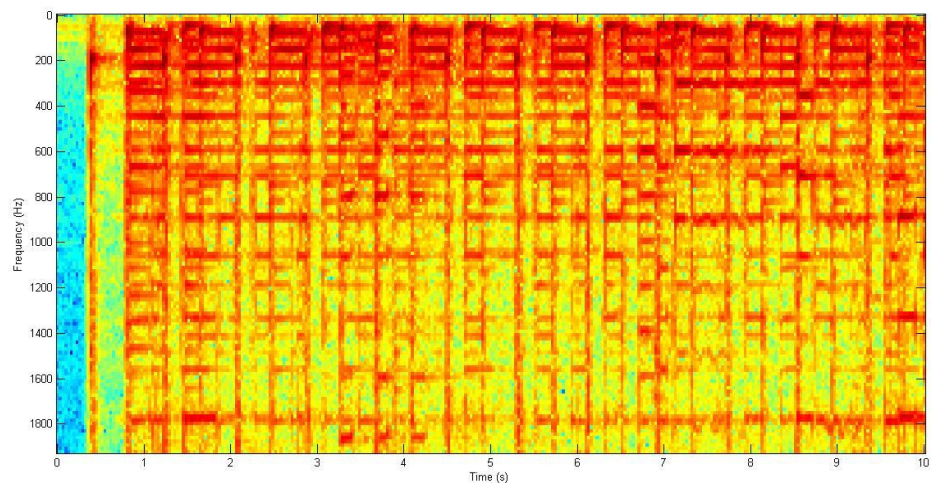


**Fig. 3.3** Fragmento de un espectrograma generado a partir de una señal armónica.

Conocido esto, de un simple vistazo y de forma muy básica, podemos asociar columnas que posean gran cantidad de energía con señales percusivas y agrupaciones de líneas horizontales de mayor duración con señales armónicas. Esta diferenciación resulta de gran utilidad ya que, como se ha supuesto, la pista rítmica de nuestra señal está formada básicamente por sonidos percusivos junto con la línea de bajos. Por tanto, será muy importante analizar el comportamiento de estas “columnas” que representan a los percusivos así como de las zonas del espectrograma en las que se encuentren las frecuencias más bajas (puesto que el bajo emite sonidos de muy baja frecuencia).

Llegado este punto podemos interpretar de forma correcta un espectrograma, queda por tanto conocer el procedimiento que se sigue para obtenerlo. Para ello será necesario realizar lo que se conoce como *STFT* lo que implica un proceso de enventanado de la señal de entrada.

El enventanado consiste en seleccionar un fragmento de la señal de  $N$  muestras (siendo  $N$  el número de muestras de la ventana) y realizar la *FFT* (*Fast Fourier Transform*) sobre dicha parte de la señal. Acto seguido desplazamos la ventana sobre la señal un número de muestras (indicado como  $S$ ) y realizamos de nuevo la FFT sobre este nuevo fragmento, concatenando los resultados obtenidos. Este proceso se realiza a lo largo de toda la señal. A menos que  $S$  tenga un valor superior a  $N$ , existirá un determinado solapamiento entre el final de un fragmento y el comienzo del siguiente.

La ecuación seguida para el cálculo de la STFT es:

$$STFT\{x[n]\} = X(m, \omega) = \sum_{n=-\infty}^{\infty} x[n]w[n-m]e^{-j\omega n} \quad (3.2)$$

Siendo  $x[n]$  la señal que queremos transformar y  $w[n-m]$  una ventana que se desplaza. Este resultado tiene valores complejos, los cuales se colocan en una matriz que contiene la magnitud y la fase para cada punto en tiempo y frecuencia. Definimos un espectrograma como el valor de la magnitud elevado al cuadrado:

$$Espectrograma = \{x[n]\} \equiv |X(m, \omega)|^2 \quad (3.3)$$

Existen muchos tipos de ventanas para realizar la STFT aunque las más utilizadas son la ventana rectangular, Bartlett, Hamming, Hanning y Blackman. El tipo de ventana escogido afecta a la amplitud relativa de los lóbulos secundarios así como a la anchura del lóbulo principal, esta también se ve afectada por la longitud de la ventana ( $N$ ), reduciéndose dicha anchura conforme aumentamos la longitud de la ventana.

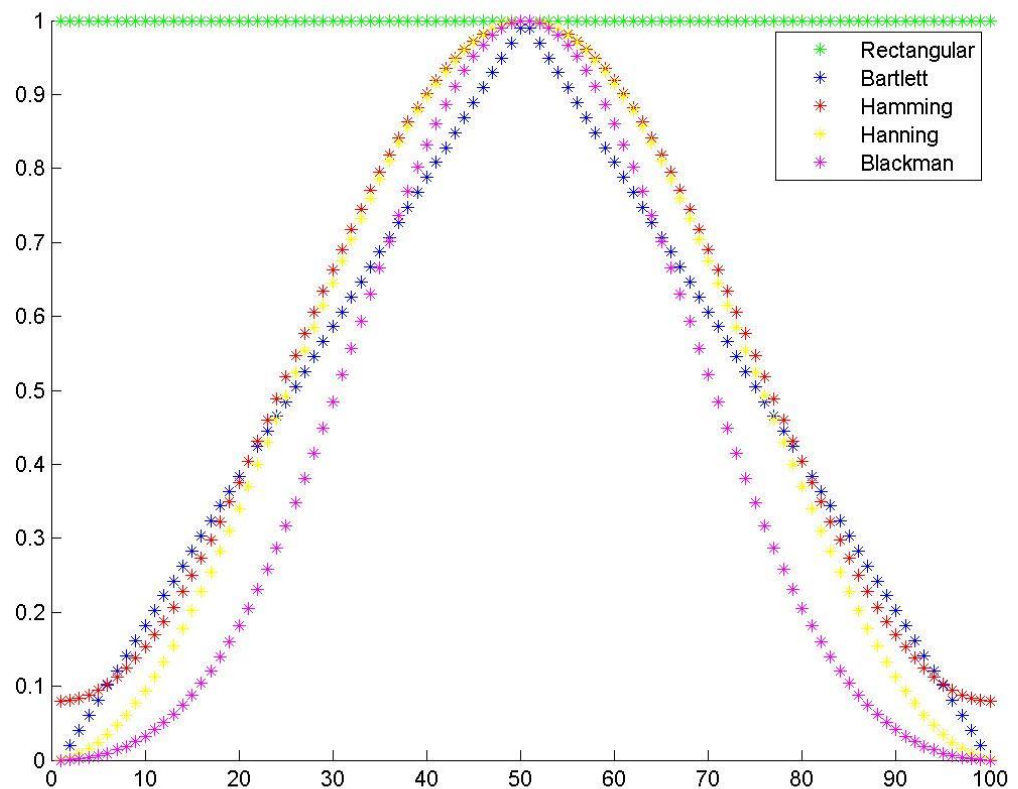


**Fig. 3.4 Representación de los principales tipos de ventana con un tamaño de 100 muestras.**

La resolución espectral de una ventana se rige por la siguiente fórmula:

$$R.E \text{ Ventana (Hz)} = \frac{Fs}{N} \quad (3.4)$$

Esta resolución, que se mide en Hz, debe ser menor que la mínima distancia entre los distintos tonos que compongan la señal enventanada. Por ejemplo, si nuestra resolución fuese de 15.625Hz (teniendo una frecuencia de muestreo de 16 kHz y un tamaño de ventana de 1024 muestras) y tuviésemos dos tonos situados en 10 y 20 Hz respectivamente nuestra resolución no alcanzaría para identificarlos como dos tonos diferentes, sino que serían considerados como un único tono con energía en ambas frecuencias.

Lo que pretende el enventanado es obtener la información en frecuencia de la señal para cada instante de tiempo, es decir, saber que frecuencias están activas en cada momento. Esta información no puede obtenerse realizando una simple Transformada de Fourier ya que esta nos indica cómo se reparte la energía de la señal en todas las frecuencias pero sin indicarnos en que momento una determinada frecuencia presenta o no energía.

### 3.3. Extracción de la pista rítmica

El principal propósito de este documento consiste en la extracción de la pista rítmica de señales musicales. Para conocer en qué estado se encuentra esta técnica a día de hoy ha sido necesaria una revisión bibliográfica sobre esta temática.

Los autores Masataka Goto y Yoichi Muraoka han indagado en profundidad en torno a estos temas y han realizado una extensa variedad de publicaciones sobre la extracción del tiempo de Beat, ritmo en español, (momentos en los que se producen los eventos que llevan el ritmo de la canción), seguimiento del Beat (para la obtención de la línea de percusivos) así como la obtención de la línea de bajos.

La información que se obtiene a partir de la obtención de estos tiempos puede ser aprovechada para la obtención de la pista rítmica, por ello vamos a presentar algunos trabajos cuyo objetivo es el seguimiento del tiempo de beat (Beat Tracking). En los siguientes párrafos describimos algunos de sus trabajos en esta área [19], [20], [21], [22], [23].

El trabajo llevado a cabo en estos documentos tiene como principal objetivo el seguimiento del beat (ritmo) de la canción y, una vez obtenidos los tiempos en los que se produce dicho beat, tratar de extraer la pista percusiva de la señal. Esto es importante ya que tiene aplicaciones en muchos ámbitos como el multimedia o el control de luces en conciertos al ritmo de la música.

El primer documento que se debe analizar [20], trata de explicar el entendimiento de la música desde un punto de vista del ritmo, esto permite adquirir importantes conocimientos que serán de gran valor a la hora de entender cómo se forma el ritmo y sus principales características. En este, se implementa también un sistema de seguimiento del ritmo cuyas principales características e innovaciones frente a documentos previos son la utilización de diferentes hipótesis que entraran en juego en función del escenario en el que nos encontremos, estas hipótesis buscan la obtención del tiempo de beat. Otra característica que marca este sistema consiste en como de ciertas pueden ser dichas hipótesis en función del escenario en el que nos encontremos, por lo que en caso de darse resultados ilógicos estos se desechan. Por último, el sistema realiza también un análisis de los parámetros frecuenciales y un ajuste dinámico de estos.

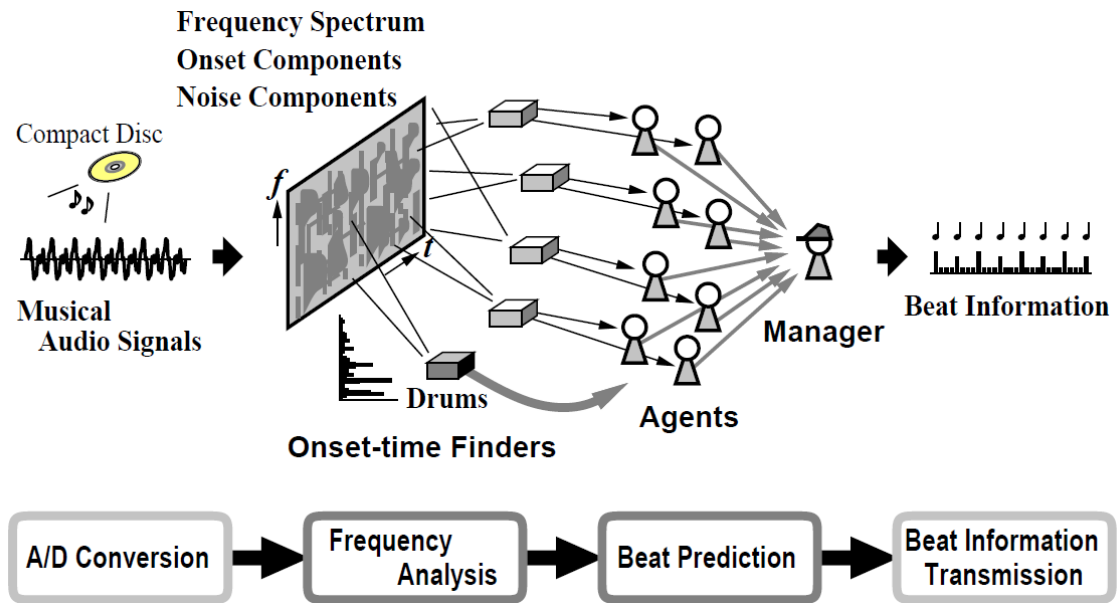


Fig. 3.5 Imagen extraída del documento: Music Understanding At The Beat Level – Real-Time Beat Tracking For Audio Signals.

En un principio, se tenían serios problemas en señales acústicas que contenían sonidos de varios instrumentos. Técnicas como la descrita en [19] partían de múltiples hipótesis en paralelo para realizar la búsqueda del beat. Esto permitía que en caso de no cumplirse algunas de dichas hipótesis, el seguimiento del beat podía continuarse dado que otras de esas hipótesis sí que se seguían cumpliendo.

Técnicas posteriores como [22], añaden a las hipótesis de [19] otras nuevas aprovechando la información que se obtiene de la autocorrelación de la señal así como de la correlación cruzada con señales de pulsos que simulan los tiempos de onset. La búsqueda de estos tiempos de onset es clave en la extracción del ritmo de la señal dado que, como ya se ha comentado en este documento, el ritmo va asociado a instrumentos percusivos y la tendencia natural de estos es la repetitividad a corto plazo (sección 1.4).

En [23], los autores van más allá de lo estudiado hasta el momento y proponen un interesante trabajo sobre la obtención de la pista rítmica, añadiendo la novedad de que tratan de extraer esta pista tanto de señales que poseen sonidos de batería (Drum), como de señales que no poseen este tipo de pistas en su composición. Esto dificulta en gran medida el trabajo a realizar puesto que no se puede buscar el ritmo basándose en las hipótesis de los anteriores trabajos. El sistema implementado funciona en tiempo real y reconoce la estructura seguida por el ritmo aprovechando otras características de la música como los tiempos de onset, cambios en los acordes (chords) y, en caso de existir, patrones de batería y percusivos.

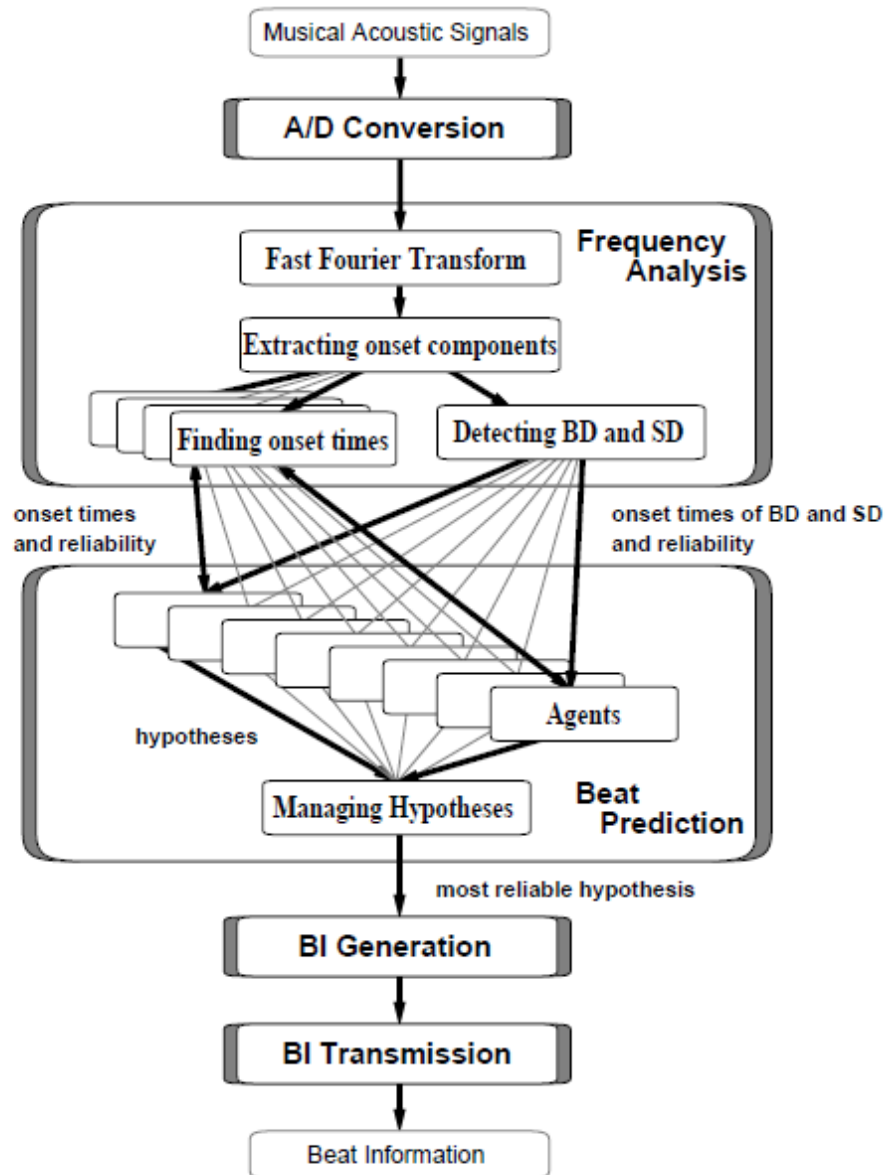


Fig. 3.6 Imagen extraída del documento: A Beat Tracking System for Acoustic Signals of Music.

Siguiendo esta línea de trabajo sobre extracción de pista rítmica en señales sin sonidos de batería nos encontramos el documento [24]. En este, se describe un sistema de seguimiento del tiempo de beat en tiempo real que nos permite obtener la estructura rítmica de una señal en ausencia de sonidos percusivos. Para ello se basa en la detección de cambios en los acordes (chords), gracias a esto, este sistema puede reconocer diferentes patrones rítmicos, por ejemplo, es capaz de encontrar el comienzo (onset) de las notas, así como seleccionar las mejores hipótesis que permiten obtener la posición del ritmo a lo largo del tiempo.

A lo largo de esta sección se ha buscado la obtención de un patrón de repetitividad que nos marque el ritmo seguido por la señal. El beat (ritmo en inglés) viene generalmente marcado por los sonidos producidos por instrumentos percusivos y suele presentar un alto índice de repetitividad, lo que implica una repetitividad a corto plazo (Hipótesis sección 1.1).

Conociendo esto, se pretende obtener lo que se conoce como Beat Spectrum gracias al cual se deduce el “Tiempo de Beat” que nos indica los tiempos en los que se produce un evento que marca el ritmo de la señal. Para llevar a cabo esta tarea se dispone a día de hoy de ciertos algoritmos ya implementados, en concreto en este documento nos centramos en el estudio de los algoritmos REPET [17] y ELLIS [27] que son algoritmos competidores de aquellos implementados por Goto y Muraoka y que describimos a continuación.

**REPET: REpeating Pattern Extraction Technique** es un algoritmo que viene descrito en el documento: A Simple Music/Voice Separation Method based on the extraction of the Underlying Repeating Structure [17]. En este se indica el procedimiento a seguir para la obtención de una determinada marca de tiempo. A grandes rasgos, las pautas seguidas por REPET son:

- En primer lugar se calcula un “correlograma” de la señal. Esto se hace calculando la autocorrelación de las filas del espectrograma en potencia de la señal.

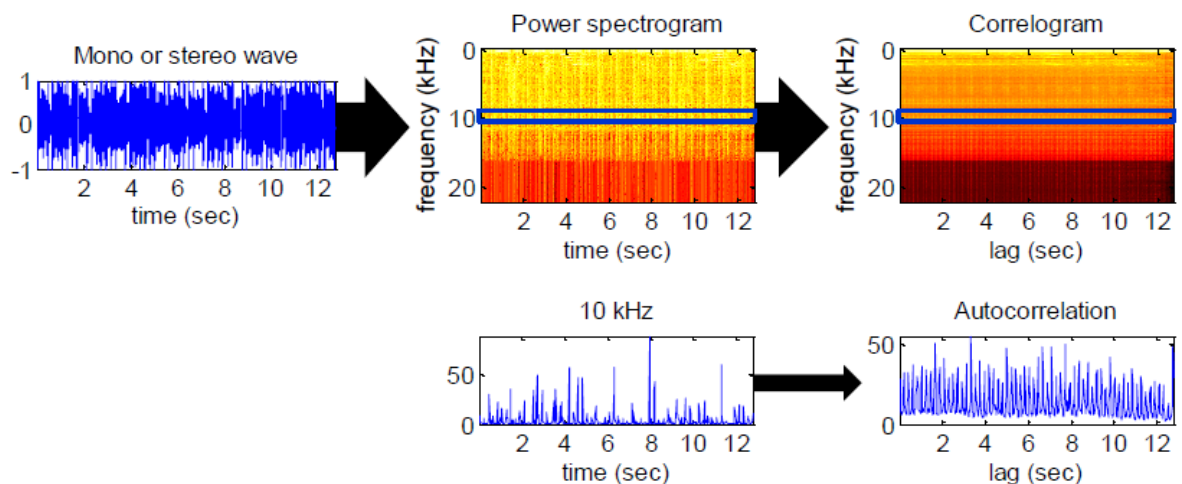


**Fig. 3.7** Imagen extraída del documento: A Simple Music/Voice Separation Method base don the extraction of the Underlying Repeating Structure.

- A continuación, se toma la media de las filas del “correlograma” obtenido, de este modo se obtiene lo que se conoce como Beat Spectrum.

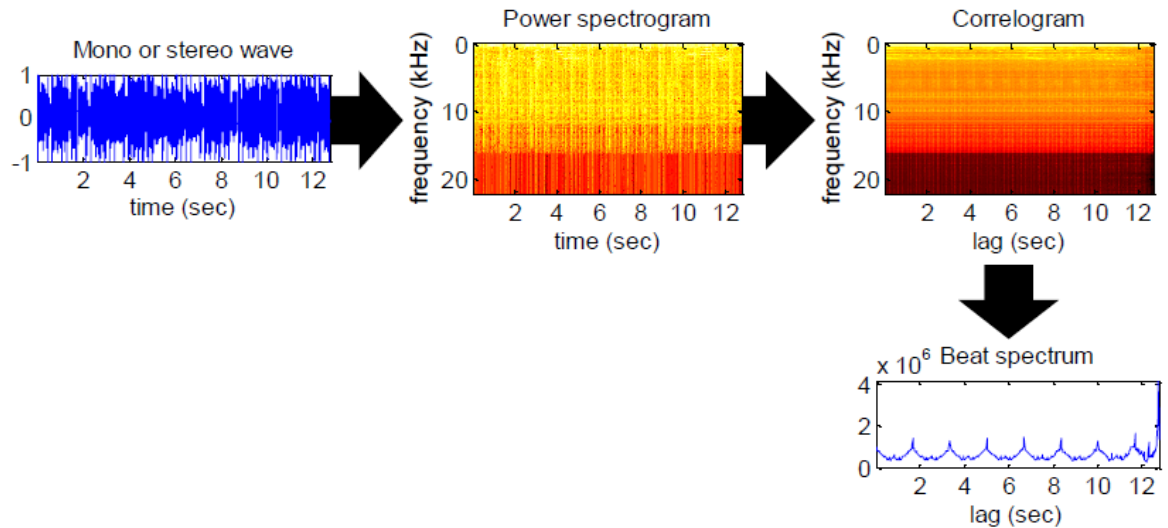


**Fig. 3.8** Imagen extraída del documento: A Simple Music/Voice Separation Method base don the extraction of the Underlying Repeating Structure.

- Este Beat Spectrum nos indica el periodo de repetición (P) de la estructura musical analizada.

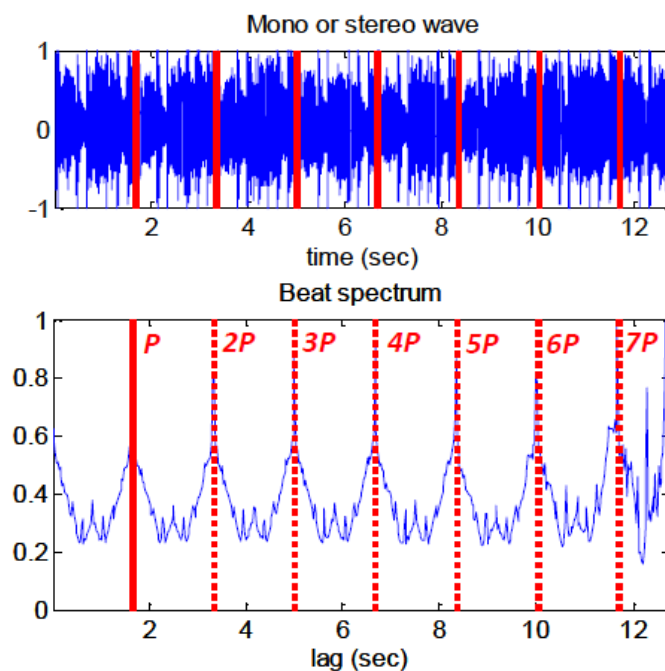


**Fig. 3.9** Imagen extraída del documento: A Simple Music/Voice Separation Method base don the extraction of the Underlying Repeating Structure.

El segundo algoritmo seguido para la obtención de los tiempos de Beat es descrito en el documento Beat Tracking by Dynamic Programming (ELLIS) [27]. Este algoritmo es generalmente conocido como **ELLIS**. Gracias a este nuevo algoritmo se nos facilitan unas determinadas marcas de tiempo que corresponden con el ritmo de la canción. El funcionamiento general de este algoritmo es el siguiente:

- En primer lugar se estima un tempo global del ritmo de la señal.
- Utilizando este valor de tempo, se construye una función de coste de transición.
- Por último, mediante el uso de programación dinámica (DTW) se busca el mejor conjunto de tiempos posible que refleje el tempo y, a su vez, coincida con aquellos momentos en los que se producen determinados onset.

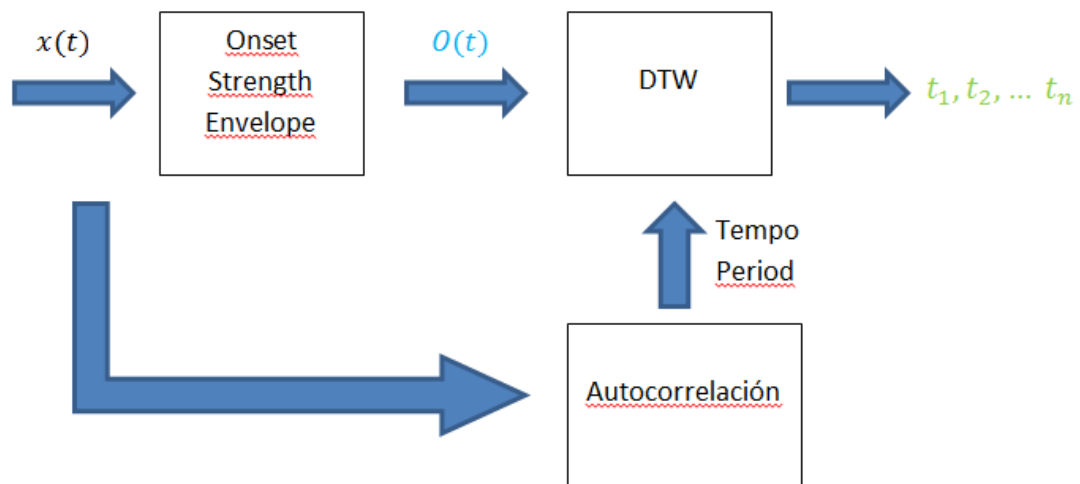


Fig. 3.10 Diagrama de bloques del funcionamiento del algoritmo ELLIS

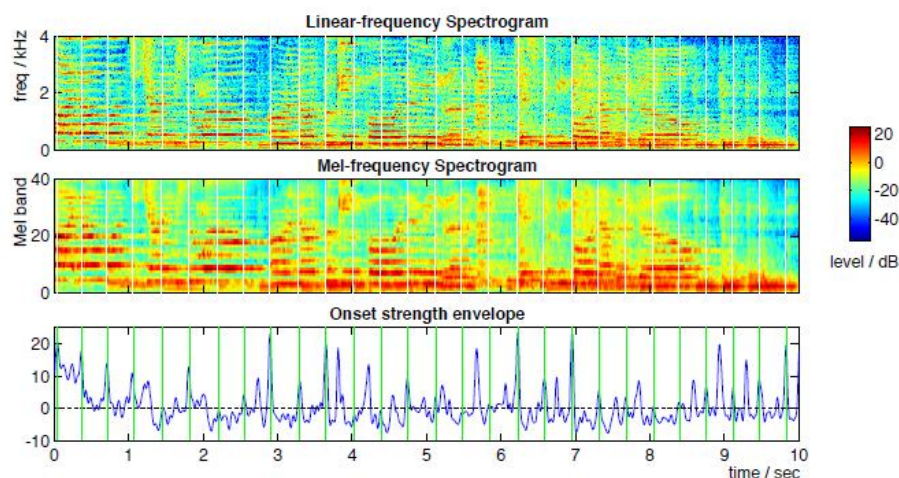


Fig. 3.11 Imagen extraída de: Beat Tracking by Dynamic Programming.

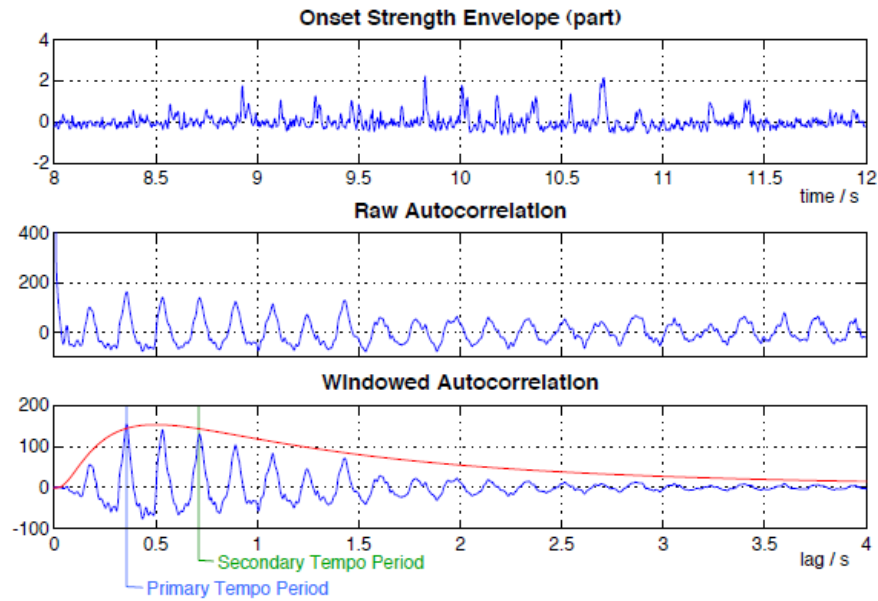


**Fig. 3.12** Imagen extraída de: *Beat Tracking by Dynamic Programming*.

Aunque a primera vista pueda parecer que los dos algoritmos desempeñan la misma función, lo cierto es que los datos obtenidos por cada uno de ellos son muy diferentes y se utilizarán a lo largo de este documento para las diferentes implementaciones de sistemas descritas más adelante.

Así, el algoritmo REPET devuelve una única marca de tiempo (P) y asume que la señal presenta un patrón de repetitividad constante a lo largo del tiempo. Esto quiere decir que para REPET los tiempos de beat que marcan el ritmo se producen en instantes de tiempo múltiplos del valor P obtenido. Esta suposición no es del todo cierta dado que, aunque el ritmo suele repetirse con un periodo casi constante, el ser humano es incapaz de generar un ritmo con un patrón de repetitividad tan exacto. Es ahí donde entra en juego el segundo método descrito.

El algoritmo facilitado por ELLIS no devuelve una única marca de tiempo como ocurría en el caso de REPET sino que nos indica todos los tiempos en los que se producen eventos que marquen el ritmo de la señal, es decir, devuelve los tiempos de beat de toda la señal. Como indicamos antes, el periodo de repetición del ritmo no es constante sino que presenta pequeñas fluctuaciones a lo largo del tiempo, esto es tenido en cuenta por ELLIS por lo que los tiempos de beat devueltos por este algoritmo no son múltiplos de un valor determinado sino que varían (obteniendo valores mayores o menores) según indica el ritmo.

Así si tomamos como ejemplo una canción en la que el ritmo venga indicado por un golpe de batería que se produce cada 0.5 segundos, el parámetro  $P$  facilitado por REPET será  $P = 0.5 \text{ sg}$  mientras que ELLIS devolvería unas marcas de tiempo de este estilo: [0.5, 0.96, 1.5, 2.04, 2.55, ...] (apréciese la fluctuación con respecto a múltiplos de  $P$ ). También ha de tenerse en cuenta que en el momento en el que el ritmo de la señal varíe (por ejemplo al tener un solo de batería o un cambio en la velocidad de la canción) el algoritmo REPET pierde su utilidad mientras que, si los cambios producidos son leves, ELLIS será capaz de ajustarse y cambiar los tiempos devueltos.

Una última consideración a tener en cuenta en estos algoritmos es que nos permiten la inclusión de unos determinados parámetros que facilitan la selección de los tiempos de beat:

- **REPET** utiliza un umbral  $[Rmin, Rmax]$  que permite seleccionar el máximo de la función de autocorrelación normalizada dentro de esos límites, así si se produce un pico como ocurre en torno al segundo 12 de la figura 3.9 este no es tenido en cuenta.
- **ELLIS** permite utilizar el parámetro *startbpm* con el que indicaremos el número de eventos de ritmo por minuto esperados. Gracias a él podemos controlar que no cualquier evento sea considerado como perteneciente a la línea rítmica

Conocidos los diferentes procedimientos que se siguen en la actualidad para la obtención del tiempo de beat (centrándonos en concreto en los algoritmos REPET y ELLIS) y, gracias a este, saber cuándo se producen eventos que marquen el ritmo de la señal. Una vez conocidos estos tiempos, será necesario llevar a cabo una segmentación del módulo del espectrograma de nuestra señal mezcla ( $X_m(t, f)$ ). Mediante este procedimiento obtendremos unos determinados espectrogramas que denominaremos  $X_i(f, t)$  siendo  $i = 1, 2, 3 \dots N_{segmentos}$

Para reducir el número de matrices con el que se trabaja se ha optado por construir una matriz tridimensional a partir de estos  $X_i(f, t)$  tal y como muestra la siguiente figura:

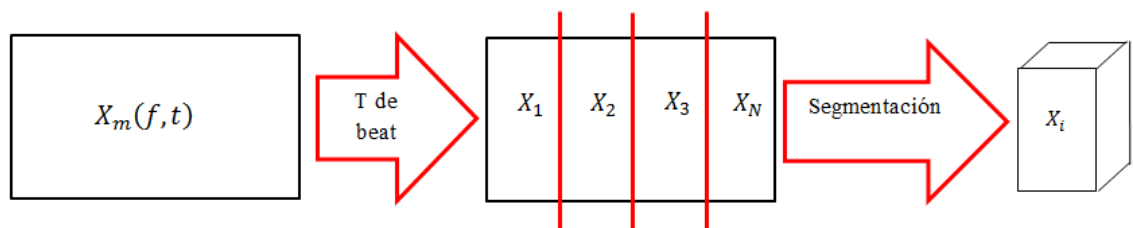


Fig. 3.13 Procedimiento seguido para realizar la segmentación de un espectrograma.

A la hora de realizar la segmentación tendremos que tener en cuenta el algoritmo que utilicemos ya que las marcas de tiempo aportadas por estos algoritmos son diferentes. Así, tenemos los siguientes casos:

- **REPET:** este es el caso más sencillo dado que únicamente tenemos una marca de tiempo ( $P$ ). La segmentación del espectrograma se realiza cortando cada  $P$  frames de tiempo ( $P$  columnas de la matriz). La única particularidad que ha de tenerse en cuenta es que en el último segmento puede darse el caso de no tener suficientes frames como para tener un segmento completo. En ese caso ha de rellenarse con ceros hasta que el último segmento tenga  $P$  frames.
- **ELLIS:** usando este algoritmo la segmentación se complica ligeramente aunque es un proceso sencillo también. En primer lugar y conocidas todas las marcas de tiempo ( $t_i$ ), será necesario buscar la mayor diferencia entre marcas de tiempo ( $D_{MAX}$ ), es decir, hemos de buscar las dos marcas de tiempo consecutivas que nos indiquen una mayor separación en lo que a tiempo se refiere. Esto es importante ya que los distintos segmentos tendrán un tamaño variable al ser estas marcas también variables y mediante este paso inicial conseguimos obtener el tamaño del mayor de los segmentos (el que mayor número de frames posee). Una vez realizado esto, lo único que queda sería obtener los segmentos según indica  $t_i$  y rellenar con ceros hasta alcanzar el tamaño que indica  $D_{MAX}$ . Conseguido esto, tendremos segmentos del mismo tamaño y podremos colocarlos en la estructura tridimensional de la imagen.

### 3.4. NMF

Para la realización de este proyecto ha sido necesario el estudio previo de bibliografía y demás documentos científicos que guardan relación con el problema que tratamos de solventar. A día de hoy, buena parte de los algoritmos de separación de fuentes se basan en el *Análisis de Componentes Independientes* (*Independent Component Analysis, ICA*) y la *Factorización de matrices no negativas* (*Non-negative Matrix Factorization*) como principales herramientas para obtener unas determinadas funciones base, estas se seleccionarán en función del objetivo que se pretenda alcanzar. En concreto en esta sección describimos los algoritmos NMF ya que será este tipo de algoritmo con el que vamos a trabajar a lo largo de este documento.

NMF consiste en un conjunto de algoritmos cuyo propósito es realizar una separación de una matriz de entrada ( $X$ ) en dos matrices de salida ( $W$  y  $H$ ). Estas matrices cumplen la propiedad de que al ser multiplicadas entre sí, obtenemos la matriz de entrada.

$$X \approx W * H \quad (3.5)$$

Mediante algoritmos de cálculo de distancias, estas matrices tratan de asemejarse a la matriz original respetando siempre que las componentes de las matrices  $W$  y  $H$  sean mayores que cero, es decir, no negativas.

El significado de las matrices  $W$  y  $H$  sería el siguiente:

- **Matriz de componentes ( $W$ ):** esta matriz contiene las diferentes componentes de la señal, también conocidas como bases. Idealmente cada una de estas bases coincidiría con un determinado instrumento de la señal. Así, esta matriz contiene la información sobre amplitud y frecuencia de cada base.
- **Matriz de activaciones ( $H$ ):** esta matriz nos aporta información sobre en qué momento se producen cada uno de los determinados eventos (información temporal). Mientras que la matriz de componentes posee información sobre todos los eventos, es  $H$  quien nos indica el momento exacto en el que este se produce.

Podríamos dividir NMF en dos fases:

- **Fase de entrenamiento:** aquí se deben entrenar los patrones espectrales de aquellos sonidos que forman la señal original que se desea separar.
- **Fase de separación:** en esta fase llevaremos a cabo un determinado número de iteraciones hasta conseguir que la distancia (diferencia) entre la matriz  $X$  y el producto  $W*H$  sea mínimo.

Existen tres tipos de algoritmos NMF en función del uso o no de fase de entrenamiento:

- **Sistema supervisado:** durante la fase de entrenamiento obtendremos las matrices de componentes ( $W$ ) ya que entrenamos los patrones espectrales de ambos sonidos que queremos separar. Esto implica que durante la fase de separación únicamente debemos calcular las matrices de activaciones ( $H$ ) de los sonidos a separar.
- **Sistema semisupervisado:** este tipo de sistemas también presenta las dos fases descritas, pero a diferencia del sistema supervisado únicamente entrenamos el patrón espectral de uno de los sonidos a separar. Esto implica que durante la fase de separación tendremos fijada una de las matrices de componentes (la del sonido entrenado) y debemos obtener la matriz de componentes del sonido que no ha sido entrenado, así como las matrices de activaciones de ambos sonidos.
- **Sistema no supervisado (ciego):** estos sistemas no presentan fase de entrenamiento (el sistema no conoce ningún patrón espectral) y debe realizar la separación por medio de otras restricciones (sparse y smoothness).

En este documento se trabajará con un sistema ciego, lo cual implica que no realizaremos ningún tipo de entrenamiento, esto supone una mayor dificultad a la hora de realizar la separación de fuentes acústicas ya que no disponemos de ningún tipo de patrón espectral de ayuda.

Una vez realizada nuestra conversión del dominio temporal, señal  $x(t)$ , al dominio frecuencial, obtendremos el espectrograma  $X(t, f)$  que será complejo. Llamaremos  $X_m(t, f)$  al módulo de este espectrograma complejo el cual tiene dimensiones  $F*T$ . Una vez acondicionada nuestra señal y trabajando en el dominio frecuencial, será necesario realizar el algoritmo NMF en algunas de las implementaciones de sistemas descritos en el próximo capítulo. Sobre el espectrograma del módulo de nuestra señal ( $X_m(t, f)$ ) realizamos el algoritmo NMF con un número de bases ( $K$ ). La selección del número de bases es un tema delicado ya que el escoger un  $K$  pequeño supone que una base deba modelar más de un sonido, por el contrario, el hecho de coger un número elevado de bases ( $K$ )

implicaría que serían necesarias varias bases para poder modelar un único sonido y por tanto su posterior selección se ve dificultada.

Recordemos que el objetivo de este proyecto es la extracción de la pista rítmica y por tanto necesitaremos seleccionar aquellas bases que modelen instrumentos percusivos así como bajo o sonidos que se repitan a lo largo de toda la señal analizada. A continuación detallamos en profundidad el procedimiento seguido para llevar a cabo la técnica NMF.

Para cumplir la ecuación  $X = W * H$  las matrices  $W$  y  $H$  deben presentar dimensiones  $F*K$  y  $K*T$  respectivamente.

Estas dimensiones indican lo siguiente:

- **F (número de filas de X y W):** número de bins utilizados. Coincide con  $N/2 + 1$  (siendo  $N$  el número de muestras de la ventana utilizada).
- **T (número de columnas de X y H):** número de frames de tiempo que tenemos.
- **K (número de columnas de W y de filas de H):** este dato es el número de componentes usadas para llevar a cabo la reconstrucción. A estas componentes también se les denomina bases y representan cada uno de los sonidos de nuestra señal. Cada patrón espectral puede ser reconstruido con un determinado número de bases ( $K$ ) que no tiene por qué coincidir en los distintos sonidos.

Conocidas tanto la entrada como las salidas de nuestro sistema, describimos el procedimiento que habrá de llevarse a cabo, recordemos antes que estamos trabajando con un **sistema ciego** y que por tanto no será necesaria una fase de entrenamiento de nuestro sistema por lo que nos ceñiremos a describir exclusivamente la fase de separación.

Sabemos que las funciones de coste con las que trabajamos son siempre positivas y cumplen que únicamente toman el valor cero cuando  $X = Y$ . Nuestra intención es conseguir que esta función de coste sea constante y de un valor mínimo lo cual nos indicará que  $X$  se parece enormemente a  $Y$  (siendo  $Y = W * H$ ), es decir que el resultado de multiplicar  $W * H$  es prácticamente idéntico a nuestra matriz de entrada  $X$ .

Se pretende también minimizar el número de iteraciones que debemos hacer mediante la búsqueda de un mínimo local. Una vez alcanzado dicho mínimo podremos decir que tenemos un grado de semejanza entre la señal de entrada y la señal reconstruida lo suficientemente bueno.

Durante la fase de separación introducimos la señal que contiene todos los sonidos mezclados  $X_m(t, f)$ . Tratamos de separar los sonidos repetitivos del resto de la señal, obteniendo de este modo la pista rítmica en un archivo y por otro el resto de sonidos que formen la señal como pueden ser voz, guitarra, piano, etc.

Partimos de la suposición de que las señales son aditivas, esto implica que tanto en el dominio frecuencial como en el temporal pueden ser sumadas por lo que tendríamos:

$$X_m = X_r + X_{nr}$$

Siendo  $X_m$  el módulo del espectrograma de la señal mezcla,  $X_r$  el módulo del espectrograma de la pista rítmica y  $X_{nr}$  el módulo del espectrograma de la señal que contiene el resto de sonidos (pista no rítmica).

Si desarrollamos partiendo de nuestra ecuación  $X = W * H$  obtenemos la siguiente descomposición:

$$X_m \approx X_r + X_{nr} \approx (W_r * H_r) + (W_{nr} * H_{nr}) \quad (3.6)$$

Teniendo en cuenta que  $W_r$  y  $W_{nr}$  son las matrices de componentes que recogen las bases de la pista rítmica y no rítmica respectivamente mientras que  $H_r$  y  $H_{nr}$  son las matrices de activaciones que indican cuando se produce cada evento.

El cometido de la fase de separación es, partiendo de unos  $W_r, W_{nr}, H_r$  y  $H_{nr}$  inicializados como matrices de números aleatorios (no negativos), llevar a cabo un determinado número de iteraciones hasta que la función de coste (divergencia) elegida llegue a un mínimo local, momento en el cual se cumplirá que  $X_m \approx X_r + X_{nr}$ .

Las funciones de coste utilizadas son las descritas a continuación:

- Distancia Euclídea:

$$d_{EUC} = \frac{1}{2} (X_m - Y)^2 \quad (3.7)$$

- Kullback Leibler:

$$d_{KL} = X_m * \log \frac{X_m}{Y} - X_m + Y \quad (3.8)$$

○ Itakura Saito:

$$d_{IS} = \frac{X_m}{Y} - \log \frac{X_m}{Y} - 1 \quad (3.9)$$

Nota:  $X_m$  se correspondería con la señal mezclada que se utiliza como entrada al sistema mientras que  $Y$  se corresponde con la señal obtenida a partir de la reconstrucción.

Llegado a un determinado número de iteraciones, la función de coste converge, esto quiere decir que se hará constante (asintótica sobre el eje de ordenadas). Llegado este punto el algoritmo puede parar de iterar ya que consideramos suficientemente parecidas la señal de entrada con la señal reconstruida. Habrá finalizado la fase de separación.

Conocido el procedimiento general que se debe seguir y el momento en el que nuestro algoritmo ha de detenerse, describimos a continuación, y diferenciando para cada una de las funciones de coste, el desarrollo que se sigue para obtener las matrices  $W$  y  $H$ .

El modo de obtención de las ecuaciones de actualización pasa por realizar unas derivadas parciales a la función de coste. Tras esto, los términos negativos de dicha derivada se sitúan en el numerador mientras que aquellos términos que sean positivos se sitúan en el denominador de la siguiente forma:

$$W = W .* \frac{\left[ \frac{\partial D}{\partial W} \right]^-}{\left[ \frac{\partial D}{\partial W} \right]^+} \quad (3.10)$$

$$H = H .* \frac{\left[ \frac{\partial D}{\partial H} \right]^-}{\left[ \frac{\partial D}{\partial H} \right]^+} \quad (3.11)$$

- **Función de coste Distancia Euclídea:**

La conocida ecuación de divergencia de esta función es la siguiente ( $D_{EUC}$ ):

$$D_{EUC} = \frac{1}{2}(X - Y)^2$$

Conociendo que  $Y = W * H$  (salida del sistema) y que  $X$  es considerado como la entrada del sistema.

$$D_{EUC} = \frac{1}{2}(X - WH)^2 = \frac{1}{2}(X^2 + W^2H^2 - 2XWH) \quad (3.12)$$

$$\begin{aligned} \frac{\partial D_{EUC}}{\partial W} &= \frac{1}{2}(2WH^2 - 2XH) = WH^2 - XH \\ W &= W .* \frac{XH^T}{WH^2} = W .* \frac{XH^T}{WHH^2} \end{aligned} \quad (3.13)$$

$$\begin{aligned} \frac{\partial D_{EUC}}{\partial H} &= \frac{1}{2}(2HW^2 - 2XW) = HW^2 - XW \\ H &= H .* \frac{XW^T}{HW^2} = H .* \frac{W^T X}{W^T WH} \end{aligned} \quad (3.14)$$

Nota: los cocientes de  $W$  y  $H$  son operaciones elemento a elemento.

- **Función de coste Kullback Leibler:**

Planteamos el desarrollo desde la ecuación de divergencia ( $D_{KL}$ ):

$$D_{KL} = X * \log \frac{X}{Y} - X + Y$$

Conociendo que  $Y = W * H$  (salida del sistema) y que  $X$  es considerado como la entrada del sistema.

$$D_{KL} = X * \log \frac{X}{WH} - X + WH \quad (3.15)$$

$$\frac{\partial D_{KL}}{\partial W} = X * \left( \frac{-XH}{Y^2} \right) + H = -\frac{XH}{Y} + H \quad (3.16)$$

$$\begin{aligned}
 W &= W .* \frac{XHY^{-1}}{H} = W .* \frac{(X .* Y^{-1}) * H^T}{1 * H^T} \\
 &= W .* \frac{(X .* (W * H)^{-1}) * H^T}{1 * H^T} \\
 \\
 \frac{\partial D_{KL}}{\partial H} &= X * \left( \frac{\frac{-XW}{Y^2}}{\frac{X}{Y}} \right) + W = -\frac{XW}{Y} + W \\
 \\
 H &= H .* \frac{XWY^{-1}}{W} = H .* \frac{W^T * (X .* Y^{-1})}{W^T * 1} \\
 &= H .* \frac{W^T * (X .* (W * H)^{-1})}{W^T * 1} \tag{3.17}
 \end{aligned}$$

Nota: los cocientes de  $W$  y  $H$  son operaciones elemento a elemento y el número 1 que aparece en el desarrollo previo consiste en una matriz de unos (todos sus elementos igual a uno) y de dimensiones (F, T).

- **Función de coste Itakura Saito:**

Para el último método de cálculo partimos también desde su ecuación de divergencia ( $D_{IS}$ ):

$$D_{IS} = \frac{X}{Y} - \log \frac{X}{Y} - 1$$

Conociendo que  $Y = W * H$  (salida del sistema) y que  $X$  es considerado como la entrada del sistema.

$$D_{IS} = \frac{X}{WH} - \log \frac{X}{WH} - 1 \tag{3.18}$$

$$\begin{aligned}
 \frac{\partial D_{IS}}{\partial W} &= \frac{-XH}{Y^2} - \frac{\frac{-XH}{Y^2}}{\frac{X}{Y}} = \frac{-XH}{Y^2} - \frac{-XH}{YX} = \frac{-XH}{Y^2} + \frac{H}{Y} \\
 W &= W .* \frac{XH(Y)^{-2}}{HY^{-1}} = W .* \frac{(X .* (Y)^{-2}) * H^T}{Y^{-1} * H^T} \\
 &= W .* \frac{(X .* (W * H)^{-2}) * H^T}{(W * H)^{-1} * H^T}
 \end{aligned} \tag{3.19}$$

$$\begin{aligned}
 \frac{\partial D_{IS}}{\partial H} &= \frac{-XW}{Y^2} - \frac{\frac{-XW}{Y^2}}{\frac{X}{Y}} = \frac{-XW}{Y^2} - \frac{-XW}{YX} = \frac{-XW}{Y^2} + \frac{W}{Y} \\
 H &= H .* \frac{XW(Y)^{-2}}{WY^{-1}} = H .* \frac{W^T * (X .* (Y)^{-2})}{W^T * Y^{-1}} \\
 &= H .* \frac{W^T * (X .* (W * H)^{-2})}{W^T * (W * H)^{-1}}
 \end{aligned} \tag{3.20}$$

Nota: los cocientes de  $W$  y  $H$  son operaciones elemento a elemento.

Como generalización de los tres desarrollos anteriores podemos utilizar las siguientes ecuaciones de actualización. En estas simplemente variando el parámetro  $\beta$  tendremos cada una de las tres ecuaciones anteriores. Los posibles valores de  $\beta$  para los distintos algoritmos son:

- $\beta = 2$  para Distancia Euclídea.
- $\beta = 1$  para Kullback Leibler.
- $\beta = 0$  para Itakura Saito.

$$H = H .* \frac{W^T * (W * H)^{(\beta-2)} * X}{W^T * (W * H)^{(\beta-1)}} \tag{3.21}$$

$$W = W .* \frac{(W * H)^{(\beta-2)} * X * H^T}{(W * H)^{(\beta-1)} * H^T} \tag{3.22}$$

Del mismo modo que podemos generalizar las ecuaciones de actualización de  $W$  y  $H$  para que se generen en función del parámetro  $\beta$ , también podemos hacer esto con las funciones de coste. A esta función se la conoce como  $\beta$ -divergencia y consiste en:

$$\beta - \text{divergencia} = \begin{cases} \frac{1}{\beta(\beta - 1)}(X^\beta + (\beta - 1)Y^\beta - \beta XY^{\beta-1}), & \beta \in \mathbb{R} \setminus \{0, 1\} \\ \frac{1}{2}(X - Y)^2, & \beta = 2 \\ X * \log \frac{X}{Y} - X + Y, & \beta = 1 \\ \frac{X}{Y} - \log \frac{X}{Y} - 1, & \beta = 0 \end{cases}$$

Visto todo esto, ya se conoce en profundidad el funcionamiento del algoritmo NMF por lo que de aquí en adelante únicamente se mencionará y en caso de aplicarse algún tipo de modificación esta será descrita en profundidad.

Estas ecuaciones de divergencia, también denominadas funciones de coste, son siempre positivas y tienden a tomar el valor 0 si y sólo si  $x = y$ . La siguiente imagen muestra las funciones de coste de distancia Euclídea, Kullback Leibler e Itakura Saito para  $x = 1$ .

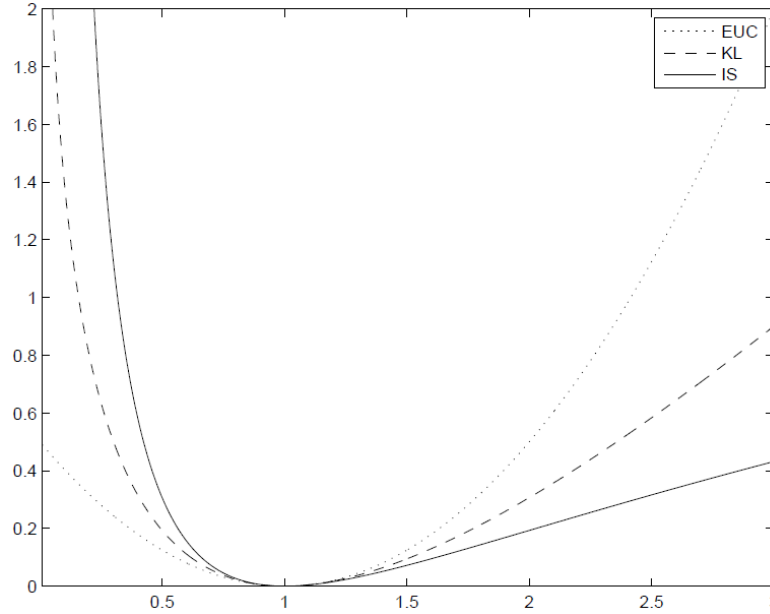


Fig. 3.14 Funciones de coste de distancia Euclídea, Kullback Leibler e Itakura Saito para  $x = 1$

Como muestra de este procedimiento vamos a realizar un sencillo ejemplo en el que descompondremos una señal de guitarra en sus matrices  $W$  y  $H$  y volveremos a obtener la señal de partida. La señal utilizada es la siguiente:

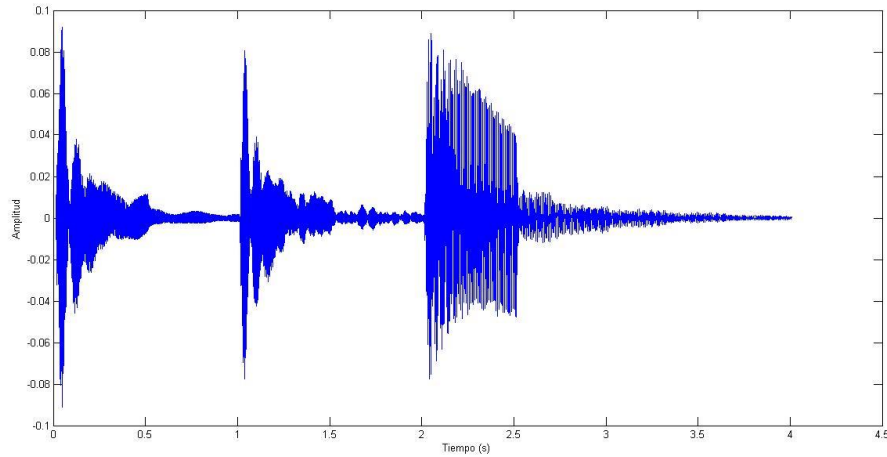


**Fig. 3.15 Señal de muestra en el dominio temporal.**

Esta señal consiste en tres notas consecutivas de frecuencia 172 Hz, 258 Hz y 129 Hz. Vamos a utilizar tres componentes dado que sólo existen tres sonidos ( $K = 3$ ). Estas componentes serán las filas de  $H$  y las columnas de  $W$ . El número de iteraciones realizado dependerá del tipo de señal, para este caso con un número reducido de iteraciones tendremos suficiente ya que trabajamos con una señal muy simple. Este número viene dado por la función de divergencia (coste), cuando esta haya convergido el algoritmo deberá detenerse. La siguiente figura muestra el valor absoluto del espectrograma de la señal de entrada ( $X$ ):

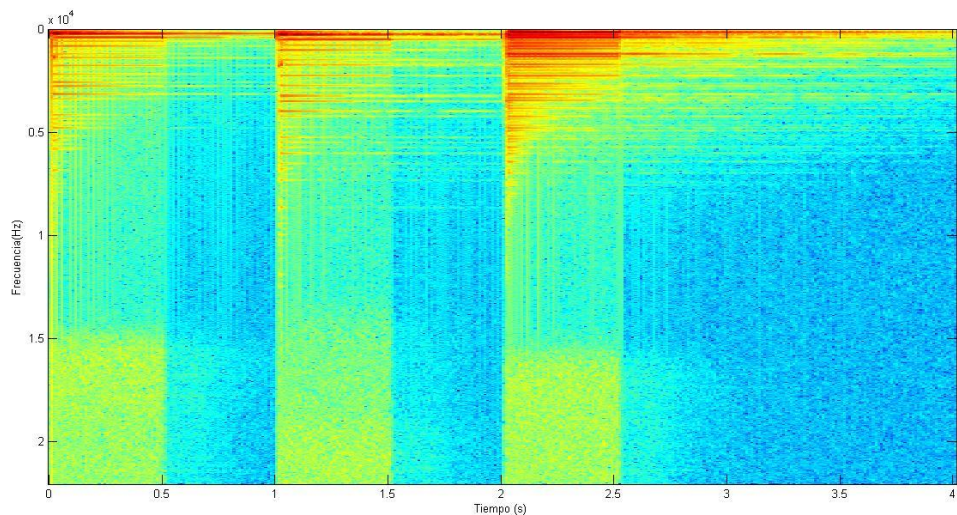


**Fig. 3.16 Espectrograma de la señal X**

Si realizamos un zoom sobre la zona superior de la imagen anterior podemos comprobar con mayor facilidad las frecuencias en las que tenemos una mayor cantidad de energía en cada momento. Tal y como se describió previamente puede verse como gran parte de la energía se concentra en la frecuencia fundamental de la

nota tocada (línea roja) y como el resto de energía se reparte entre los armónicos de dicha nota (líneas menos intensas en frecuencias múltiplos de la frecuencia fundamental).

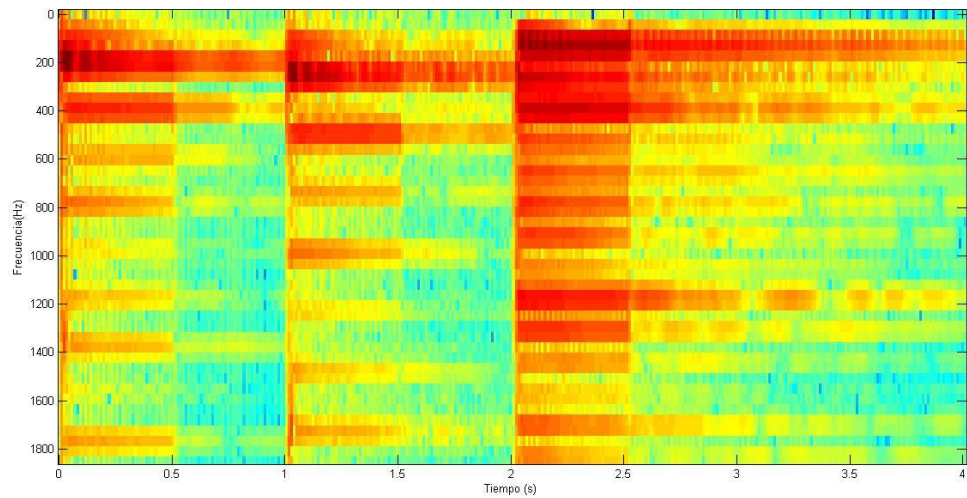


**Fig. 3.17 Zoom sobre la parte superior del espectrograma.**

A continuación procedemos a la ejecución del algoritmo NMF, una vez finalizado este podemos comprobar cómo ha convergido mediante la siguiente imagen:

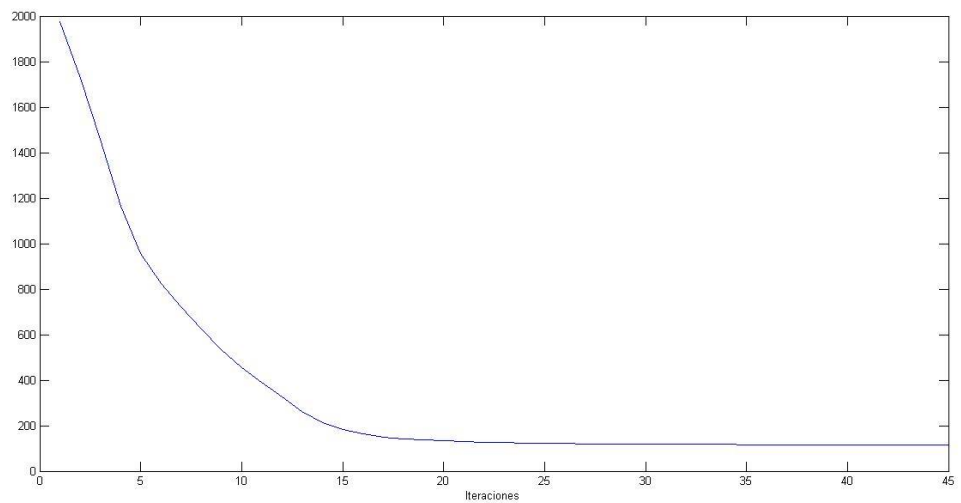


**Fig. 3.18 Representación de la divergencia a lo largo de las iteraciones.**

En ella se aprecia que tras haber realizado 45 iteraciones el algoritmo se ha detenido. Comprobemos los resultados obtenidos. Mostramos tanto la matriz de componentes ( $W$ ), que contiene los patrones de la señal, como la matriz de activaciones ( $H$ ), la cual nos indica en que parte de la señal se activa cada patrón.

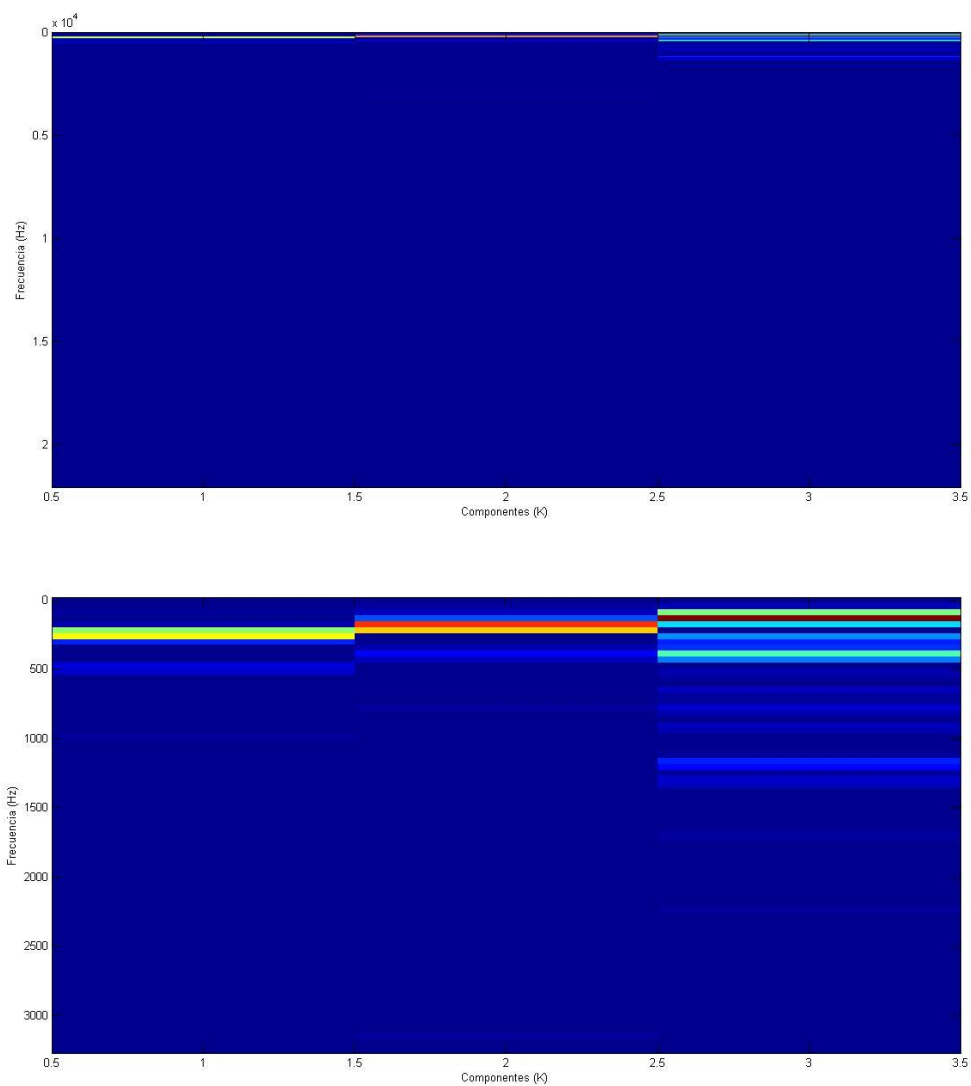


**Fig. 3.19** Matriz de componentes (W) en la imagen superior y zoom de la misma en la imagen inferior.

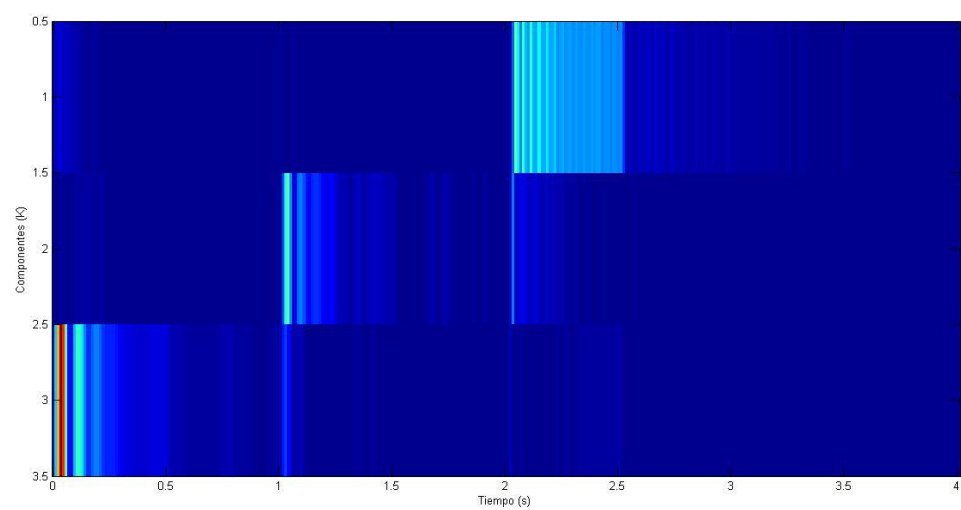
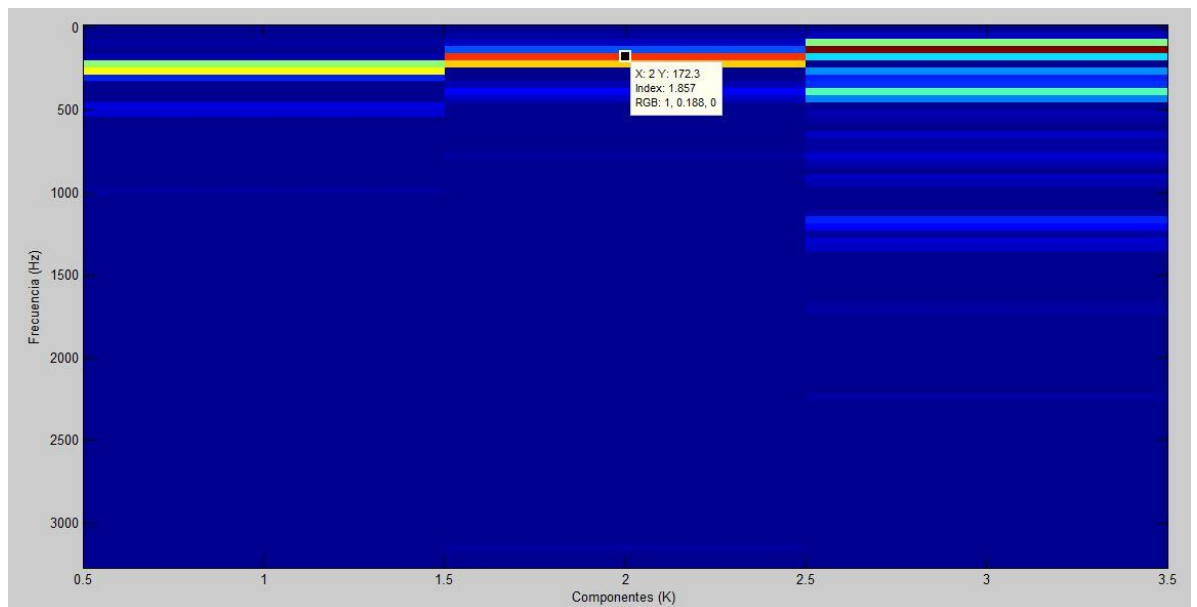
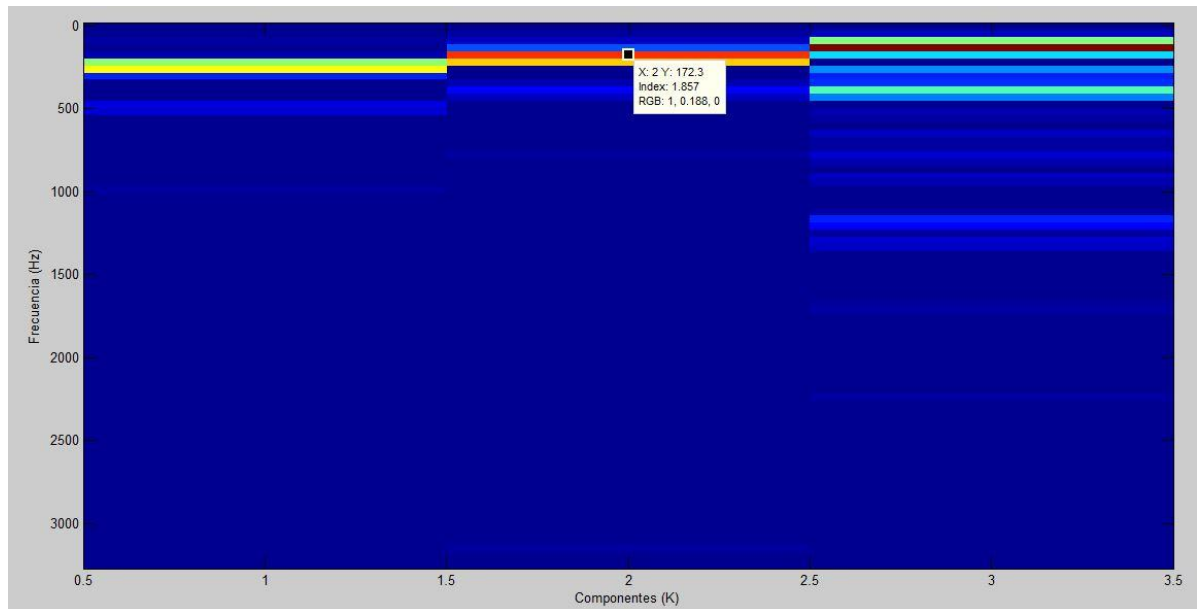


**Fig. 3.20** Matriz de activaciones (H).

Al fijarnos en la matriz de activaciones podemos comprobar fácilmente como la primera nota tocada (172 Hz) se asocia con la segunda componente, la segunda nota (258 Hz) se corresponde con la primera componente y la tercera nota (129 Hz) con la tercera componente. Las siguientes figuras muestran de forma clara el valor de dichas componentes y la frecuencia sobre la que se encuentran:



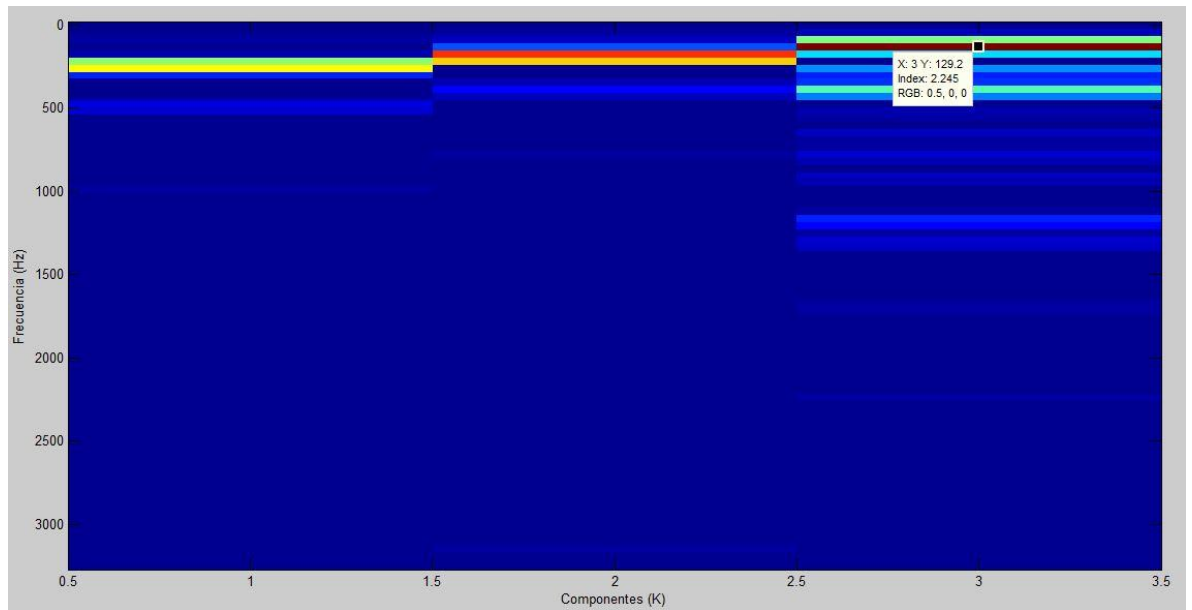


**Fig. 3.21** Detalle sobre la posición en frecuencia que ocupa cada una de las bases en la matriz  $W$ .

Conocidos  $W$  y  $H$ , así como habiendo comprobado que el sistema ha convergido y nuestras componentes se sitúan en torno a la frecuencia fundamental correcta sólo quedaría realizar la multiplicación de  $W$  y  $H$  para comprobar el resultado y asegurar que el espectrograma obtenido se asemeja al de partida:

$$Y = W * H$$

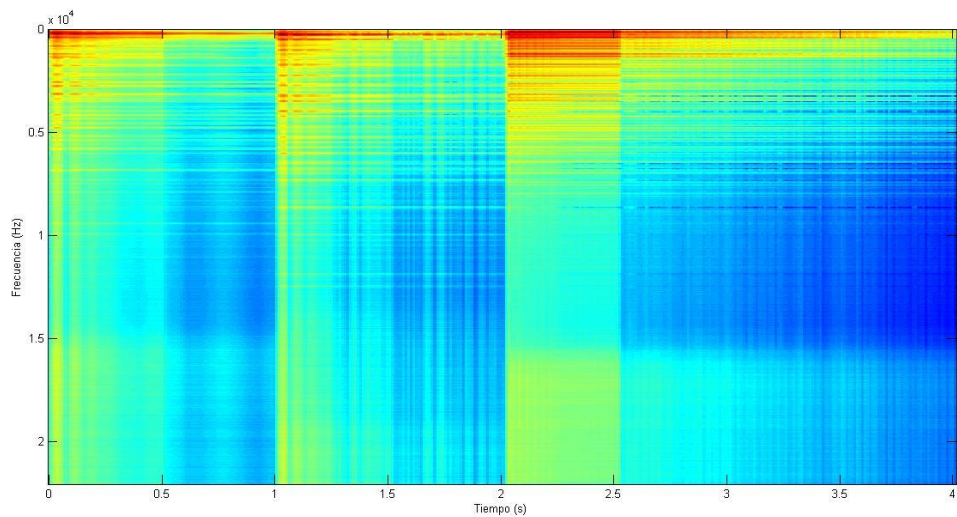


**Fig. 3.22** Espectrograma reconstruido, resultado de  $Y = W * H$

### 3.5. NMPCF

El método descrito en la sección 4.2 consiste en una variación del algoritmo descrito en el artículo *NONNEGATIVE MATRIX PARTIAL CO-FACTORIZATION FOR DRUM SOURCE SEPARATION* [26] por ello será necesario explicar el funcionamiento de este algoritmo.

NMPCF (Non-negative Matrix Partial Co-Factorization) busca realizar el algoritmo NMF pero mediante el uso de una cofactorización. Esto nos permite obtener dos conjuntos de bases diferentes así como dos matrices de activaciones. Lo que se busca mediante esta división de las bases es obtener de forma separada las bases correspondientes a los percusivos de la señal del resto de bases.

El método seguido parte de dos señales de entrada:

- **X**: módulo del espectrograma de la señal completa.
- **Y**: módulo del espectrograma de la señal de percusivos únicamente.

A partir de estas matrices realiza la descomposición mediante unas reglas de actualización y una función de coste determinadas (consultar documentación [26]), al cambiar la función de coste también cambian las diferentes funciones de actualización ya que estas provienen de la derivación de la función de coste. Se describen las matrices que forman parte del algoritmo:

- $U_D$ : matriz de componentes que contiene las bases de los percusivos. Estas bases serán comunes para ambas reconstrucciones.
- $V_D$ : matriz de activaciones correspondiente a las bases percusivas para la reconstrucción de X.
- $U_H$ : matriz de componentes del resto de bases (no percusivas).
- $V_H$ : matriz de activaciones del resto de componentes (no percusivas).
- $W_D$ : matriz de activaciones de las bases de  $U_D$  para la señal Y.

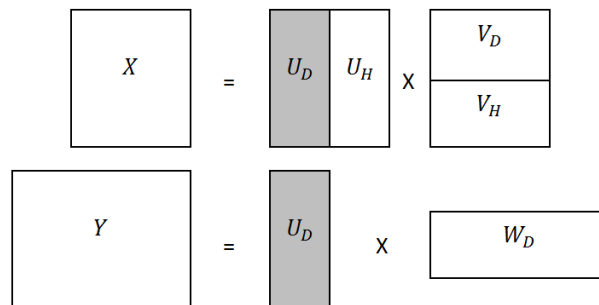


**Fig. 3.23 Ilustración del funcionamiento de NMPCF. Las componentes que contiene la matriz  $U_D$  son comunes para la señal de percusivos como para la señal original, a pesar de tener matrices de activaciones diferentes, ya que ambas señales pueden tener sonidos iguales pero que estos se den en momentos diferentes.**

La función de coste utilizada en dicho documento es la siguiente:

$$L = \frac{1}{2} \|X - U_D V_D - U_H V_H\|^2 + \frac{\lambda}{2} \|Y - U_D W_D\|^2 \quad (3.23)$$

Apréciase que esta función de coste consiste en la de Distancia Euclidea a la que se añade un segundo término ponderado por  $\lambda$ , este parámetro ajustable pondera la importancia (parecido) que tiene la señal de percusivos, Y, con respecto a la señal X.

Las reglas de actualización utilizadas se obtienen de la forma explicada previamente en la descripción de NMF y son las siguientes:

$$U_D = U_D .* \frac{X V_D + \lambda Y W_D}{U_D V_D' V_D + U_H V_H' V_D + \lambda U_D W_D' W_D} \quad (3.24)$$

$$V_D = V_D .* \frac{X' U_D}{V_D U_D' U_D + V_H U_H' U_D} \quad (3.25)$$

$$U_H = U_H .* \frac{X V_H}{U_H V_H' V_H + U_D V_D' V_H} \quad (3.26)$$

$$V_H = V_H .* \frac{X' U_H}{V_H U_H' U_H + V_D U_D' U_H} \quad (3.27)$$

$$W_D = W_D .* \frac{Y' U_D}{W_D U_D' U_D} \quad (3.28)$$

Sabiendo que tiene que cumplirse las condiciones  $X \approx U_D V_D + U_H V_H$  e  $Y \approx U_D W_D$  obtendremos una separación bastante fiel de las partes percusiva ( $X_D$ ) y armónica ( $X_H$ ) de nuestra señal:

$$X_D = U_D V_D \quad (3.29)$$

$$X_H = U_H V_H \quad (3.30)$$

### 3.6. Máscaras

Una vez hayamos conseguido obtener los espectrogramas de la pista rítmica ( $X_r(t, f)$ ) y del resto de la canción, es decir, el espectrograma de la parte no rítmica de la señal ( $X_{nr}(t, f)$ ), habremos alcanzado el objetivo de este documento. A pesar de ello, será necesario continuar tratando la señal para mejorar los resultados obtenidos. Para esto se utilizan lo que comúnmente se conoce como “*Máscaras*”. Existen muchos tipos de máscaras aunque en este documento nos centramos en explicar cómo se obtienen dos de estas máscaras: la máscara Wiener (Soft Masc) y la máscara binaria (Hard Masc).

A lo largo de este documento se trabajará únicamente con el módulo del espectrograma de la señal mezcla  $x_m(t)$ , sin tener en cuenta la información contenida en la fase. Así, vamos a generar unas máscaras que se aplicarán sobre  $X(t, f)$  de la siguiente manera:

$$X_r(t, f) = X(t, f) .* M_r \quad (3.31)$$

$$X_{nr}(t, f) = X(t, f) .* M_{nr} \quad (3.32)$$

Donde  $X(t, f)$  es el espectrograma complejo original (compuesto tanto por su módulo como por su fase) mientras que  $M_r$  y  $M_{nr}$  consisten en las máscaras que vamos a generar. El modo de obtención varía en función del tipo de máscara que se desee obtener, así para generar una máscara binaria (Hard) se tiene que:

$$M_r = \begin{cases} 1, & X_r(t, f) \geq X_{nr}(t, f) \\ 0, & \text{en otro caso} \end{cases} \quad (3.33)$$

$$M_{nr} = \begin{cases} 1, & X_{nr}(t, f) > X_r(t, f) \\ 0, & \text{en otro caso} \end{cases} \quad (3.34)$$

Estas máscaras toman únicamente los valores “1” ó “0” en función de que porcentaje de energía sea mayor, así por ejemplo, si creamos una máscara  $M_r$ , cabrá esperar que los “1” se sitúen en columnas en los instantes en los que se estén dando los golpes de batería dado que estos suelen presentar un patrón de repetitividad elevado.

Como ya se comentó antes (apartado 3.2), la energía de cada bin proviene de la suma de la energía de dichos bin de los espectrogramas  $X_r(t, f)$  y  $X_{nr}(t, f)$ , es por ello que este tipo de máscaras no suele ser recomendable ya que al trabajar

exclusivamente con “1” y “0” reparten la totalidad de la energía de cada bin a una u otra máscara. Para solventar este inconveniente se tiende a trabajar con máscaras Wiener, también conocidas como máscara Soft.

Este tipo de máscaras se construyen de la siguiente manera:

$$M_r = \left( \frac{X_r^p(t, f)}{(X_r^p(t, f) + X_{nr}^p(t, f))} \right)^{\frac{1}{p}} \quad (3.35)$$

$$M_{nr} = \left( \frac{X_{nr}^p(t, f)}{(X_r^p(t, f) + X_{nr}^p(t, f))} \right)^{\frac{1}{p}} \quad (3.36)$$

Siendo  $p$  generalmente un parámetro que varía entre 1 y 2, con  $p = 2$  como definición de máscara de Wiener.

Las máscaras Wiener solución el problema de las máscaras binarias ya que reparten la energía de cada bin entre ambas máscaras ( $M_r$  y  $M_{nr}$ ) obteniéndose unos resultados con una calidad de audio muy superior. La siguiente figura muestra un ejemplo de máscaras Hard y Soft, así como el resultado de multiplicar estas por un determinado espectrograma.

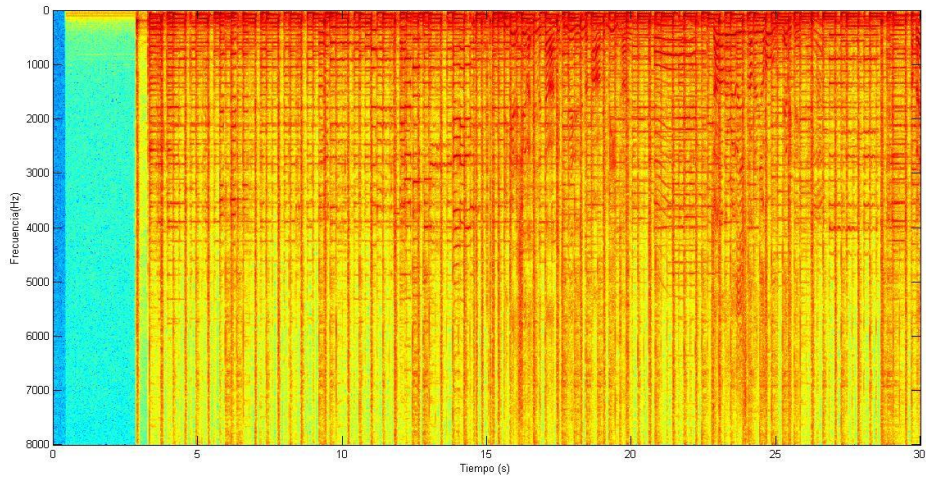


**Fig. 3.24** Espectrograma de la señal original.

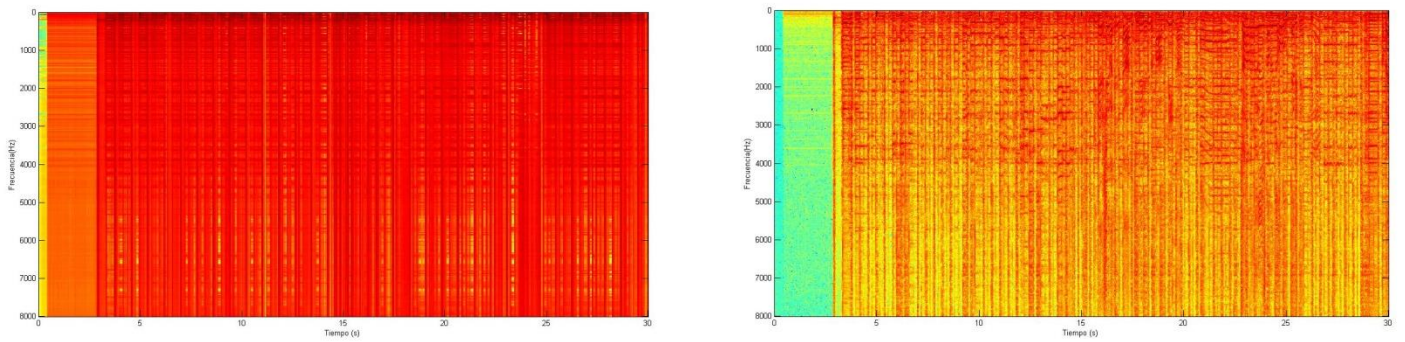


**Fig. 3.25 Izquierda: Espectrograma de la pista rítmica. Derecha: Espectrograma de la pista no rítmica.**

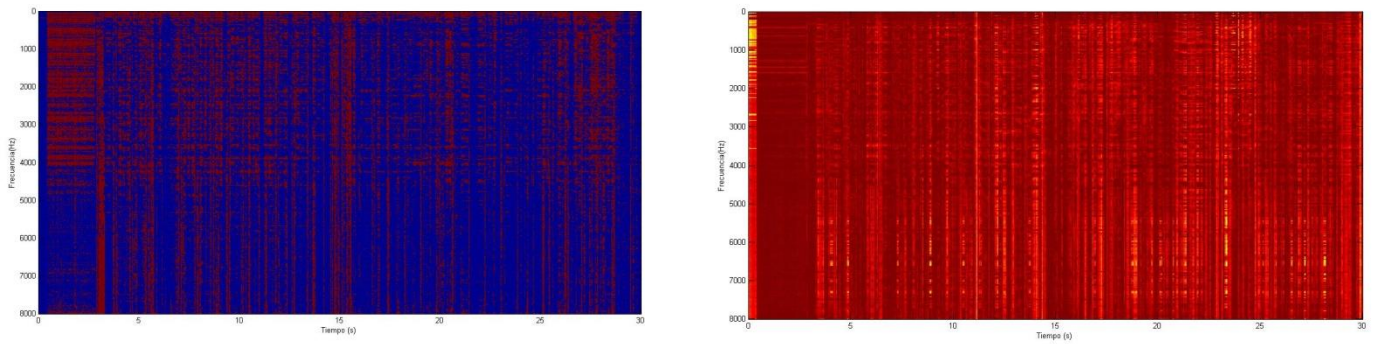


**Fig. 3.26 A la izquierda se aprecia la máscara binaria (Hard) obtenida a partir del espectrograma de la pista rítmica. A la derecha tenemos el resultado de multiplicar dicha máscara por el espectrograma original.**

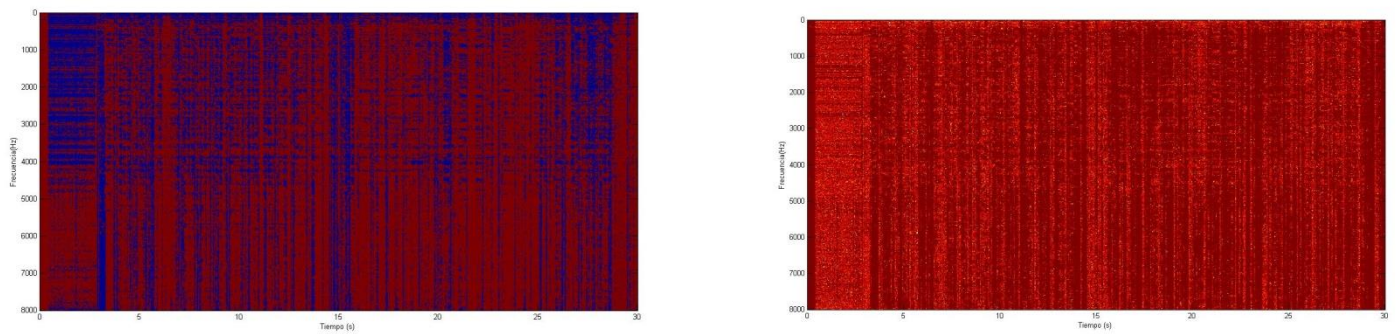


**Fig. 3.27 A la izquierda se aprecia la máscara binaria (Hard) obtenida a partir del espectrograma de la pista no rítmica. A la derecha tenemos el resultado de multiplicar dicha máscara por el espectrograma original.**

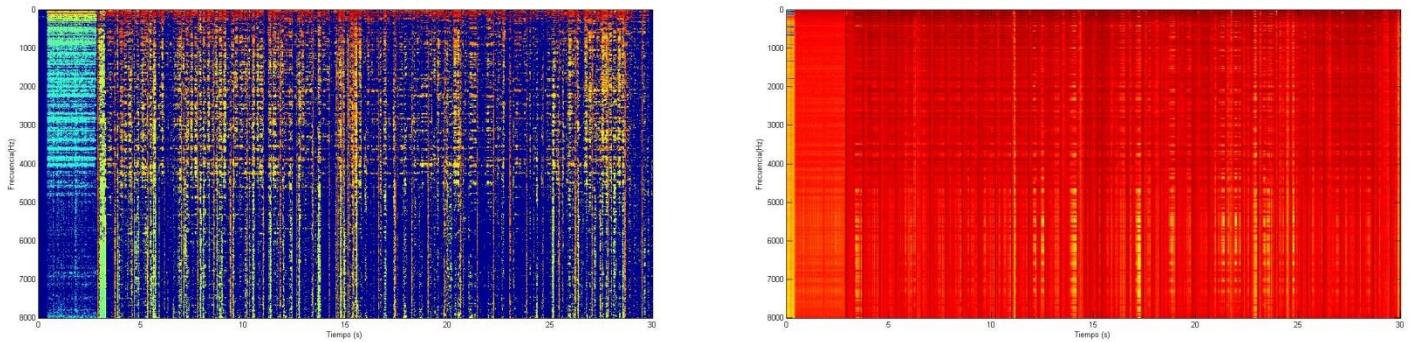


**Fig. 3.28** A la izquierda se aprecia la máscara Wiener (Soft) obtenida a partir del espectrograma de la pista rítmica. A la derecha tenemos el resultado de multiplicar dicha máscara por el espectrograma original.

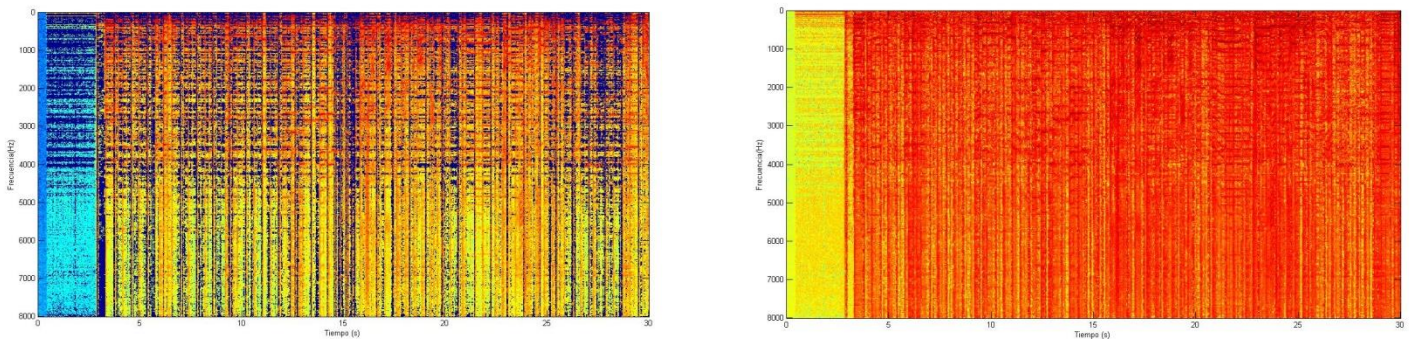


**Fig. 3.29** A la izquierda se aprecia la máscara Wiener (Soft) obtenida a partir del espectrograma de la pista no rítmica. A la derecha tenemos el resultado de multiplicar dicha máscara por el espectrograma original.

Como matiz en este apartado vamos a detallar que por norma general, la información perteneciente a muy baja frecuencia va asociada a la pista rítmica de la señal. Los instrumentos armónicos suelen generar sonidos a unas frecuencias intermedias. Por ello, a la hora de generar nuestras máscaras, debemos realizar un pequeño cambio consistente en asignar un valor de “1” todos los bins que se encuentren en muy baja frecuencia en la máscara utilizada para el cálculo de la pista rítmica.

En este documento se utiliza como frecuencia límite el valor de 100 Hz. Esto implica que todos los bins que se encuentren entre 0 y 100 Hz en la máscara de la pista rítmica tomarán el valor “1” mientras que todos esos mismos bins pertenecientes a la máscara de la pista no rítmica tomarán un valor de “0”.

Esto permite una ligera mejora del sistema dado que algunos restos de ciertas bases que hayan sido consideradas como no rítmicas pueden contener información

de la pista rítmica y, de este modo, conseguimos que esa información desaparezca de la pista no rítmica y acabe en su lugar apropiado.

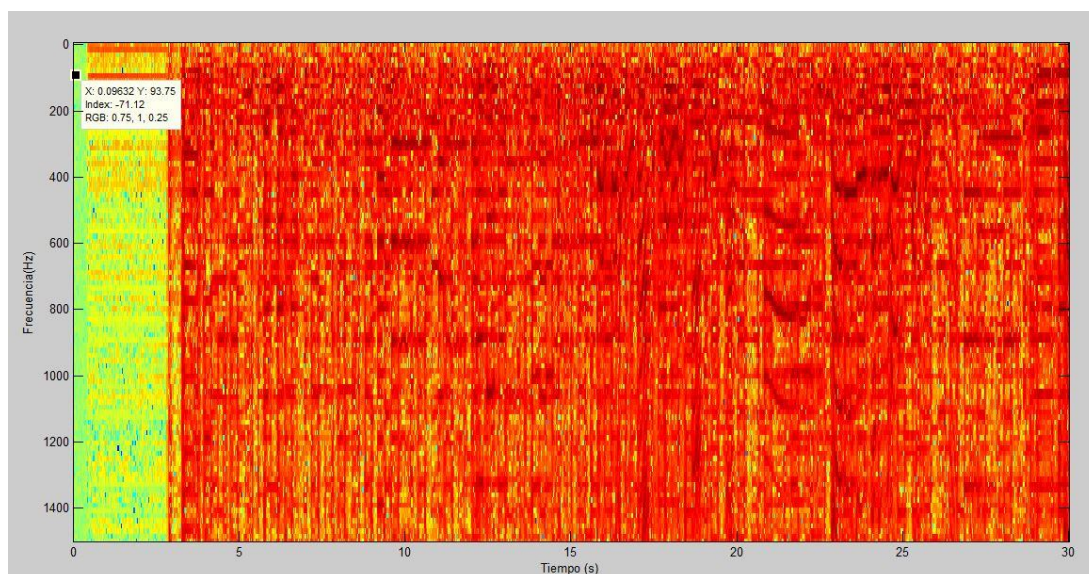


**Fig. 3.30** Espectrograma de la pista rítmica tras multiplicar por su respectiva máscara.

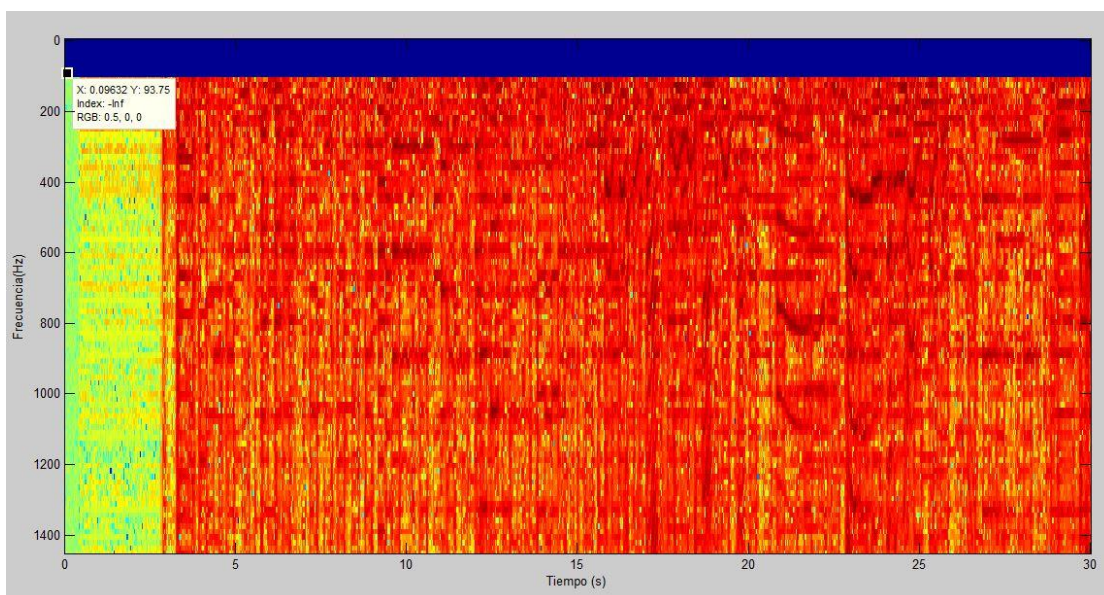


**Fig. 3.31** Espectrograma de la pista no rítmica tras multiplicar por su respectiva máscara.

# Capítulo 4

## 4. Implementación

Este capítulo se centra en explicar el procedimiento seguido para la extracción de la pista rítmica. Para ello se han seguido diversas líneas de trabajo, obteniéndose una diversa variedad de posibles caminos para lograr el objetivo. Algunas de estas líneas no han proporcionado los resultados que se esperaban o no llegan a ser tan útiles como sí lo son otros algoritmos pero sí que han servido como punto de partida para otras posibles ideas. Por todo ello, se explicará todo el procedimiento seguido por el alumno hasta alcanzar el objetivo final de este proyecto.

Ya que se conocen las herramientas utilizadas por los distintos algoritmos (capítulo 3), este capítulo intenta explicar las particularidades de cada algoritmo y las posibles variaciones que han sido realizadas para cumplir el objetivo propuesto.

### 4.1. Hipótesis inicial

La hipótesis inicial siempre estuvo presente desde el inicio de este proyecto y por ello el punto de partida del proyecto fue la búsqueda de unas determinadas bases que guardasen parecido entre sí a lo largo del tiempo, para la obtención de dichas bases se decide utilizar el algoritmo NMF.

Mediante el uso de la STFT obtenemos el espectrograma de nuestra señal de audio. Una vez obtenido este trabajaremos con su módulo  $X_m(f, t)$  aunque la información de fase  $X_p(f, t)$  será vital a la hora de devolver nuestra señal al dominio temporal por lo que esta queda almacenada para su posterior uso.

Al igual que desde un principio se pensó que la pista rítmica iría asociada a aquellas bases que se repitiesen a lo largo del tiempo, también se tuvo en cuenta el concepto de ritmo de la música. Por norma general, la música no presenta una estructura aleatoria dado que ese tipo de sonidos no resultarían agradables al oído humano. Es por ello que desde siempre, el ser humano ha compuesto música siguiendo un determinado patrón que se repite y sobre el que giran el resto de instrumentos. Este patrón es diferente en cada composición lo que convierte la extracción de la pista rítmica en un desafío de gran complejidad.

La búsqueda de un determinado parecido entre nuestras funciones base que se repitan a lo largo del tiempo nos lleva a hacernos la siguiente pregunta: *si el ritmo*

*es diferente en cada canción ¿cada cuántos frames de tiempo se repetiría un determinado patrón rítmico?*

Es ahí donde entran en juego los algoritmos descritos en la sección 4.3 sobre seguimiento del ritmo. En concreto en este sistema se utilizará el método implementado por REPET. Gracias a este, podremos obtener un valor  $P$  determinado que nos indique de forma aproximada el momento en el que se produce el evento que marca el ritmo de la señal.

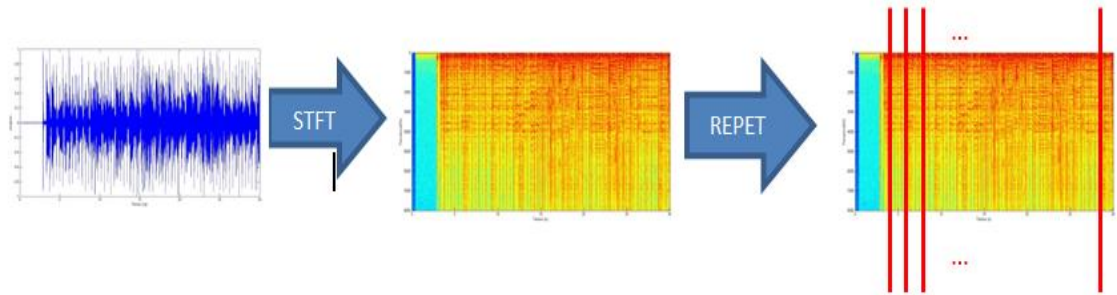


**Fig. 4.1 Procedimiento a realizar para la obtención del tiempo de beat mediante el uso de REPET.**

Realizamos a continuación el proceso de segmentación de nuestro espectrograma  $X_m(f, t)$  cada  $P$  frames de tiempo por lo que obtendríamos  $X_i(f, t)$  siendo  $i = 1, 2, 3 \dots N$

Conocido  $X_i(f, t)$  procedemos a realizar la técnica NMF. De este modo seremos capaces de obtener las bases que componen cada uno de estos fragmentos de nuestro espectrograma original. Como se ha explicado previamente, gracias a NMF somos capaces de obtener dos matrices de datos:

- La primera de estas contiene la información de las bases y se denomina *matriz de componentes* ( $W_i$ ). Esta será la matriz que más nos interesa dado que buscamos la comparación entre bases.
- La segunda matriz obtenida sería la *matriz de activaciones* ( $H_i$ ) que nos indica en qué momento se producen los sonidos contenidos en la matriz de componentes.

Aunque la matriz de activaciones no nos aporte demasiada información por el momento sí que será necesaria en un futuro a la hora de reconstruir el espectrograma que contenga la pista rítmica ya que, recordando la fórmula 3.1:

$$X \approx W * H$$

Esta NMF realizada puede llevarse a cabo con un número relativamente bajo de componentes ( $K$ ) ya que al tratarse de fragmentos del espectrograma  $X_m(f, t)$  la

duración de estos será bastante reducida y, por tanto, el número de sonidos que pueden darse en dicho intervalo de tiempo será también reducido.

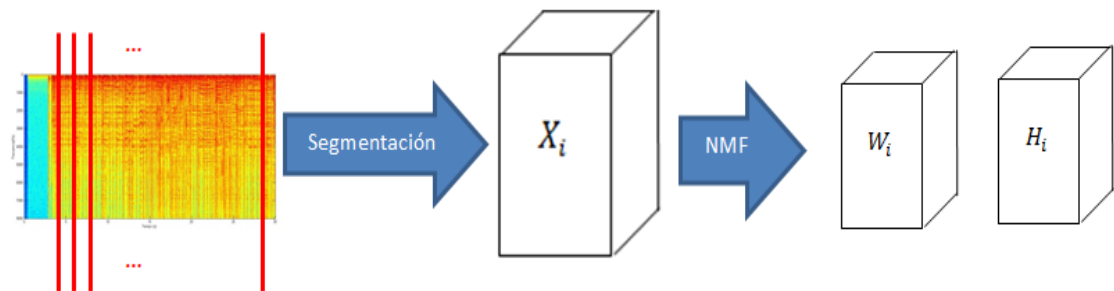


**Fig. 4.2 Realización de la segmentación del espectrograma y del posterior NMF de cada uno de los segmentos.**

Llegados a este punto se habrá logrado representar nuestra señal mediante una serie de funciones base que modelan cada uno de los sonidos que se producen a lo largo de la señal. Este es uno de los objetivos iniciales de este documento ya que seleccionando un determinado conjunto de dichas funciones base podremos obtener unos sonidos u otros de nuestra señal, pudiendo realizar la separación deseada.

El siguiente paso de este algoritmo consistiría en la comparación entre las bases de los diferentes segmentos. La realización de esta comparación se hizo de forma gradual, partiendo de una comparación hacia adelante (comparar con segmentos futuros) hasta realizar una comparación hacia adelante y hacia atrás (segmentos futuros y pasados).

Como paso previo a la realización de esta comparación vamos a normalizar cada una de las bases para trabajar con datos que oscilen entre “0” y “1”. Una vez realizada dicha normalización procedemos a multiplicar la base que nos interese (transpuesta) con aquella con la que queramos realizar la comparación. De este modo obtenemos un valor entre 0 y 1 que nos indica el grado de parecido entre ambas bases.

- 1) En un primer momento únicamente se realizan comparaciones con el segmento inmediatamente posterior. El barrido seguido para realizar dichas comparaciones es el siguiente:
  - Selecciono la primera base ( $K_1$ ) del primer segmento ( $W_1$ ) y la comparo con la primera base ( $K_1$ ) del segundo segmento ( $W_2$ ) y almaceno el resultado.
  - Selecciono la primera base ( $K_1$ ) del primer segmento ( $W_1$ ) y la comparo con la segunda base ( $K_1$ ) del segundo segmento ( $W_2$ ) y almaceno el resultado.

- Repito el proceso hasta comparar con todas las bases del segundo segmento.
  - De este modo se obtiene una matriz tridimensional que contiene los resultados de todas estas comparaciones donde la fila indica la base de mi segmento que estoy comparando y la columna representa la base del siguiente segmento con la que estoy realizando la comparación, por último la tercera dimensión indica el segmento en el que me encuentro. Así por ejemplo el elemento [3, 7, 5] de dicha matriz me indica el parecido entre la base 3 del segmento 5 y la base 7 del segmento inmediatamente posterior, en este caso sería la base 7 del segmento 6.
- 2) Más adelante se propuso extender este procedimiento de modo que la comparación se realice no sólo con el segmento inmediatamente posterior sino que pueda realizarse con N segmentos posteriores. El procedimiento aunque más largo es exactamente idéntico al descrito en 1). En este caso no se obtiene una matriz de tres dimensiones con resultados sino que pasamos a tener cuatro dimensiones. Aunque pueda parecer complicado, la interpretación de los datos es sencilla como puede verse en el siguiente ejemplo: el elemento [3, 7, 6, 2] se corresponde a la comparación entre la base 3 del segmento en el que me encuentro (en este caso el segmento 2, cuarta coordenada) con la base 7 (segunda coordenada) del segmento del segmento 8 (dado que me encuentro en el 2 y la tercera coordenada me indica que estoy comparando con el segmento que ocupa 6 posiciones más adelante).
- 3) Finalmente se decidió realizar comparaciones tanto hacia segmentos futuros como pasados ya que el parecido entre bases no tiene por qué darse únicamente entre segmentos que se encuentren más adelante (temporalmente hablando) sino que un sonido que se haya producido antes puede repetirse en el momento actual o en un futuro. Por ello, en este último paso se ha decidido realizar comparaciones en ambos sentidos. El algoritmo implementado en esta última versión necesita la introducción de un parámetro que sea el número de segmentos (hacia adelante y hacia atrás) con el que queremos realizar las comparaciones:  $U$ . Así si seleccionásemos ese parámetro como por ejemplo 2 nuestro algoritmo compararía cada segmento con sus dos anteriores y con sus dos posteriores. Esto es importante para entender cómo deberían interpretarse los datos devueltos. Siguiendo con el ejemplo anterior, si tenemos  $U = 2$  y nuestro espectrograma se divide en 7 segmentos, nuestro algoritmo realiza las siguientes comparaciones:

- El segmento 1 se compara con los segmentos 2, 3, 4 y 5.

|   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|

- El segmento 2 se compara con los segmentos 1, 3, 4 y 5.

|   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|

- El segmento 3 se compara con los segmentos 1, 2, 4 y 5.

|   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|

- El segmento 4 se compara con los segmentos 2, 3, 5 y 6.

|   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|

- El segmento 5 se compara con los segmentos 3, 4, 6 y 7.

|   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|

- El segmento 6 se compara con los segmentos 3, 4, 5 y 7.

|   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|

- El segmento 7 se compara con los segmentos 3, 4, 5 y 6.

|   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|

Por esto, para cada segmento en el que nos encontremos tendremos una serie de segmentos con los que comparar, esto vendrá indicado en la tercera coordenada de nuestra matriz de resultados. Por ejemplo el valor [2, 5, 3, 2] se corresponde con la comparación entre la base 2 (primera coordenada) del segmento en el que nos encontramos (segmento 2, cuarta coordenada) y la base 5 (segunda coordenada) de del segmento 4 (esto es así ya que el segmento 2 lo comparamos con los segmentos 1, 3, 4 y 5).

Tras realizarse todas las posibles comparaciones entre las diversas bases habrá llegado el momento de seleccionar aquellas que mayor parecido guarden entre sí. Se escoge un determinado umbral y se seleccionaran todos aquellos datos de la matriz de resultados que sean mayores o iguales que dicho umbral. Pero esto no es suficiente de por sí ya que para considerar que una base pertenece a la pista rítmica de la señal no basta con que se parezca mucho a otra base en un único

segmento sino que deberá tener parecido con bases en todos los segmentos con los que se haya realizado la comparación. Esto debe ser así debido al uso del algoritmo REPET que nos indica la longitud de los segmentos a fragmentar.

Por todo ello, únicamente seleccionaremos aquellas bases que posean un parecido superior al umbral con bases en el resto de segmentos con los que se ha comparado.

Un ejemplo sería el siguiente, supongamos que tenemos  $N = 7$  segmentos y  $U = 2$ , realizando un NMF con 5 bases para cada segmento ( $K = 5$ ). Una matriz de cuatro dimensiones puede interpretarse como N matrices tridimensionales, siendo N el valor de la cuarta dimensión. Utilizando un determinado umbral se obtendrían 7 matrices de parecidos (una matriz por cada segmento). Si la condición impuesta por el umbral se cumple a lo largo de toda la tercera dimensión podemos utilizar dicha base para realizar la reconstrucción. Una forma gráfica de verlo sería la siguiente, representamos la comparación realizada para el **cuarto segmento** con los segmentos 2, 3, 5 y 6. En **rojo** aparecen aquellas comparaciones que superan el umbral de parecido mientras que en **azul** aparecen aquellas que no lo superan.

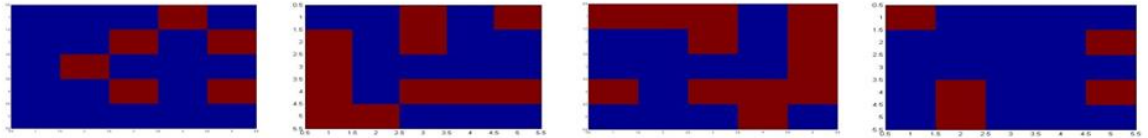


**Fig. 4.3 Resultados obtenidos tras aplicar el umbral de parecido. En rojo aparecen aquellas comparaciones que superen dicho umbral y en azul aquellas que no.**

Puede comprobarse como la única comparación que se cumple al realizarla con los cuatro segmentos es la comparación de la **base 4** (fila 4) con la base 5 (columna 5). Visto esto, a la hora de reconstruir el segmento número 4 únicamente utilizaremos la **base 4**.

Realizando este procedimiento con todos los segmentos veremos con que bases se reconstruirá cada uno de ellos. Para la reconstrucción simplemente se ha de multiplicar las bases que deseemos utilizar (columna en la matriz W) por esa misma base en la matriz de activaciones (fila en la matriz H). Con esto tendremos nuestro segmento reconstruido  $X_{i\_rec}(f, t)$ , a este se unirán el resto formando de nuevo un espectrograma  $X_{r\_rec}(f, t)$ , este será devuelto al dominio temporal mediante el uso de la ISTFT (Inverse STFT).

Antes de realizar la ISTFT vamos a realizar una serie de procedimientos para mejorar la calidad de la señal generada. En primer lugar se generan las máscaras Wiener (o Soft) a partir del espectrograma obtenido mediante la reconstrucción  $X_{r\_rec}(f, t)$ . Estas máscaras se aplican al espectrograma de partida ( $X_m(f, t)$ ) tal y como se detalla en la sección 4.6. Una vez obtenido nuestro espectrograma con su respectiva máscara  $X_{r\_masc}(f, t)$  realizamos un último procedimiento de

postprocesado para seleccionar las componentes más bajas de frecuencia tal y como se explica en la sección 4.7

Tras todo esto ha llegado el momento de realizar la ISTFT gracias a la cual devolvemos nuestra señal al dominio temporal y podremos reproducirla para escuchar los resultados obtenidos.

## 4.2. NMF modificado

En el documento previamente citado [26] se ha trabajado con dos señales de entrada ( $X$  e  $Y$ ) y se buscaba la separación armónico-percusiva, sin embargo, para el desarrollo de este trabajo el procedimiento a seguir será ligeramente diferente. En lugar de utilizar dos señales de entrada nuestro sistema utilizará exclusivamente el espectrograma segmentado y al no utilizar ningún tipo de entrenamiento ni patrones espectrales auxiliares podemos afirmar que utilizamos un sistema ciego (no supervisado). Esto supone una mejora con respecto a sistemas previos. A diferencia de [26], en este documento tratamos de realizar una extracción de la pista rítmica de la señal en lugar de la separación armónico-percusiva explicada anteriormente.

Al no contar con la señal  $Y$  el procedimiento a seguir será diferente, y únicamente trabajaremos con las matrices que describimos a continuación.

La analogía entre la nomenclatura utilizada en [26] y en este documento es:

- $U_D = W_r$
- $U_H = W_{nr}$
- $V_D = H_r$
- $V_H = H_{nr}$

Conocido el funcionamiento general del algoritmo NMPCF (Estado del arte sección 3.4) vamos a detallar los cambios que aplicamos para poder trabajar con NMF. Esto nos permitirá poder realizar la separación entre matriz de componentes rítmicas y matriz de componentes no rítmicas y entre matriz de activaciones rítmicas y matriz de activaciones no rítmicas.

Para la puesta en marcha de este sistema ha sido necesario reescribir las ecuaciones generales de divergencia:

$$H = H .* \frac{W^T * (W * H)^{(\beta-2)} * X}{W^T * (W * H)^{(\beta-1)}}$$

$$W = W .* \frac{(W * H)^{(\beta-2)} * X * H^T}{(W * H)^{(\beta-1)} * H^T}$$

Esto es así dado que no trabajamos únicamente con una matriz de componentes y una de activaciones, sino que serán necesarias las que describimos a continuación:

- $W_r$ : esta matriz de componentes contendrá aquellas componentes que se repitan a lo largo de toda la señal, esto es así ya que se almacena y actualiza en cada iteración de cada segmento. Esta matriz es común para todos los segmentos.
- $H_r$ : matriz de activaciones de  $W_r$ . Esta matriz es diferente para cada uno de los segmentos.
- $W_{nr}$ : matriz de componentes del resto de sonidos que no tienden a repetirse a lo largo de la señal. Esta matriz es diferente para cada uno de los segmentos.
- $H_{nr}$ : matriz de activaciones de  $W_{nr}$ . Esta matriz es diferente para cada uno de los segmentos.

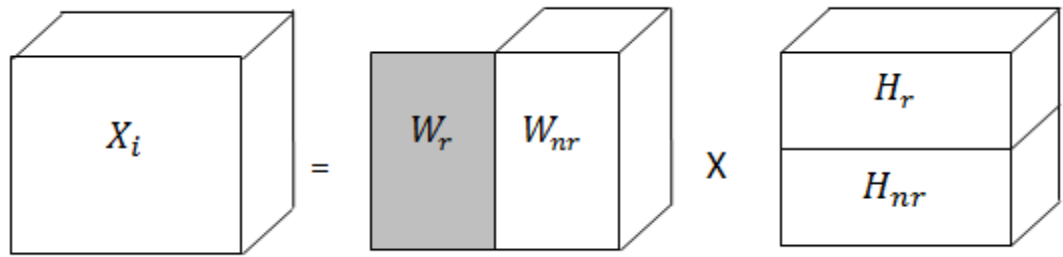


**Fig. 4.4 Ilustración del funcionamiento de NMF modificado. En este caso trabajamos con una matriz de componentes ( $W_r$ ) bidimensional. El resto de matrices tendrán 3 dimensiones dado que serán diferentes para cada segmento.**

Las ecuaciones generales que permiten actualizar las matrices de componentes y activaciones hasta alcanzar la convergencia son:

$$H = H .* \frac{W^T * (X_{i,reconstruida})^{(\beta-2)} * X}{W^T * (X_{i,reconstruida})^{(\beta-1)}} \quad (4.1)$$

$$W = W .* \frac{(X_{i,reconstruida})^{(\beta-2)} * X * H^T}{(X_{i,reconstruida})^{(\beta-1)} * H^T} \quad (4.2)$$

Siendo por norma general  $X_{i,reconstruida} = W * H$  pero para nuestro caso particular:

$$X_{i,reconstruida} = W_r H_r + W_{nr} H_{nr} \quad (4.3)$$

Las ecuaciones de actualización de [26] consisten en un desarrollo de las expresiones generales utilizando como parámetro  $\beta = 2$ , como  $X_{i,reconstruida} = U_D V_D + U_H V_H$  y como  $X_{i,reconstruida} = U_D W_D$ .

Si utilizamos ahora la nomenclatura de este documento y escribimos las ecuaciones de actualización de su forma más genérica (sin particularizar para ningún  $\beta$ ), tenemos las siguientes funciones:

$$\begin{aligned} W_r &= W_r .* \frac{\left[ \left( (X_{i,reconstruida})^{(\beta-2)} .* X \right) * H'_r \right]}{\left[ (X_{i,reconstruida})^{(\beta-1)} * H'_r \right]} \\ H_r &= H_r .* \frac{W'_r * \left[ (X_{i,reconstruida})^{(\beta-2)} .* X \right]}{W'_r * (X_{i,reconstruida})^{(\beta-1)}} \\ W_{nr} &= W_{nr} .* \frac{\left[ (X_{i,reconstruida})^{(\beta-2)} .* X \right] * H'_{nr}}{(X_{i,reconstruida})^{(\beta-1)} * H'_{nr}} \\ H_{nr} &= H_{nr} .* \frac{W'_{nr} * \left[ (X_{i,reconstruida})^{(\beta-2)} .* X \right]}{W'_{nr} * (X_{i,reconstruida})^{(\beta-1)}} \end{aligned}$$

Ecuaciones (4.4), (4.5), (4.6), (4.7) y (4.8)

Recordemos que para este caso que estamos detallando  $X_{i,reconstruida} = W_r H_r + W_{nr} H_{nr}$ .

Estas ecuaciones de actualización permiten obtener por separado tanto las componentes y activaciones de la pista rítmica como las componentes y activaciones de la pista no rítmica. A la hora de iterar durante este algoritmo hemos de respetar que la matriz  $W_r$  ha de actualizarse en cada iteración, es decir, para cada segmento. Por otro lado, el resto de matrices ( $H_r$ ,  $W_{nr}$  y  $H_{nr}$ ) que son tridimensionales se actualizan exclusivamente cuando nos encontramos en el segmento correspondiente. Esto quiere decir que si tuviésemos un espectrograma dividido en tres segmentos tendríamos que:

- En la primera iteración se actualizan las matrices  $W_r$ ,  $H_r$ (segmento 1),  $W_{nr}$ (segmento 1) y  $H_{nr}$ (segmento 1).

- En la segunda iteración se actualizan las matrices  $W_r$ ,  $H_r$ (segmento 2),  $W_{nr}$ (segmento 2) y  $H_{nr}$ (segmento 2).
- En la tercera iteración se actualizan las matrices  $W_r$ ,  $H_r$ (segmento 3),  $W_{nr}$ (segmento 3) y  $H_{nr}$ (segmento 3).
- En la cuarta iteración se actualizan las matrices  $W_r$ ,  $H_r$ (segmento 1),  $W_{nr}$ (segmento 1) y  $H_{nr}$ (segmento 1).
- Y así sucesivamente hasta alcanzar la convergencia mediante la función de coste utilizada.

Una vez conocidas estas ecuaciones vamos a exponer como quedarían las funciones de divergencia o de coste más generales teniendo en cuenta el nuevo término que debe aparecer acompañando al parámetro  $\lambda$ , recordando que estas eran la distancia Euclídea, Kullback Leibler e Itakura Saito:

- Distancia Euclídea:

$$d_{EUC} = \frac{1}{2} \|X_i - X_{i,reconstruida}\|^2$$

$$d_{EUC} = \frac{1}{2} \|X_i - W_r H_r - W_{nr} H_{nr}\|^2 \quad (4.9)$$

- Kullback Leibler:

$$d_{KL} = \left( X_i * \log \frac{X_i}{X_{i,reconstruida}} - X_i + X_{i,reconstruida} \right)$$

$$d_{KL} = \left( X_i * \log \frac{X_i}{W_r H_r + W_{nr} H_{nr}} - X_i + W_r H_r + W_{nr} H_{nr} \right) \quad (4.10)$$

- Itakura Saito:

$$d_{IS} = \left( \frac{X_i}{X_{i,reconstruida}} - \log \frac{X_i}{X_{i,reconstruida}} - 1 \right)$$

$$d_{IS} = \left( \frac{X_i}{W_r H_r + W_{nr} H_{nr}} - \log \frac{X_i}{W_r H_r + W_{nr} H_{nr}} - 1 \right) \quad (4.11)$$

#### 4.2.1. REPET + NMF modificado

Tras la implementación del sistema inicial se comprobó que el coste computacional de dicho algoritmo era bastante elevado ya que el número de comparaciones a realizar era enorme. Debido a esto se busca un nuevo método que permita la obtención de unas determinadas bases que se correspondan únicamente a la pista rítmica de la señal, sin tener que decidir entre el total de las bases generadas por NMF sino “generar” directamente unas bases que reproduzcan los sonidos característicos de la pista rítmica.

Es aquí donde entra en juego el algoritmo NMF modificado que nos permite obtener dos matrices de componentes diferentes. El funcionamiento de este algoritmo ha sido previamente detallado a lo largo de la sección 4.5.

En un primer momento tratamos de implementar un sistema parecido al descrito en el documento [26] pero dado que en él no utilizan un sistema ciego sino que necesitan la información aportada por una señal de percusivos, se tomó un enfoque diferente que nos permitiese trabajar únicamente con la señal de entrada original  $X(t)$ .

Partimos de la señal en el dominio temporal a la que aplicamos la STFT para generar su espectrograma  $X(f, t)$  a partir del cual obtenemos el módulo del mismo  $X_m(f, t)$ . Al igual que en el sistema anterior, recurrimos al algoritmo REPET para obtener una marca de tiempo que nos indique en qué momento se produce un sonido rítmico que nos marque la pista rítmica. Conocido dicho P procedemos a realizar una segmentación de  $X_m(f, t)$  obteniendo de nuevo  $X_i(f, t, N)$ .

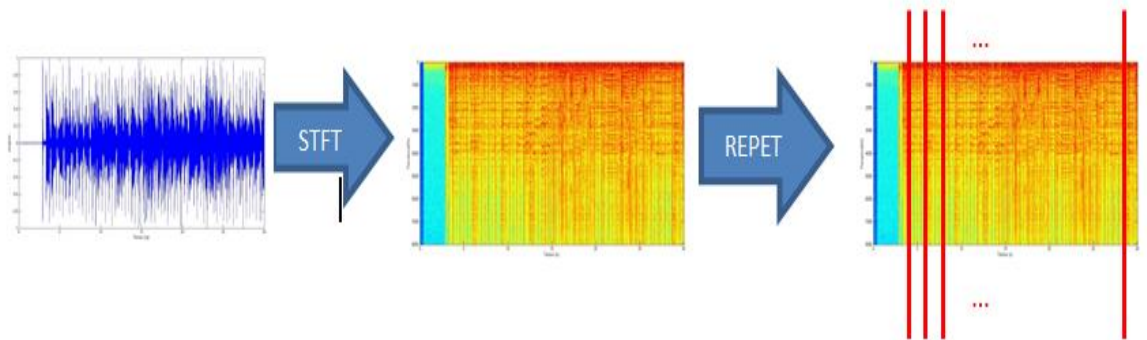


Fig. 4.5 Procedimiento a realizar para la obtención del tiempo de beat mediante el uso de REPET.

Como paso previo a la ejecución del algoritmo NMF modificado se ha llevado a cabo una primera ejecución de NMF sobre el espectrograma  $X_m(f, t)$  con un número de bases reducido, en concreto el número de bases que se desee que tenga la matriz  $W_r$ . De este modo al ejecutar el segundo NMF la matriz  $W_r$  ya habrá sido inicializada y se respetará sin actualizar (congelada) a lo largo del resto del algoritmo.

Como datos de salida obtendremos las matrices  $H_r$ ,  $W_{nr}$  y  $H_{nr}$  y si multiplicamos la matriz de componentes de la pista rítmica por cada una de las matrices de activaciones de la pista rítmica obtendremos cada uno de los segmentos de nuestro espectrograma reconstruidos.

$$X_{r\_rec,i}(f, t, N) = W_{rep} * H_{rep,i} \quad (4.12)$$

Tras esto, desdoblamos la matriz  $X_{r\_rec,i}(f, t, N)$ , que es tridimensional, de modo que obtengamos una matriz bidimensional de dimensiones iguales a la matriz de partida  $X_m(f, t)$ . Al igual que en el sistema anterior realizamos el procedimiento de creación de máscaras que aplicaremos sobre nuestro espectrograma original. Con todo ello habremos concluido la implementación de este sistema.

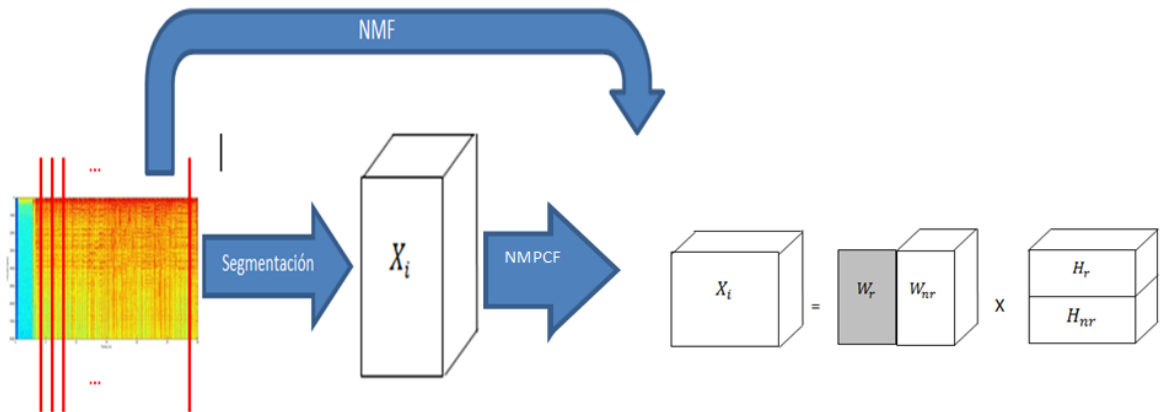


**Fig. 4.6 Realización de NMF para inicializar  $W_r$  así como segmentación del espectrograma según indica REPET. Posterior realización de NMF modificado.**

Cabe destacar que a lo largo del desarrollo de este proyecto se han llevado a cabo implementaciones de algunas posibles variantes de este algoritmo:

- En un primer momento se optó por ejecutar NMF modificado sin ningún otro algoritmo que ayudase a la inicialización de las matrices. Tras comprobar la efectividad del uso de un NMF previo, este se ha

mantenido a lo largo del resto del proyecto en las sucesivas implementaciones.

- Otras implementaciones incluían la información sobre la matriz de activaciones obtenida del NMF inicial. Esta se matriz  $H$  se fragmenta cada  $P$  frames de tiempo ( $P$  viene indicado por REPET) y se toma como punto de partida de la matriz  $H_r$ . Con esto se consigue una pequeña mejora del sistema aunque apenas si es apreciable.

#### 4.2.2. ELLIS + NMF modificado

Esta sección presenta un gran parecido con la anterior dado que en ambos se implementa un sistema que se base en el uso del algoritmo NMF modificado tomando como datos de entrada una matriz tridimensional obtenida a partir de la segmentación del módulo del espectrograma de la señal de entrada  $X_m(f, t)$ .

De nuevo comenzamos el sistema realizando la STFT sobre la señal de audio para obtener el espectrograma  $X(f, t)$  a partir del cual obtenemos su módulo como  $X_m(f, t)$ . Es ahora donde se produce la mayor diferencia con respecto al algoritmo implementado en la sección anterior. Dado que REPET únicamente devuelve una marca de tiempo y supone que el resto de eventos relacionados con el ritmo de la señal, es bastante común que esa suposición lleve a errores. Ya se comentó en la sección 3.3 el hecho de que el ser humano no interpreta música con una exactitud tan precisa, es por ello que se recurre al algoritmo ELLIS para obtener unas marcas de tiempo adecuadas y acordes con el ritmo de la señal.

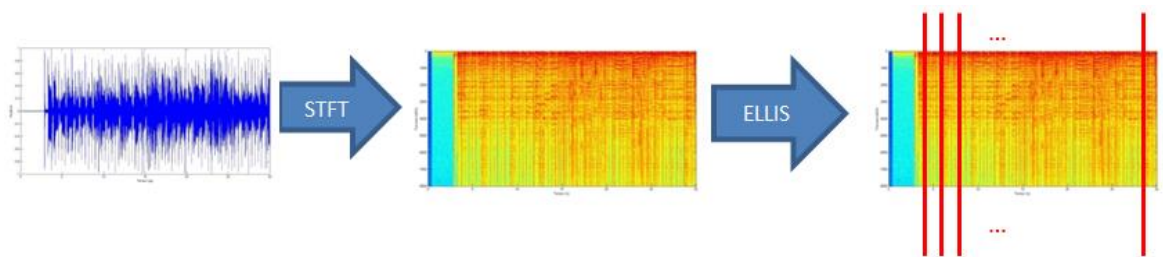


**Fig. 4.7 Procedimiento a realizar para la obtención del tiempo de beat mediante el uso de ELLIS.**

El siguiente paso tras la obtención de los tiempos de beat consistiría en la segmentación de  $X_m(f, t)$  para obtener  $X_i(f, t, N)$ . Al usar este algoritmo hay que tener muy en cuenta el relleno con ceros que se debe llevar a cabo tal y como fue descrito en la sección 3.3 y que más adelante habrá que revertir. Habiendo realizado la segmentación de nuestro espectrograma y obtenido de nuevo un espectrograma

tridimensional procedemos a sumar en profundidad las componentes de  $X_i(f, t, N)$ , esto es, tratamos de obtener una matriz de dos dimensiones mediante la suma de todos los elementos en la tercera dimensión. Por ejemplo si nuestro espectrograma  $X_i(f, t, N)$  consiste en una matriz tridimensional de  $(3 \times 5 \times 4)$ , es decir, 4 matrices de 3 filas y 5 columnas que se colocan una tras otra formando un cubo, realizamos una suma de todos los elementos a lo largo de la tercera dimensión. El resultado de esto sería una matriz de  $(3 \times 5)$  cuyos elementos provienen de la suma en profundidad (tercera dimensión).

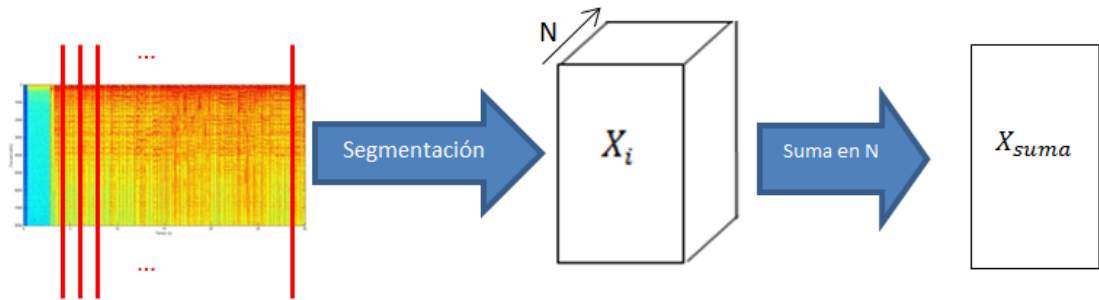


**Fig. 4.8 Segmentación del espectrograma (rellenando con ceros) y posterior suma a lo largo de la dimensión N.**

Este paso resulta de utilidad ya que si el algoritmo ELLIS funciona de forma correcta y las marcas de tiempo coinciden con los instantes en los que se marca el ritmo, al principio de cada uno de los segmentos generados tras cortar  $X_m(f, t)$  se producirá un evento característico de la pista rítmica, por tanto, la matriz  $X_{suma}$  quedará aproximadamente de la siguiente forma:

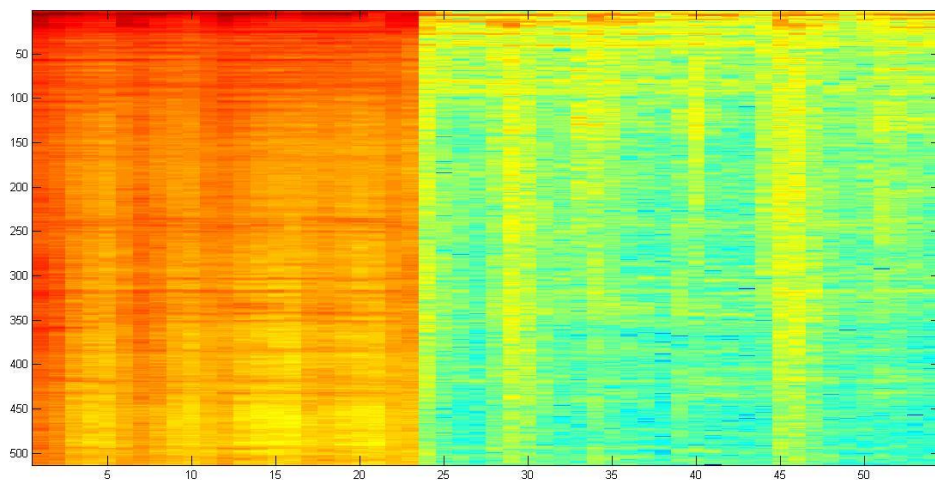


**Fig. 4.9 Ejemplo del resultado obtenido tras realizar la suma a lo largo de N.**

Donde es fácilmente apreciable que al comienzo del segmento se produce un golpe de batería o algún tipo de sonido percusivo que marque el ritmo de la señal,

mientras que en el resto del segmento los sonidos tienden a ser de tipo armónico (no rítmico).

Sobre este espectrograma  $X_{suma}$  procedemos a realizar el algoritmo NMF con un propósito parecido al de la sección anterior, es decir, obtener unas bases  $W_r$  pero que a diferencia del sistema anterior se respetarán sin actualizarse cuando se realice la segunda ejecución del algoritmo NMF modificado. Podría pensarse en utilizar también la matriz de activaciones devuelta tras realizar el NMF pero no aporta información de utilidad ya que al haber realizado esa suma en profundidad se habrá perdido la información sobre cuando se producen los eventos. Es por ello que únicamente mantenemos la matriz de componentes devuelta por NMF.

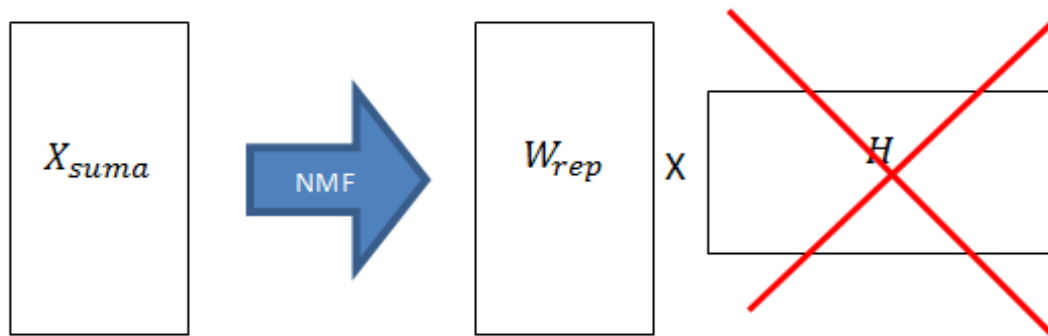


**Fig. 4.10 Realización de NMF sobre  $X_{suma}$  para obtener los valores de  $W_r$ .**

El siguiente paso será la realización del algoritmo NMF modificado introduciendo como datos de entrada la matriz tridimensional  $X_i(f, t, N)$  y la matriz de componentes  $W_r$ . El procedimiento será el mismo que el que se explica en la sección 4.2.2, ya conocemos cuales serán las bases de la pista rítmica ( $W_r$ ), es por esto que a la hora de actualizar el resto de matrices ha de respetarse (manteniéndose congelada) la matriz  $W_r$ , al igual que hacíamos en la implementación del anterior sistema.

Cuando la segunda ejecución de NMF finalice y haya convergido habrá llegado el momento de reconstruir los segmentos. La reconstrucción de estos es sencilla ya que una vez tengamos, además de nuestra matriz  $W_r$  que teníamos del NMF inicial, la matriz  $H_{r,i}$  que procede de la realización de NMF modificado. Realizando el producto  $W_r * H_{r,i}$  para cada uno de los  $i$  que corresponden a los diferentes segmentos tendremos una reconstrucción de estos, manteniendo exclusivamente la información perteneciente a la pista rítmica.

Una vez los segmentos han sido reconstruidos y tenemos una matriz tridimensional  $X_{r\_rec,i}(f, t, N)$  es la hora de “desdoblarla” para que vuelva a ser una matriz de dos dimensiones. En este punto hemos de tener especial cuidado puesto que como ya se mencionó, el uso del algoritmo ELLIS implica la necesidad de un relleno con ceros en cada uno de los segmentos. Esto provoca que al desdoblar

$X_{r\_rec,i}(f, t, N)$  para devolverla a su forma original de dos dimensiones aparezca algo por este estilo:

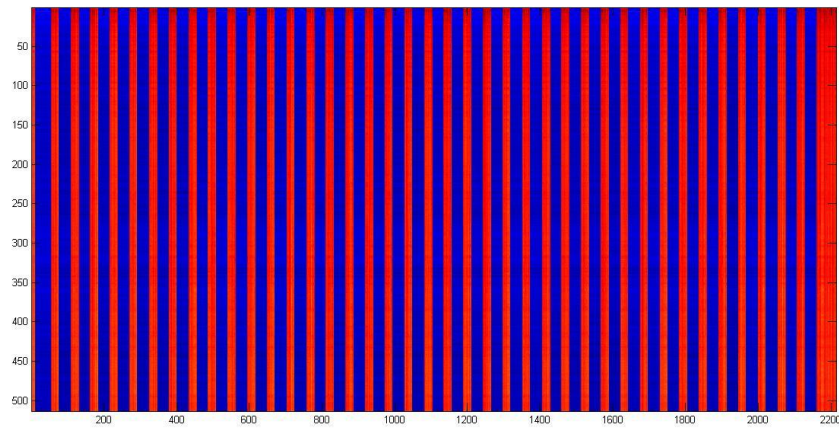


**Fig. 4.11 Resultado de la reconstrucción del espectrograma, prestar especial atención a las franjas azules correspondientes al relleno con ceros.**

Donde las franjas se corresponden con el relleno de ceros que se dio tras el uso de ELLIS. Tras su eliminación el espectrograma que resulta sería:

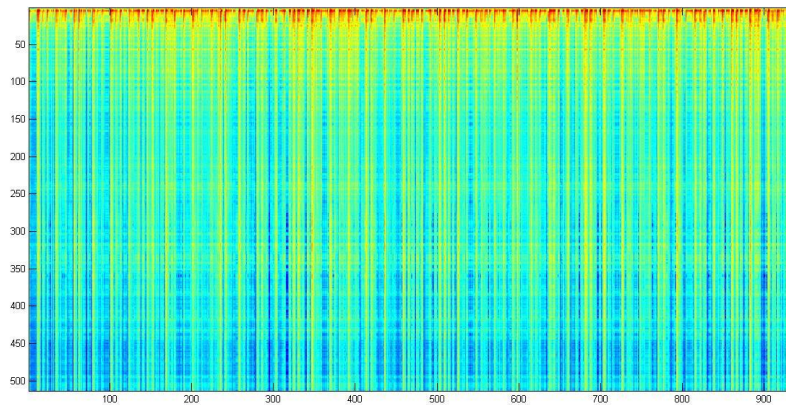


**Fig. 4.12 Resultado obtenido tras la eliminación de los ceros sobrantes.**

Tras haber tenido esto en cuenta volvemos a realizar el cálculo de las máscaras correspondientes. Mediante el uso de ISTFT devolvemos la señal al dominio temporal para poder escuchar el resultado obtenido.

### 4.3. Selección de bases con menor varianza

Tras la implementación de los sistemas anteriores se llegó a la conclusión de que, al jugar estos con una gran cantidad de variables (número de bases de la pista rítmica, número de bases de la pista no rítmica y parámetro  $\beta$  que indique la función de coste a utilizar) resultaba muy difícil encontrar los parámetros que optimizasen el resultado. Además al poder ser tan variable el ritmo de una señal con respecto al de otra, una posible combinación de parámetros que optimizasen el resultado para una determinada señal no conseguían producir un buen resultado en otra señal diferente. Es por ello que se pensó en usar otro sistema cuya optimización fuese más sencilla y el coste computacional menor.

Hasta este momento el proyecto ha girado en torno a la búsqueda de unas buenas bases que sean capaces de modelar con exactitud aquellos sonidos que caracterizan la pista rítmica de las señales. Mediante el primer sistema implementado tratábamos de seleccionar aquellas bases que guardaban un mayor parecido a lo largo de varios segmentos (sección 4.1) mientras que el segundo y tercer sistema puestos en marcha (sección 4.2.1 y 4.2.2) trataban de generar unas bases que se pareciesen lo máximo posible a aquellos sonidos que se repiten a lo largo de toda la señal (sonidos generalmente asociados a la pista rítmica), esto lo conseguían gracias al hecho de congelar la matriz  $W_r$  y actualizar el resto de matrices.

En esta sección se utiliza un enfoque ligeramente diferente. Lo que se intenta ahora es la selección de unas buenas bases (matriz de componentes  $W$ ) que definan bien los sonidos deseados. Pero ¿cómo sabemos qué bases son las que van asociadas a sonidos pertenecientes a la pista rítmica de nuestra señal? Es ahí cuando entra en juego la información aportada por la matriz de activaciones  $H$ .

La implementación del sistema comienza, al igual que en los sistemas previamente implementados, mediante la obtención del espectrograma de nuestra señal ( $X(f, t)$ ) tras lo que nos quedamos exclusivamente con su módulo  $X_m(f, t)$ . A continuación realizamos el algoritmo REPET para obtener el  $P$  característico de esta señal.

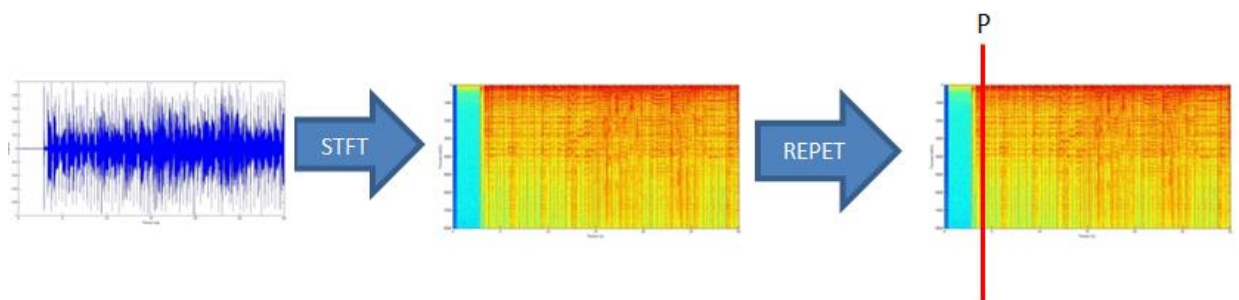


**Fig. 4.13** Procedimiento a realizar para la obtención del tiempo de beat ( $P$ ) mediante el uso de REPET.

El siguiente paso a realizar consiste en poner en marcha el algoritmo NMF sobre el espectrograma completo  $X_m(f, t)$ . Es aquí donde se aprovecha la información obtenida por REPET ya que tras realizar diversas pruebas con diferente número de bases, se ha comprobado que por norma general, los mejores resultados se obtienen realizando el NMF con un número de bases igual al valor del parámetro P aportado por REPET.

Tras la realización de este NMF inicial habremos obtenido las matrices de componentes (W) y de activaciones (H). Dejando de lado por el momento la información contenida en la matriz de componentes, nos centramos en buscar el patrón de repetitividad de las bases, lo cual es fácilmente apreciable en la matriz de activaciones. Para ello lo primero que debemos hacer es fragmentar la matriz H en segmentos del tamaño indicado por P (información aportada por REPET).

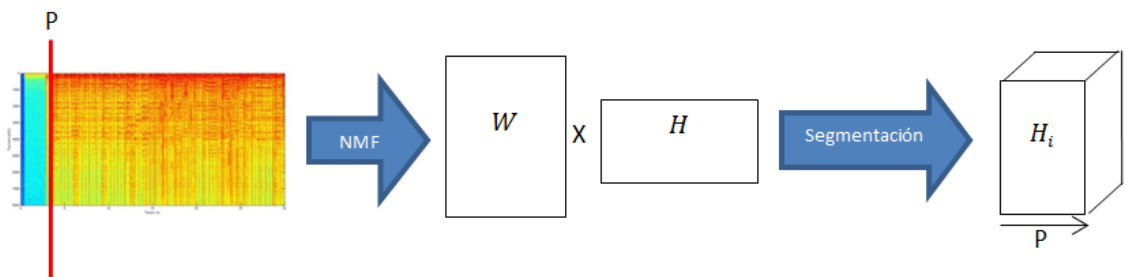


**Fig. 4.14 Realización de NMF con un número de bases de tamaño P o menor. Tras esto fragmentamos la matriz H en segmentos de tamaño P.**

Una vez hemos colocado los segmentos de forma que tengamos una matriz tridimensional tratamos de comprobar que bases se van repitiendo a lo largo de los distintos segmentos. Para ellos mediremos la energía de cada una de las bases en cada segmento. Esta medida de la energía se realiza elevando al cuadrado los valores de esta matriz  $H_i(k, t, N)$  dado que contiene valores de amplitud mientras que se buscan valores de energía. Una vez la matriz contenga valores de energía realizamos para cada base (filas de  $H_i(k, t, N)$ ) la suma de sus valores a lo largo de la duración del segmento (a lo largo de las columnas), es decir, desde el primer frame de tiempo hasta el frame que ocupe la posición P. Repitiendo este procedimiento para cada segmento llegaremos a obtener algo parecido a la siguiente figura:

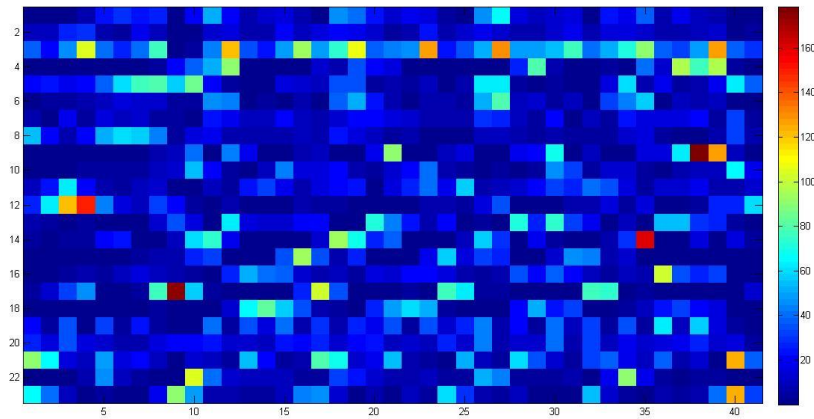


**Fig. 4.15** Resultado obtenido tras el cálculo de las energías de las diferentes bases. Cada bin muestra la energía de una determinada base en un segmento.

Esta figura contiene un ejemplo en el que puede apreciarse la información sobre la energía que poseen cada una de las 23 bases (eje Y) en cada uno de los 41 segmentos (eje X) en los que se ha dividido la matriz de activaciones de esta señal en concreto.

El objetivo para seleccionar unas determinadas bases consiste en el cálculo de la varianza de la parte alterna de cada una de las bases. Gracias a esto podremos ver cuales se repiten a lo largo del tiempo y cuáles no. Para ello comenzamos trabajando con la amplitud de cada una de las bases por lo que debemos aplicar la raíz cuadrada a la información contenida en la matriz de energías previamente calculada.

Una vez tengamos nuestra información sobre la amplitud de las bases a lo largo de los segmentos será necesario llevar a cabo una normalización de la misma dividiendo entre la energía que posee cada base en todo el tiempo que se encuentra activa, es decir, dividimos cada uno de los bins de la matriz de amplitudes entre la suma de su energía a lo largo del tiempo.

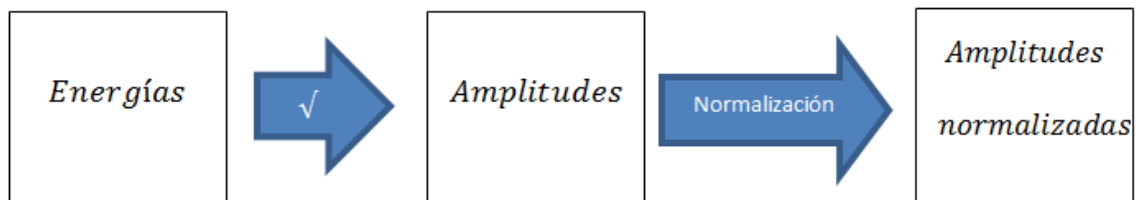


**Fig. 4.16** Tras tener la información sobre la energía de las bases aplicamos la raíz cuadrada para obtener información sobre amplitud y normalizamos.

Una sencilla forma de calcular la parte alterna de las bases consiste en restar a la amplitud normalizada de estas (ya calculadas en el paso previo) su media, ya que recordemos que la parte alterna de una señal puede calcularse como el total de la señal menos su parte continua (calculada como la media).

En este momento ya conocemos la parte alterna de cada una de las bases, es el momento de calcular la varianza de estas. Una vez realizado este cálculo tendremos algo parecido a esto:

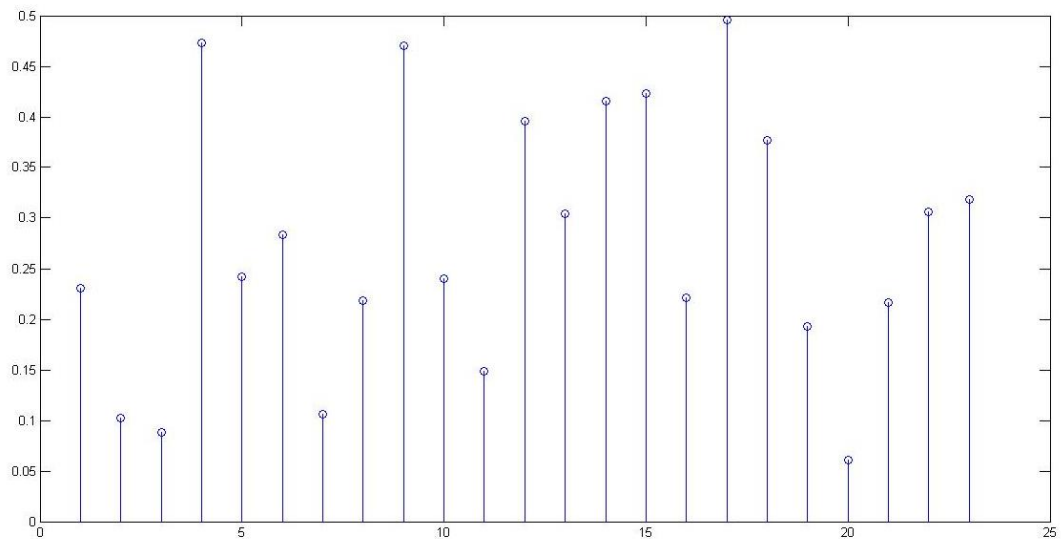


**Fig. 4.17** Varianza de cada una de las bases.

Donde cada uno de los puntos representa la varianza de las diferentes bases. Para poder seleccionar las bases que más nos interesan las ordenaremos de menor a mayor varianza y seleccionamos un conjunto de ellas empezando por la de menor varianza.

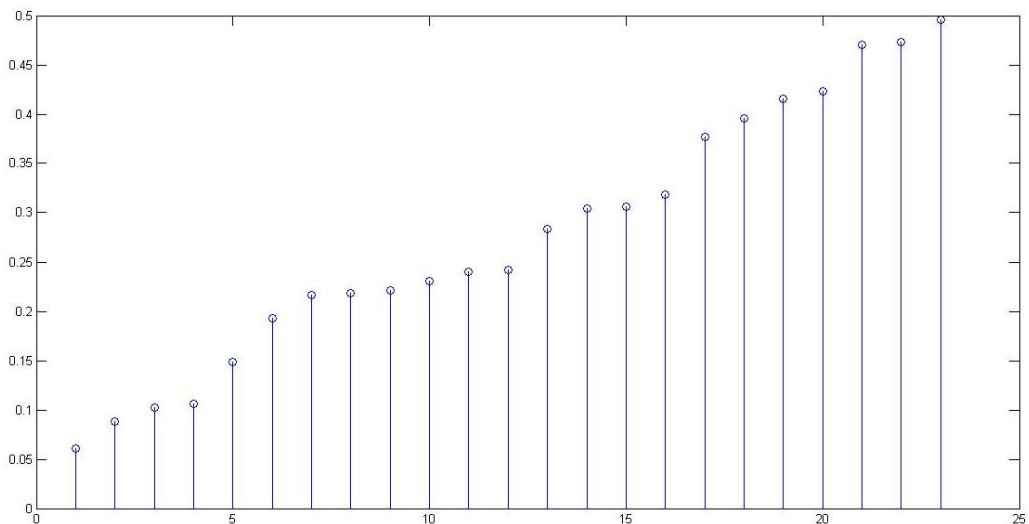


**Fig. 4.18** Representación de las diferentes bases ordenadas según su varianza de menor a mayor.

La razón por la que se escogen aquellas bases que presentan una menor varianza va ligada a la hipótesis de partida la cual nos indica que la pista rítmica de una canción va ligada a aquellos sonidos que presentan una repetitividad constante a lo largo de toda la señal. En términos de varianza, una base que aparezca a lo

largo de la mayor parte de los segmentos presentará una menor varianza (puesto que aparece muy a menudo) que una base que únicamente aparezca o se active en un pequeño grupo de segmentos. Ese podría ser el caso de una base que modele el sonido de la voz pronunciando una determinada palabra o frase, al no repetirse esta, la varianza que presenta será muy elevada y por tanto no resulta de interés para la reconstrucción de la pista rítmica de nuestra señal.

Por tanto, para reconstruir la pista rítmica de nuestra señal nos serviremos de un determinado grupo de bases, estas serán las  $N$  bases que presenten una menor varianza. Una vez seleccionadas dichas bases tan sólo será necesario multiplicar las componentes de  $W$  y  $H$  correspondientes a dichas bases. Tras esto crearemos las máscaras correspondientes. Una vez tengamos nuestro espectrograma correspondiente a la pista rítmica devolveremos la señal al dominio temporal mediante la ISTFT.

# Capítulo 5

## 5. Evaluación

Para la evaluación de los datos se van a elaborar unas bases de datos sobre las que realizaremos una serie de medidas. Estas medidas serán SDR, SIR y SAR, cuyo significado se explica en apartados posteriores. El procedimiento de obtención de las dos bases de datos se explica en la sección 5.1, las medidas utilizadas así como su significado y modo de obtención vienen detalladas en la sección 5.2. Una vez obtenidos todos los resultados se realizará un proceso de optimización para establecer qué parámetros permiten maximizar la calidad de la separación de audio (sección 5.3) y finalmente obtendremos unos resultados finales, los cuales se explican en la sección 5.4.

Las señales utilizadas han sido muestreadas con una frecuencia de muestreo  $f_s = 16000 \text{ Hz}$ . Esta frecuencia de muestreo nos permite obtener señales con una calidad suficiente para ser escuchadas y a su vez no produce un número de muestras demasiado elevado, lo que conllevaría un coste computacional muy elevado. Para la realización de la *STFT (Short-Time Fourier Transform)* utilizaremos una ventana Hanning de 1024 muestras con un solapamiento de  $N/2$ , es decir, 512 muestras de solapamiento. El uso de este tamaño de ventanas nos permite trabajar con señales de una duración de 64 mS, tamaño suficientemente pequeño como para considerar estacionaria una señal de audio.

### 5.1. Bases de datos

Para la creación de las bases de datos, es necesario disponer primero de las señales que van a integrarlas. Por tanto, describimos el procedimiento seguido en la obtención de estas señales. Como punto de partida tendremos las distintas pistas por separado, generalmente una pista de guitarra, una de bajos, otra de voz, una de bombo y platillos entre otras. Estas pistas se unen de forma que se obtiene un total de tres señales diferentes cuyo contenido será:

- La primera señal corresponde a la suma de todas las pistas, obteniendo como resultado la señal completa, es decir, la canción completa con todos los instrumentos, pista vocal y efectos, se la conoce como señal mezcla.
- Se obtiene una segunda señal correspondiente a la pista rítmica de la señal (señal rítmica). Para ello se han utilizado exclusivamente aquellas pistas

correspondientes a instrumentos percusivos aunque también se utiliza la pista del bajo ya que, como ya dijimos previamente, este instrumento generalmente también indica el ritmo de la canción.

- Por último, mediante la unión del resto de pistas, es decir, aquellas que no han sido utilizadas en la obtención de la pista rítmica, obtenemos una tercera señal que contiene la información sobre la pista no rítmica de nuestra canción (señal no rítmica).

En este momento disponemos de las señales con las que vamos a trabajar, por tanto, vamos a describir la construcción de las dos bases de datos. Estas consisten en una base de datos de optimización cuyo objetivo es darnos a conocer los parámetros que optimizan los resultados obtenidos en cada uno de los sistemas implementados, la segunda, consiste en una base de datos de testeo sobre la que comprobaremos la veracidad y utilidad de los parámetros obtenidos en la base de datos de optimización.

### 5.1.1. Base de datos de optimización

Esta primera base de datos busca la optimización de los resultados utilizando todas las posibles combinaciones de los parámetros que gobiernan cada uno de los sistemas implementados.

El procedimiento a seguir para la implementación de esta base de datos ha consistido en la utilización de un total de 20 señales de audio monoaurales, polifónicas y multitímbricas. Estas 20 señales consisten en fragmentos de 30 segundos de duración de las siguientes canciones:

| Identificador | Artista                            | Título                      |
|---------------|------------------------------------|-----------------------------|
| O1            | Iron Maiden                        | 2 Minutes To Midnight       |
| O2            | Lenny Kravitz                      | Are You Gonna Go My Way     |
| O3            | Beck                               | Gamma Ray                   |
| O4            | Blur                               | Song 2                      |
| O5            | The Smashing Pumpkins              | Bullet With Butterfly Wings |
| O6            | Dinosaur Jr.                       | Fell The Pain               |
| O7            | Fleetwood Mac                      | Go Your Own Way             |
| O8            | Bob Seger & The Silver Bullet Band | Hollywood Nights            |
| O9            | Eagles                             | Hotel California            |
| O10           | John Mellencamp                    | Hurts So Good               |
| O11           | Coldplay                           | In My Place                 |
| O12           | MC5's Wayne Kramer                 | Kick Out The James          |
| O13           | Los Lobos                          | La Bamba                    |

|     |                         |                     |
|-----|-------------------------|---------------------|
| O14 | Bon Jovi                | Livin On A Prayer   |
| O15 | Queens Of The Stone Age | Make It Wit Chu     |
| O16 | Santana                 | No One To Depend On |
| O17 | Blondie                 | One Way Or Another  |
| O18 | Johnny Cash             | Ring Of Fire        |
| O19 | Lostprophets            | Rooftops            |
| O20 | Joe Satriam             | Satch Boogie        |

**Tabla 1: Base de datos de optimización, la primera columna indica el identificador de la señal, la segunda nos informa del autor de la canción y la tercera del título de la misma.**

\*Todas estas señales provienen de los videojuegos Guitar Hero 5 y Guitar Hero World Tour: <https://www.guitarhero.com/es/>

### 5.1.2. Base de datos de testeo

La base de datos de testeo está integrada por un conjunto de 9 señales sobre las que se comprobarán los resultados obtenidos en la base de datos de optimización. Una vez obtenidos los parámetros que optimizan los resultados, estos se aplican sobre la base de datos de testeo para comprobar el correcto funcionamiento de los sistemas implementados.

Esta base de datos se compone de 9 señales monoaurales, polifónicas y multitímbricas correspondientes a fragmentos de 30 segundos de duración de las siguientes canciones:

| Identificador | Artista                 | Título            |
|---------------|-------------------------|-------------------|
| T1            | Bullet For My Valentine | Scream Aim Fire   |
| T2            | Coldplay                | Shiver            |
| T3            | Kiss                    | Shout It Loud     |
| T4            | The Police              | So Lonely         |
| T5            | Black Label Society     | Still Born        |
| T6            | Dire Straits            | Sultan Of Swing   |
| T7            | Metallica               | Trapped Under Ice |
| T8            | Queen & David Bowie     | Under Pressure    |
| T9            | Deep Purple             | Woman From Tokyo  |

**Tabla 2: Base de datos de testeo, la primera columna indica el identificador de la señal, la segunda nos informa del autor de la canción y la tercera del título de la misma.**

\*Todas estas señales provienen de los videojuegos Guitar Hero 5 y Guitar Hero World Tour: <https://www.guitarhero.com/es/>

Sobre estas señales se realizarán los algoritmos de extracción de la pista rítmica utilizando los parámetros (óptimos) obtenidos mediante la base de datos de optimización comprobándose la validez de los resultados obtenidos con otro tipo de señales diferentes a aquellas con las que estos parámetros han sido optimizados.

Estos resultados se muestran en la sección 5.4.

## 5.2. Métricas

Para la evaluación de los sistemas implementados se utilizarán un conjunto de métricas ampliamente utilizadas en el ámbito de la separación de fuentes sonoras. Estas métricas son descritas en profundidad en el documento *Performance Measurement in Blind Audio Source Separation* [25]. En este documento se describen las medidas *SDR* (*Source to Distortion Ratio*), *SIR* (*Source to Interferences Ratio*), *SNR* (*Source to Noise Ratio*) y *SAR* (*Source to Artifacts Ratio*). Las medidas de SDR, SIR y SAR consisten en un tipo de medidas cuantitativas y no subjetivas por lo que no dependen del oyente ni de la persona que realice la medida.

Existen otro tipo de medidas como pueden ser la *Interferencia entre Símbolos* (*Inter-Symbol Interference, ISI*) o *D*, que consiste en la comparación directa de la señal reconstruida comparándola con la señal original. Este tipo de medidas sufren serias limitaciones, dado que no tienen en cuenta otros tipos de distorsiones que pueden ser de mayor o menor importancia en ciertos casos. Por ejemplo, en aplicaciones musicales hi-fi la distorsión que modifica el timbre de los instrumentos prima sobre otro tipo de distorsión, mientras que en aplicaciones de habla se da una mayor importancia a la distorsión provocada al realizar ciertos filtrados que provocan la pérdida de inteligibilidad del habla. Las dos medidas descritas previamente no tienen en cuenta diferencia alguna entre estos tipos de distorsiones, es por ello que se recurre a otro tipo de medidas [25].

Para aplicaciones de audio, es importante medir de forma separada la cantidad de interferencias proveniente de fuentes no deseadas, la cantidad de ruido proveniente del sensor y la cantidad de ruido por “artefactos” (debido a una mala reconstrucción de la señal). Estos artefactos son considerados más molestos que las interferencias y estas, a su vez, son más molestas que el ruido proveniente del sensor.

Si consideramos la señal reconstruida como  $x_{rec}$ , podemos descomponer esta en la suma de los siguientes términos:

$$x_{rec} = x_{dist} + e_{interf} + e_{ruido} + e_{artef} \quad (5.1)$$

Donde  $x_{dist}$  consiste en la versión distorsionada de  $x$  (señal original) y  $e_{interf}$ ,  $e_{ruido}$  y  $e_{artef}$  son los términos del error introducido por interferencias, ruido y artefactos respectivamente. Estos cuatro términos representan el ruido de

sensor, la parte de señal que proviene de la fuente deseada así como de otras fuentes no deseadas y otro tipo de distorsiones.

A continuación se calcularán diferentes proporciones de energía para evaluar qué cantidad de estos cuatro términos se encuentra en la señal reconstruida.

En [25] se explica en detalle el cálculo de los parámetros  $x_{dist}$ ,  $e_{interf}$ ,  $e_{ruido}$  y  $e_{artef}$ . Una vez obtenidos dichos parámetros, se muestran las ecuaciones seguidas para la obtención de las medidas expuestas al inicio de este apartado:

$$SDR = 10 \log_{10} \left( \frac{\|x_{dist}\|^2}{\|e_{interf} + e_{ruido} + e_{artef}\|^2} \right) \quad (5.2)$$

$$SIR = 10 \log_{10} \left( \frac{\|x_{dist}\|^2}{\|e_{interf}\|^2} \right) \quad (5.3)$$

$$SNR = 10 \log_{10} \left( \frac{\|x_{dist} + e_{interf}\|^2}{\|e_{ruido}\|^2} \right) \quad (5.4)$$

$$SAR = 10 \log_{10} \left( \frac{\|x_{dist} + e_{interf} + e_{ruido}\|^2}{\|e_{artef}\|^2} \right) \quad (5.5)$$

Para la obtención de estas medidas, los autores se basan en la definición de SNR introduciendo una serie de modificaciones.

Estas medidas pueden ser definidas de forma básica del siguiente modo:

- SDR: nos ofrece una medida general de la calidad de la señal. Como se aprecia en (5.2) la forma de obtener esta medida consiste en dividir la señal que hemos obtenido tras la separación entre la suma de todas las posibles fuentes de ruido. De este modo, un SDR elevado nos indica que la señal obtenida posee mayor energía que la que poseen el ruido, interferencias y artefactos obtenidos y por tanto, se tratará de una señal limpia.

- SIR: esta medida se obtiene gracias a la división de la señal reconstruida entre las interferencias generadas (5.3). Es un modo de medir la cantidad de señal no deseada que tenemos. Así por ejemplo el SIR de nuestra pista rítmica nos indica la cantidad de señal no rítmica que no hemos sido capaces de separar. Es por ello que es conveniente tener un SIR elevado.
- SAR: es una medida de la calidad de la reconstrucción de nuestra señal que no tiene en cuenta si se produce una buena o mala separación de la misma. Se obtiene dividiendo la señal separada más el ruido proveniente de la fuente y de la otra señal entre el ruido por artefactos (5.5). Un SAR elevado implica una buena reconstrucción de la señal aunque la separación llevada a cabo no haya sido buena.

Una vez conocidos los parámetros de medida con los que se va a trabajar para comprobar la calidad de la separación y reconstrucción llevadas a cabo, es conveniente explicar que los principales parámetros que han de tenerse en cuenta son el SDR y el SIR puesto que el SAR no nos aporta información sobre la calidad de la separación. Es por ello que ha de buscarse aquella combinación de parámetros de las bases de datos que ofrezcan un SDR y SIR óptimos.

Habrà de llegarse a un compromiso puesto que existiràn combinaciones que ofrezcan unos valores de SDR muy elevados a costa de un SIR bajo o viceversa, por ello se recomienda buscar un punto en el que ambas medidas ofrezcan un nivel de calidad suficiente.

## 5.3. Optimización

Una vez obtenidas nuestras bases de datos y descritos los parámetros que miden la calidad de nuestras reconstrucciones (SDR, SIR y SAR) tratamos de buscar que valores de las variables de cada sistema nos ofrecen un mejor resultado.

Este paso es muy sencillo dado que, gracias a las métricas descritas en la sección 5.2, tenemos una forma sencilla de cuantificar la calidad del audio obtenido con la reconstrucción.

Este procedimiento de optimización consiste en realizar un barrido entre todas las combinaciones posibles de los parámetros que manejan cada uno de los sistemas implementados. El número de ejecuciones realizadas así como otros datos de interés se describen a continuación para cada sistema. Este proceso se realiza usando los 20 ficheros contenidos en la base de datos de optimización (identificadores O1  $\rightarrow$  O20).

En concreto nos centraremos en los resultados obtenidos tras realizar esta base de datos sobre los sistemas implementados en las secciones 4.2.1 (REPET + NMF modificado), 4.2.2 (ELLIS + NMF modificado) y 4.3 (Selección de bases con menor varianza). Esto es así dado que el sistema implementado en la sección 4.1 (Hipótesis inicial) implicaba un coste computacional elevadísimo junto con unos resultados prácticamente inteligibles. A pesar de ello, sirvió como punto de partida del resto de sistemas cuyos resultados, como se comprueba a continuación, son mucho mejores.

Para los sistemas REPET + NMF modificado y ELLIS + NMF modificado se disponían de tres parámetros que pueden variar y que ofrecerán diferentes resultados para cada canción, estos son:

- **k1**: número de bases de la pista rítmica. Indica el número de componentes que van a utilizarse para la reconstrucción de la pista rítmica de la señal. Se define a la hora de ejecutar el NMF inicial que servía de inicialización para  $W_r$ .
- **k2**: número de bases de la pista no rítmica. Indica el número de componentes que se utilizarán para la reconstrucción de la pista no rítmica de la señal.
- **$\beta$** : parámetro que nos indica el tipo de divergencia (función de coste) utilizada, tenemos las siguientes posibilidades:
  - $\beta = 0$ , para la función de coste Itakura Saito.
  - $\beta = 1$ , para la función de coste Kullback Leibler.

- $\beta = 2$ , para la función de coste de Distancia Euclídea.  
Aunque el parámetro  $\beta$  puede oscilar entre 0 y 2.

Los valores que han tomado los diferentes parámetros para la búsqueda de un SDR óptimo son los siguientes:

- $k_1 \rightarrow [1\ 3\ 5\ 7\ 9\ 10\ 12\ 15\ 20\ 25]$
- $k_2 \rightarrow [5\ 7\ 9\ 12\ 15\ 18\ 20\ 25\ 30\ 40\ 50\ 75]$
- $\beta \rightarrow [1\ 2]$

Esto hace un total de 240 posibles combinaciones ( $10\ (k_1) \times 12\ (k_2) \times 2\ (\beta)$ ) por cada uno de los 20 ficheros de audio utilizados en la base de datos de optimización. Será necesario realizar un total de 4800 ejecuciones para cada uno de estos dos sistemas. Por cada una de estas ejecuciones se obtienen dos ficheros de audio: uno correspondiente a la pista rítmica de la señal y un segundo correspondiente a la pista no rítmica de la misma.

Como se puede comprobar, el volumen de datos que se generan y que se deberá analizar es bastante elevado. Para realizar este barrido entre las combinaciones de los parámetros ha sido necesario invertir aproximadamente unas 16 horas de ejecución de código en el entorno MATLAB para cada uno de estos dos primeros sistemas.

El tercer sistema (Selección de bases con menor varianza) dispone de tres parámetros ajustables, estos son:

- **K**: número de bases con el que ejecutar el algoritmo NMF. A diferencia de los sistemas previos, este trabaja con el algoritmo NMF sin modificar, por tanto, no es necesaria la particularización entre bases para la pista rítmica ( $k_1$ ) y bases para la pista no rítmica ( $k_2$ ). Por tanto, este parámetro hace referencia al número de bases total.
- **N**: número de bases con el que realizar la reconstrucción de la pista rítmica. Una vez realizado el algoritmo NMF y haber calculado la varianza de las bases tendremos que seleccionar un determinado número de estas (aquellas cuya varianza sea menor). Este parámetro indica el número de bases que seleccionaremos del total ( $K$ ) para reconstruir nuestra pista rítmica.
- **$\beta$** : parámetro que nos indica el tipo de divergencia (función de coste) utilizada.

Para la realización de esta base de datos estos parámetros han tomado los siguientes valores:

- $K \rightarrow [P, 2*P]$ . (siendo P el valor indicado por el algoritmo REPET). Para realizar el barrido y buscar aquellos parámetros que optimicen los resultados obtenidos el parámetro K (nº de bases con el que realizar el NMF) podrá tomar dos posibles valores. Estos son P, es decir, realizaremos el NMF con un nº de bases igual al nº de frames de tiempo que nos indique el algoritmo REPET y como segundo posible valor K será igual a 2P, es decir, realizamos un NMF con el doble de bases. Realizamos el barrido con estos dos posibles valores para comprobar si el hecho de usar un mayor o menor número de bases influye en la calidad de los resultados obtenidos.

- $N \rightarrow [1\ 2\ 3\ 4\ \dots\ \min(P)]$ .

Dado que la base de datos de optimización posee 20 canciones diferentes, cada una de ellas presentará un valor de P diferente (en función de la velocidad del ritmo que posea cada canción), para poder ejecutar el algoritmo de forma correcta debemos buscar el mínimo valor de esos 20 P y este marcará el máximo N posible con el que reconstruir la pista rítmica.

El hecho de usar el valor mínimo de los 20 posibles P que tenemos se puede apreciar con el siguiente ejemplo: supongamos que tras ejecutar el algoritmo REPET sobre las 20 señales de nuestra base de datos de optimización obtenemos un P mínimo de 15 (siendo este valor el mínimo de los 20 valores que obtenemos, uno por cada señal). Esto implica que una de esas 20 señales (aquella que haya obtenido un P igual a 15) realizará el algoritmo NMF con un total de 15 bases (tal y como se describe en el punto anterior). Para comprobar que valor de N obtiene unos mejores resultados comenzaríamos a reconstruir la pista rítmica con la base que tenga menor varianza, luego con las dos bases que tengan menor varianza, luego con las 3 y así sucesivamente hasta que reconstruyamos la pista rítmica utilizando el total de las bases. Siguiendo este ejemplo, si no respetásemos el hecho de que N se mueva entre  $[1\ 2\ 3\ 4\ \dots\ \min(P)]$  y tratásemos de reconstruir la pista rítmica con un número de bases superior a 15 el sistema daría un error puesto que el algoritmo NMF se ha realizado con 15 bases y resulta del todo imposible usar 16 o más dado que no existen tantas bases.

- $\beta \rightarrow [1\ 2]$ .

Utilizamos las funciones de coste Euclidea y Kullback Leibler.

Para cada una de las posibles combinaciones obtendremos los valores de SDR de cada una de las 20 señales. Se realiza a continuación la media de estos valores 20 valores obteniéndose de este modo un único valor de SDR que irá asociado a una determinado combinación de los parámetros de cada sistema.

Consideraremos como valores óptimos aquellos que maximicen el valor de SDR obtenido.

Tras realizar el proceso de optimización sobre la base de datos previamente diseñada podemos apreciar los siguientes resultados:

|     | REPET + NMF | ELLIS + NMF | Menor varianza |
|-----|-------------|-------------|----------------|
| SDR | 7.5204 dB   | 7.4894 dB   | 6.8334 dB      |

**Tabla 3: Resultados de SDR obtenidos con los parámetros óptimos de cada sistema.**

Donde los parámetros óptimos que se han obtenido para cada uno de los algoritmos serían:

|             | k1 | k2 | $\beta$ |
|-------------|----|----|---------|
| REPET + NMF | 10 | 12 | 2       |

**Tabla 4: Valores de los parámetros que optimizan los resultados del sistema REPET + NMF modificado.**

|               | k1 | k2 | $\beta$ |
|---------------|----|----|---------|
| REPET + ELLIS | 25 | 15 | 2       |

**Tabla 5: Valores de los parámetros que optimizan los resultados del sistema ELLIS + NMF modificado.**

|                | k | N           | $\beta$ |
|----------------|---|-------------|---------|
| Menor varianza | P | 12, 13, 14* | 1       |

**Tabla 6: Valores de los parámetros que optimizan los resultados del sistema de selección de bases con menor varianza.**

\* NOTA: tras realizar la búsqueda de óptimos varias veces, los resultados más repetidos eran estos tres.

Las siguientes imágenes muestran los resultados obtenidos en lo que a SDR respecta:

Esta primera imagen corresponde a los resultados obtenidos utilizando el primer sistema implementado (REPET + NMF modificado). En ella puede comprobarse la variación del SDR para los distintos K1 y K2. Esta imagen se ha construido manteniendo el parámetro  $\beta = 2$  puesto que es el que mejores resultados ofrece. Se puede comprobar como tenemos un SDR máximo para en la fila 4 y columna 6, si recordamos que  $k1 \rightarrow [1 \ 3 \ 5 \ 7 \ 9 \ 10 \ 12 \ 15 \ 20 \ 25]$  y  $K2 \rightarrow [5 \ 7 \ 9 \ 12 \ 15 \ 18 \ 20 \ 25 \ 30 \ 40 \ 50 \ 75]$  tenemos que el máximo se da en  $K1 = 10, K2 = 12$  y  $\beta = 2$ . Se comprueba que el resultado coincide con lo expuesto en la tabla 4.

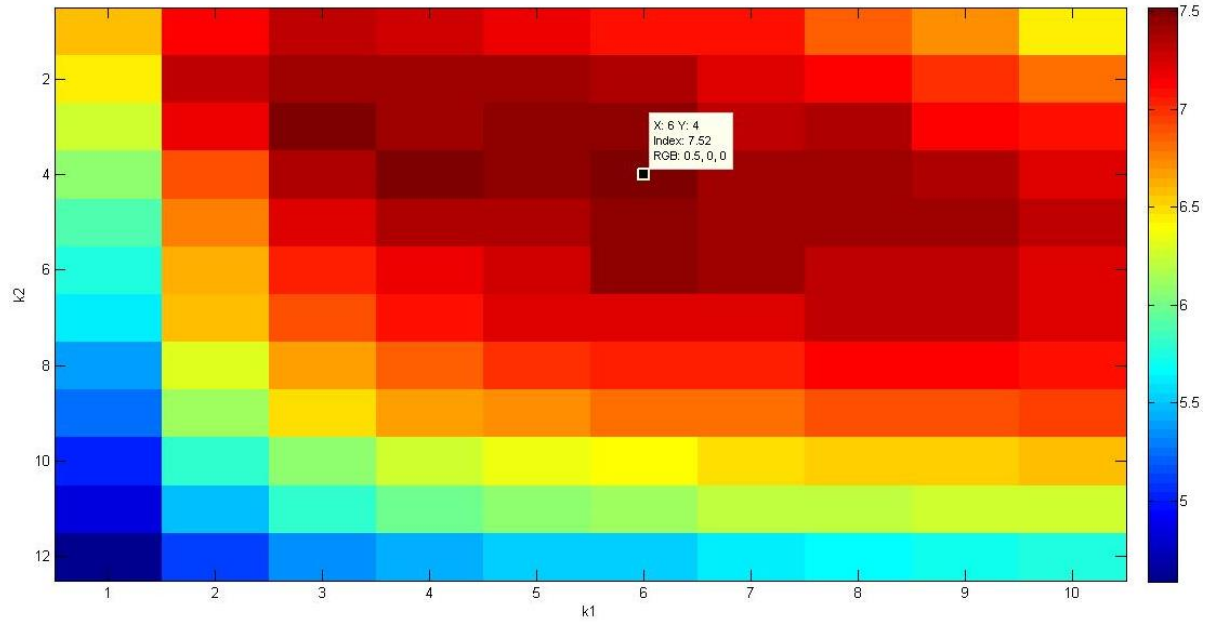


**Fig. 5.1** Imagen que contiene los resultados de SDR para el sistema REPET + NMF modificado con  $\beta = 2$ .

Esta imagen corresponde a los resultados obtenidos utilizando el segundo sistema implementado (ELLIS + NMF modificado). Se construye del mismo modo que la anterior y se puede comprobar como tenemos un SDR máximo para en la fila 5 y columna 10, si recordamos que  $k_1 \rightarrow [1 \ 3 \ 5 \ 7 \ 9 \ 10 \ 12 \ 15 \ 20 \ 25]$  y  $K_2 \rightarrow [5 \ 7 \ 9 \ 12 \ 15 \ 18 \ 20 \ 25 \ 30 \ 40 \ 50 \ 75]$  tenemos que el máximo se da en  $K_1 = 15, K_2 = 25$  y  $\beta = 2$ . Se comprueba que el resultado coincide con lo expuesto en la tabla 5.

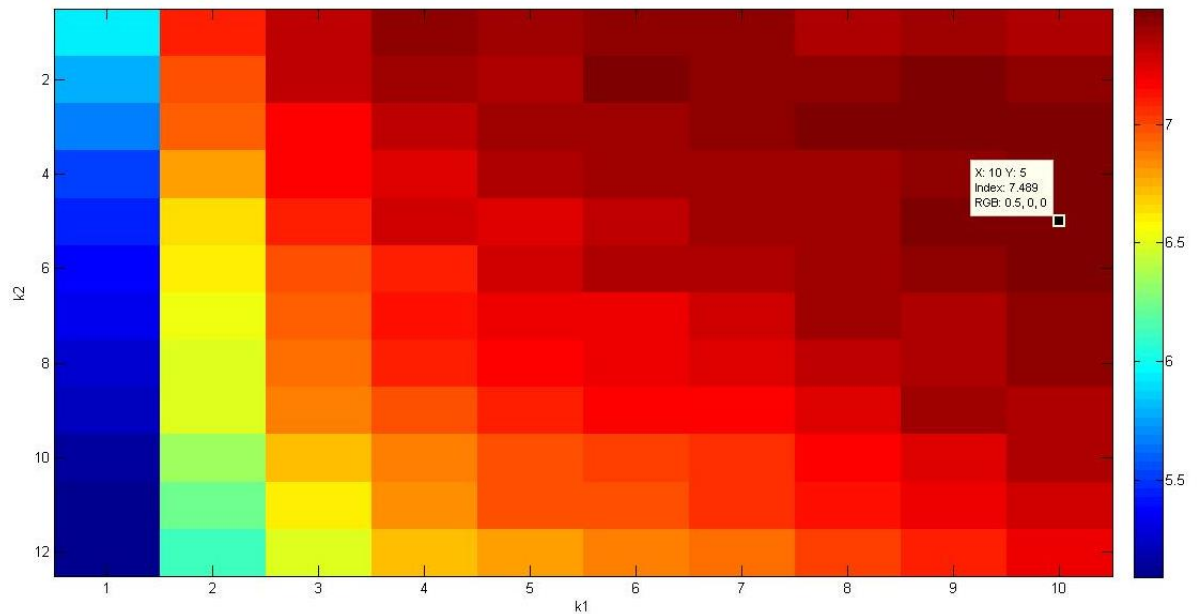


**Fig. 5.2** Imagen que contiene los resultados de SDR para el sistema ELLIS + NMF modificado con  $\beta = 2$ .

Esta imagen corresponde a los resultados obtenidos utilizando el tercer sistema implementado (Selección de bases con menor varianza). Se puede comprobar como tenemos un SDR máximo para en la fila 1 y columna 14, si recordamos que  $K \rightarrow [P, 2*P]$  y  $N \rightarrow [1 \ 2 \ 3 \ 4 \ \dots \ \min(P)]$  tenemos que el máximo se da en  $K = P, N = 14$  y  $\beta = 1$ . Se comprueba que el resultado coincide con lo expuesto en la tabla 6.

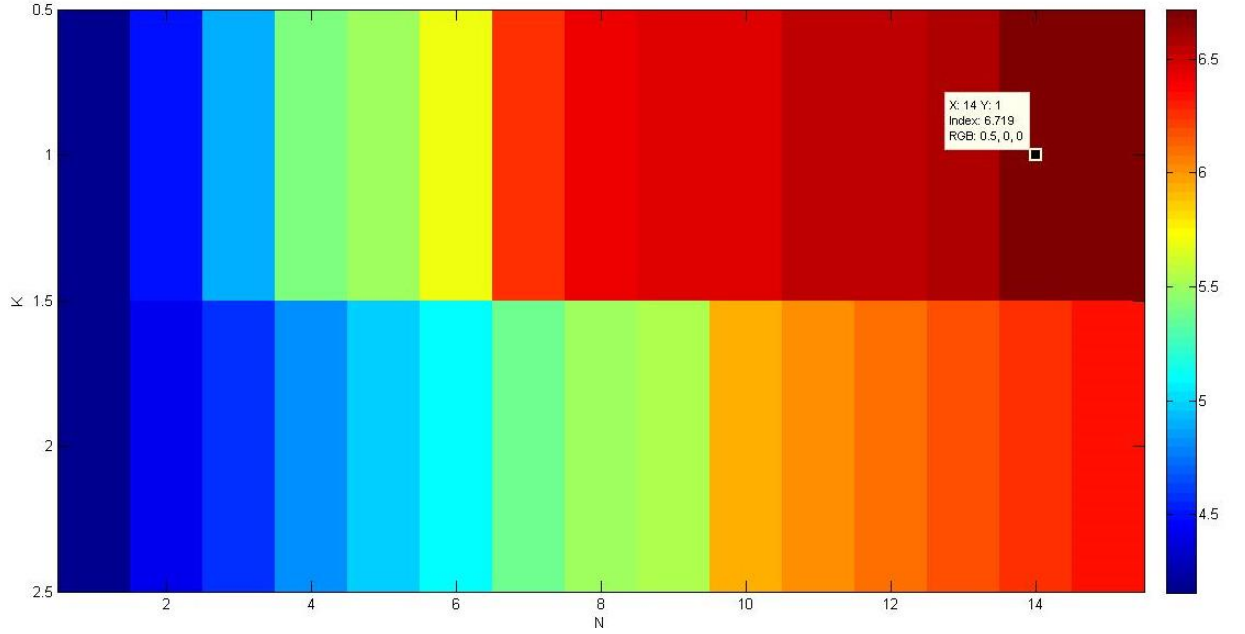


**Fig. 5.3** Imagen que contiene los resultados de SDR para el sistema de Selección de bases con menor varianza con  $\beta = 1$ .

## 5.4. Resultados

En esta sección nos dedicaremos a mostrar los resultados obtenidos tras el uso de los diversos algoritmos implementados.

La siguiente tabla muestra los resultados generados por cada uno de los algoritmos utilizándose sobre las diferentes canciones de la base de datos de testeo con los parámetros óptimos, estos algoritmos son:

- 1) REPET + NMF modificado
- 2) ELLIS + NMF modificado.
- 3) Selección de bases con menor varianza.

| Canción | REPET + NMF |         |         | ELLIS + NMF |         |         | Menor varianza |         |         |
|---------|-------------|---------|---------|-------------|---------|---------|----------------|---------|---------|
|         | SDR         | SIR     | SAR     | SDR         | SIR     | SAR     | SDR            | SIR     | SAR     |
| T1      | 7.5756      | 9.7962  | 12.1455 | 7.66        | 9.458   | 12.3166 | 7.4904         | 8.5176  | 13.7017 |
| T2      | 8.8334      | 12.1093 | 13.8666 | 8.5422      | 11.938  | 12.5503 | 8.122          | 12.9972 | 12.343  |
| T3      | 4.6369      | 5.9546  | 8.4429  | 4.6705      | 8.4513  | 6.8667  | 3.0071         | 8.6342  | 5.9012  |
| T4      | 9.0844      | 19.1708 | 11.7633 | 8.5996      | 18.0213 | 10.9755 | 9.8145         | 16.3869 | 12.6798 |
| T5      | 9.5651      | 14.0119 | 13.892  | 10.6099     | 13.0056 | 14.8359 | 9.6176         | 10.9016 | 16.064  |
| T6      | 6.8425      | 8.792   | 10.9218 | 6.8232      | 10.2005 | 10.0825 | 6.9497         | 8.584   | 12.1531 |
| T7      | 7.9336      | 9.3809  | 13.4341 | 7.3299      | 9.1483  | 11.7307 | 8.644          | 10.7221 | 14.2424 |
| T8      | 6.6021      | 13.8642 | 9.3701  | 5.892       | 12.5207 | 8.1935  | 5.3573         | 9.2566  | 8.8102  |
| T9      | 8.1173      | 12.8077 | 11.9084 | 8.5913      | 11.093  | 12.3889 | 3.367          | 11.8764 | 7.4415  |

**Tabla 7: Valores de las medidas tomadas (SDR, SIR y SAR) sobre las distintas canciones de la base de datos de testeo empleando los 3 sistemas implementados.**

Si realizamos la media de los resultados obtenidos tendremos que:

|       | REPET + NMF |         |         | ELLIS + NMF |         |         | Menor varianza |        |         |
|-------|-------------|---------|---------|-------------|---------|---------|----------------|--------|---------|
|       | SDR         | SIR     | SAR     | SDR         | SIR     | SAR     | SDR            | SIR    | SAR     |
| Media | 7.6879      | 11.7653 | 11.7494 | 7.6354      | 11.5374 | 11.1045 | 6.93           | 8.5176 | 11.4819 |

**Tabla 8: Media de las medidas representadas en la tabla 7.**

A la vista de los datos obtenidos tanto en la base de datos de optimización como en la base de datos de testeo podemos afirmar que el sistema REPET + NMF modificado es el que mejores resultados ofrece en lo que a medidas de SDR, SIR y SAR respecta.

Cabe destacar algunas curiosidades que pueden apreciarse en los resultados obtenidos. A pesar de que, por norma general, los resultados poseen una calidad

más que aceptable, es cierto que para determinadas señales, el resultado obtenido no es tan bueno como cabría esperar.

Es el ejemplo de la señal T3. Con un SDR de unos 4 dB menos que el resto de señales podemos encontrar una explicación a este hecho en el tipo de voz que se escucha en dicha señal así como en la repetitividad de la misma. Si se presta atención podemos comprobar como los golpes de voz coinciden con los golpes de batería por lo que los sistemas implementados no pueden realizar con claridad la separación ya que no son capaces de detectar si las bases que generan la voz pertenecen a la pista rítmica o a la pista no rítmica.

Los resultados obtenidos en T6 (algo más bajos que la media) posiblemente se deban al bajo tono de voz utilizado por el cantante. Las frecuencias más bajas, como ya se indicó previamente, suelen ir asociadas a la línea de bajos. Por esto, es posible que los sistemas implementados seleccionen las bases que generan la voz y las asignen a la pista rítmica de la señal. Al realizar las medidas pertinentes y comparar con la pista rítmica de partida (que no contiene la pista de voz) los resultados tienden a ser más bajos.

En T8 ocurre algo que también merece la pena resaltar. Durante una buena parte de la señal analizada, se produce una repetición de una señal de guitarra. Siguiendo la hipótesis de partida, los sistemas que hemos implementado consideran como parte de la pista rítmica esa señal de guitarra puesto que se va repitiendo a lo largo del tiempo y a corto plazo. Es por ello que esa señal se asigna a la pista rítmica y, por tanto, a la hora de realizar las medidas, el SDR es mucho menor puesto que en la pista rítmica original, esa señal de guitarra no existía.

## Capítulo 6

### 6. Conclusión y líneas futuras

Mediante la realización de este proyecto se ha comprobado la dificultad inherente a la separación de fuentes acústicas. El uso de señales monoaurales implica un menor conocimiento de la distribución espacial de los instrumentos y, por tanto, una mayor dificultad a la hora de realizar esta separación.

Tras la implementación de diversos sistemas que, cada uno a su manera, tratan de obtener la pista rítmica de las señales, podemos afirmar que la hipótesis de partida que asocia esta pista rítmica con aquellos sonidos que se repiten de forma cuasi-periódica a lo largo del tiempo es adecuada. A pesar de los diferentes resultados obtenidos todos los sistemas implementados permiten, en mayor o menor medida, extraer unas señales bastante fieles a la pista rítmica de estas.

Es importante destacar la fuerte dependencia que se tiene con el tipo de señal utilizado ya que la gran variedad de ritmos y de formas de marcarlos implica un amplísimo abanico de posibilidades. Esto conlleva que no todos los sistemas descritos en este documento ofrezcan un resultado positivo para todas las señales e incluso, un sistema que funcione correctamente con una determinada señal podría no ser de utilidad si no se utilizan los parámetros apropiados para su puesta en marcha.

Toda esta variedad de señales, sistemas implementados y parámetros utilizados por estos implican cierta dificultad a la hora de buscar la optimización de alguna señal.

Como se ha comprobado en la sección de resultados, el primer sistema ofrece los mejores resultados en lo que a las métricas seleccionadas (SDR, SIR y SAR) se refiere.

El último sistema descrito (Selección de bases con menor varianza) nos ofrece una solución rápida y con una eficacia bastante elevada a este problema. Aunque la separación realizada no sea perfecta, sí que posee una calidad suficiente como para detectar la pista rítmica, con el inconveniente de que un pequeño porcentaje de esta sigue manteniéndose en la pista no rítmica.

De cara a futuros estudios en este campo sería interesante profundizar en los sistemas descritos en las secciones 4.2.1 y 4.2.2 puesto que de encontrarse unos parámetros óptimos, los resultados obtenidos poseen una calidad considerable.

Con respecto al sistema implementado en la sección 4.3 una posible continuación consistiría en intentar clasificar las bases por si solas mediante restricciones de suavidad y “sparse”.

Otro posible camino en el que continuar la investigación podría ser la búsqueda de una matriz  $W_r$  tridimensional que nos indique cuales son las bases de la pista rítmica de cada uno de los segmentos en los que se ha dividido el espectrograma de partida y utilizar un  $H_r$  bidimensional, lo que consistiría en un proceso diferente al implementado en las secciones 4.2.1 y 4.2.2.

# Capítulo 7

## 7. Bibliografía

- [1] M. Recuero López. *Ingeniería Acústica*, 1999.
- [2] P Lindsay and D Norman. *Human information processing: An introduction to psychology*, 1977.
- [3] Perfecto Herrera-Boyer, Anssi Klapuri, and Manuel Davy. *Signal Processing Methods for Music Transcription*. Springer US, Boston, MA, 2006.
- [4] A. Klapuri. Sound onset detection by applying psychoacoustic knowledge. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1999.
- [5] JP Bello, L. Daudet, and MB Sandler. Automatic Piano Transcription Using Frequency and Time-Domain Information. *IEEE Transactions on Acoustic, Speech and Signal Processing*, 14(6):2242-2251, 2006.
- [6] E Zwicker and H Fastl, *Psychoacoustics, Facts and Models*. Springer, 1990.
- [7] T Rousing. *The Science of Sound*, 1990.
- [8] A Oppenheim. *Signals & Systems*, 1997.
- [9] E Wold, T Blum, D Keislar, and J Wheaton. Content-based Classification, Search and Retrieval of Audio. *IEEE Multimedia*, 3(3):27-36, 1996.
- [10] A Bregman. *Auditory Scene Analysis: The Perceptual Organization of Sound*. MIT Press, 1990.
- [11] A Eronen. Comparison of Features of Musical Instrument Recognition. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2001.
- [12] De Pedro, Dionisio. *Teoría completa de la música*. Real musical, 1990.
- [13] Toussaint, Godfried. *The geometry of musical rhythm*. CRC, 2013.

- [14] Nishikawa, T Abe, H Saruwatari, and H Shikano. Overdetermined blind source separation of real acoustic sounds based on multistage ICA using subarray processing. *Signal Processing and Information Technology, 2003. ISSPIT 2003. Proceedings of the 3rd IEEE International Symposium on, March 2003.*
- [15] D FitzGerald. Harmonic/Percussive separation using median filtering. *Proc. of the 13th Int. Conference on Digital Audio Effects*, 2010.
- [16] Z Rafii, B Pardo. Music/Voice Separation Using the Similarity Matrix.
- [17] Z Rafii, B Pardo. A Simple Music/Voice Separation Method Based on the Extraction of the Repeating Musical Structure.
- [18] P Smaragdis, B Raj, and M Shashanka. Supervise and Semi-Supervised Separation of Sounds from Single-Channel Mixtures.
- [19] M Goto, and Y Muraoka. A Beat Tracking System for Acoustic Signals of Music.
- [20] M Goto, and Y Muraoka. Music Understanding At The Beat Level – Real-Time Beat Tracking For Audio Signals –
- [21] M Goto, and Y Muraoka. Beat Tracking based on Multiple-agent Architecture – A Real-time Beat Tracking System for Audio Signals
- [22] M Goto, and Y Muraoka. A Real-time Beat Tracking System for Audio Signals.
- [23] M Goto. An Audio-based Real-time Beat Tracking System for Music With or Without Drum-sounds.
- [24] M Goto, and Y Muraoka. Real-time beat tracking for drumless audio signals: Chord change detection for musical decisions.
- [25] E Vincent, R Gribonval and C Févotte. *Performance Measurement in Blind Audio Source Separation. IEEE Transactions on Audio, Speech and Language Processing, Institute of Electrical and Electronics Engineers*, 2006.
- [26] J Yoo, M Kim, K Kang, and S Choi. Nonnegative Matrix Partial Co-Factorization for Drum Source Separation.
- [27] Daniel P.W. Ellis. Beat Tracking by Dynamic Programming.
- [28] J Barry. Polyphonic music transcription using independent component analysis. MSc, 2003.

- [29] M Plumbey. Adaptative lateral inhibition for non-negative ICA. *Proceedings of the International Conference on Independent Component Analysis and Blind Signal Separation (ICA)*, 2001.
- [30] M Plumbey. Algorithms for nonnegative independent component analysis. *IEEE Transactions on Neural Networks*, 14(3): 534-543, 2003.
- [31] From Learning Music to Learning to Separate. Paris Smaragdis. Mitsubishi Electric Research Laboratories
- [32] On the Use of Masking Filters in Sound Source Separation. Derry Fitzgerald and Rajesh Jaiswal. Dublin Institute of Technology.
- [33] Percussive/harmonic sound separation by non-negative matrix factorization with smoothness/sparseness constraints. Francisco Jesus Cañadas Quesada, Pedro Vera Candeas, Nicolas Ruiz Reyes, Julio Carabias Orti y Pablo Cabañas Molero.
- [34] Separación de fuentes sonoras en señales musicales. Francisco José Rodríguez Serrano.

# Capítulo 8

## 8. Manual de usuario

Una vez realizado el propósito de este proyecto, se detalla en este capítulo una sencilla interfaz de usuario que permite, de forma clara y sin necesidad de tener conocimientos sobre programación en lenguaje Matlab, realizar el procedimiento descrito a lo largo de este documento. En concreto, dado que se han implementado varios sistemas, se ha optado por trabajar en mayor profundidad con el último de ellos debido a los buenos resultados que ofrece así como por tratarse del sistema que menor coste computacional conlleva. Es por esta razón que la interfaz que se describe a continuación permite realizar este último algoritmo.

En primer lugar se muestra una imagen en la que se aprecia el aspecto que posee este interfaz al inicializarla.

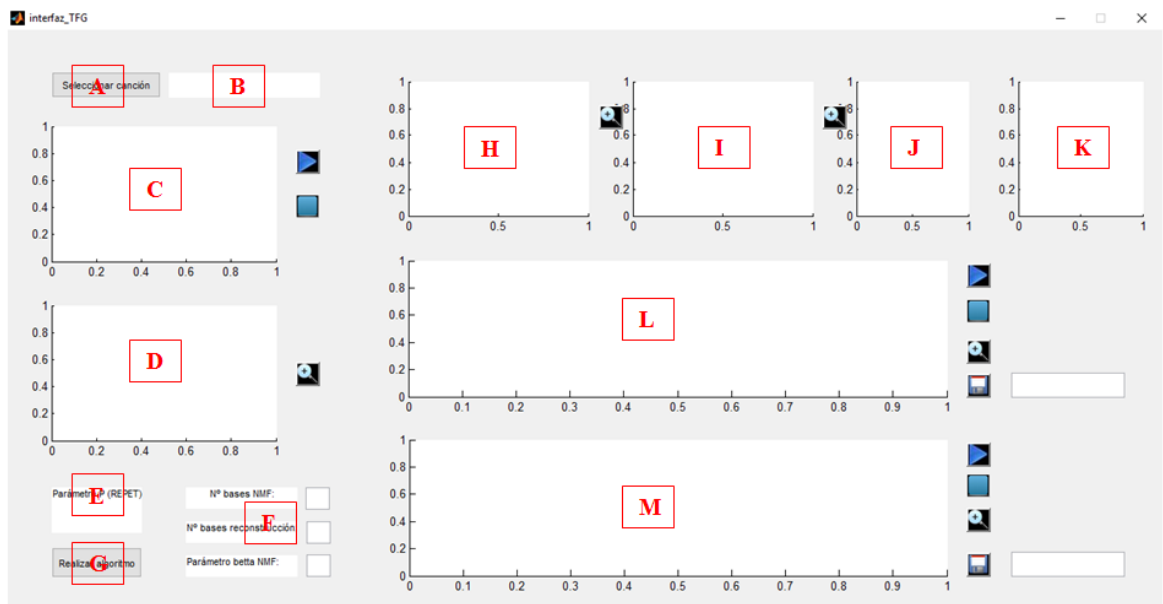


**Fig. 8.1** Aspecto inicial del interfaz.

Describimos a continuación cada una de las secciones que aparecen en este interfaz:

- Seleccionar canción: haciendo clic sobre este botón se abrirá un directorio a través del cual podremos seleccionar la señal sobre la que queremos realizar el algoritmo.
- En este cuadro de texto aparecerá el nombre del archivo previamente seleccionado.

- c) Representación temporal de la señal escogida. A la derecha tendremos dos botones, el botón superior nos permite reproducir la señal escogida mientras que el botón inferior detiene la reproducción de la misma.
- d) Espectrograma de la señal. A la derecha aparece un botón con una lupa que permite desplegar una nueva figura en la que se podrá observar este espectrograma con todos los detalles. El espectrograma se realiza mediante una FFT con los siguientes parámetros:
  - N° de muestras de la FFT: 1024.
  - Tipo de ventana: Hamming.
  - N° de muestras de solapamiento: 512.
- e) Parámetro P (REPET): al seleccionar la canción deseada mediante el botón descrito en a) se realiza automáticamente el algoritmo REPET. En este cuadro de texto se nos facilita el parámetro P devuelto por este algoritmo, esto nos sirve de indicador para escoger los parámetros con los que realizar el algoritmo más adelante.
- f) Estos tres cuadros de texto nos permiten introducir los parámetros deseados para la realización del algoritmo implementado en esta interfaz. Estos son:
  - Número de bases con el que realizar el algoritmo NMF.
  - Número de bases con el que reconstruir la pista rítmica.
  - Parámetro  $\beta$  con el que realizar el algoritmo NMF.
- g) Haciendo clic sobre este botón una vez se han introducido los parámetros descritos en f) comenzaremos a realizar el algoritmo, mostrándose los resultados en las siguientes secciones.
- h) Imagen que representa la matriz W (componentes). Haciendo clic sobre el botón de la derecha desplegaremos una nueva figura en la que se ve en profundidad esta misma imagen.
- i) Imagen que representa la matriz H (activaciones). Haciendo clic sobre el botón de la derecha desplegaremos una nueva figura en la que se ve en profundidad esta misma imagen.
- j) Imagen en la que se aprecia la divergencia del NMF realizado. En caso de no obtener un resultado satisfactorio esta imagen nos permite comprobar si el algoritmo NMF se ha realizado de forma correcta y ha convergido.

- k) Representación de la varianza de las diferentes bases a lo largo del tiempo. Nuestro algoritmo selecciona aquellas que menor varianza presentan.
- l) Espectrograma correspondiente a la pista rítmica de nuestra señal tras haberse llevado a cabo el algoritmo. A la derecha aparecen cuatro botones, estos permiten:
- Reproducir la pista rítmica generada por nuestro sistema.
  - Detener esta reproducción.
  - Desplegar una nueva figura que nos permita ver en detalle este espectrograma.
  - Guardar la señal generada en el directorio de trabajo que nos encontremos actualmente. Para ello será necesario introducir previamente el nombre que queremos dar al archivo generado. Este nombre se introduce en el cuadro de texto que aparece a la derecha del cuarto botón.
- m) Espectrograma correspondiente a la pista no rítmica de nuestra señal tras haberse llevado a cabo el algoritmo. A la derecha aparecen cuatro botones, estos permiten:
- Reproducir la pista rítmica generada por nuestro sistema.
  - Detener esta reproducción.
  - Desplegar una nueva figura que nos permita ver en detalle este espectrograma.
  - Guardar la señal generada en el directorio de trabajo que nos encontremos actualmente. Para ello será necesario introducir previamente el nombre que queremos dar al archivo generado. Este nombre se introduce en el cuadro de texto que aparece a la derecha del cuarto botón.

Conocidos los diferentes botones y gráficos que aparecerán en nuestro interfaz vamos a realizar un breve tutorial sobre el procedimiento que se debe seguir para el uso del mismo.

En primer lugar se debe iniciar el interfaz y, haciendo clic sobre el botón de “Seleccionar canción”, elegir la señal sobre la que queramos trabajar. Es importante destacar que este sistema trabaja con archivos en formato .wav por lo que para otros formatos será necesaria una conversión previa.

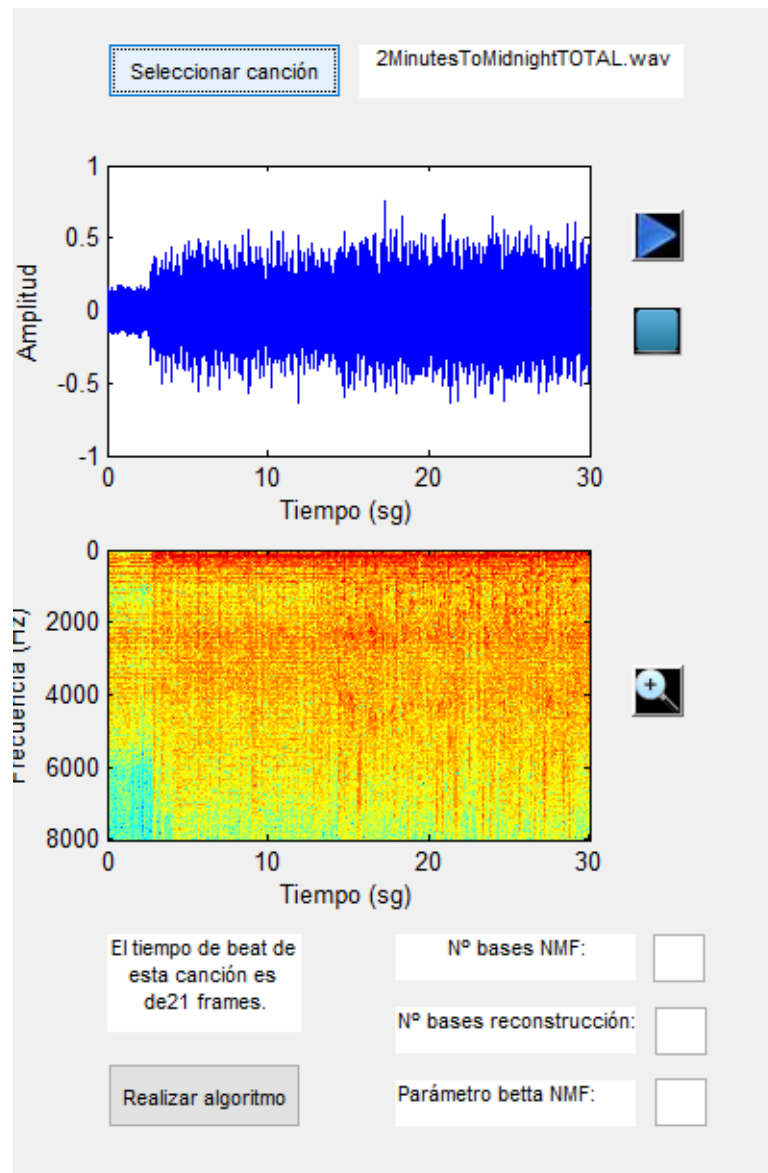


**Fig. 8.2 Selección de la canción.**

Una vez seleccionada la canción deseada aparecerán de forma automática la representación temporal de esta y su espectrograma. También se realizará el algoritmo REPET y se nos facilitará el parámetro  $P$  calculado por el mismo.

El siguiente paso a realizar consiste en la introducción de los parámetros con los que se desee llevar a cabo el algoritmo NMF. Una vez introducidos haremos clic sobre el botón “Realizar algoritmo”.

El tiempo de beat de esta canción es de 21 frames.

Nº bases NMF: 21

Nº bases reconstrucción: 10

Parámetro betta NMF: 1

Realizar algoritmo

Fig. 8.3 Introducción de los parámetros del NMF.

Una vez introducidos estos, en la parte derecha del interfaz aparecerá algo como la siguiente imagen:

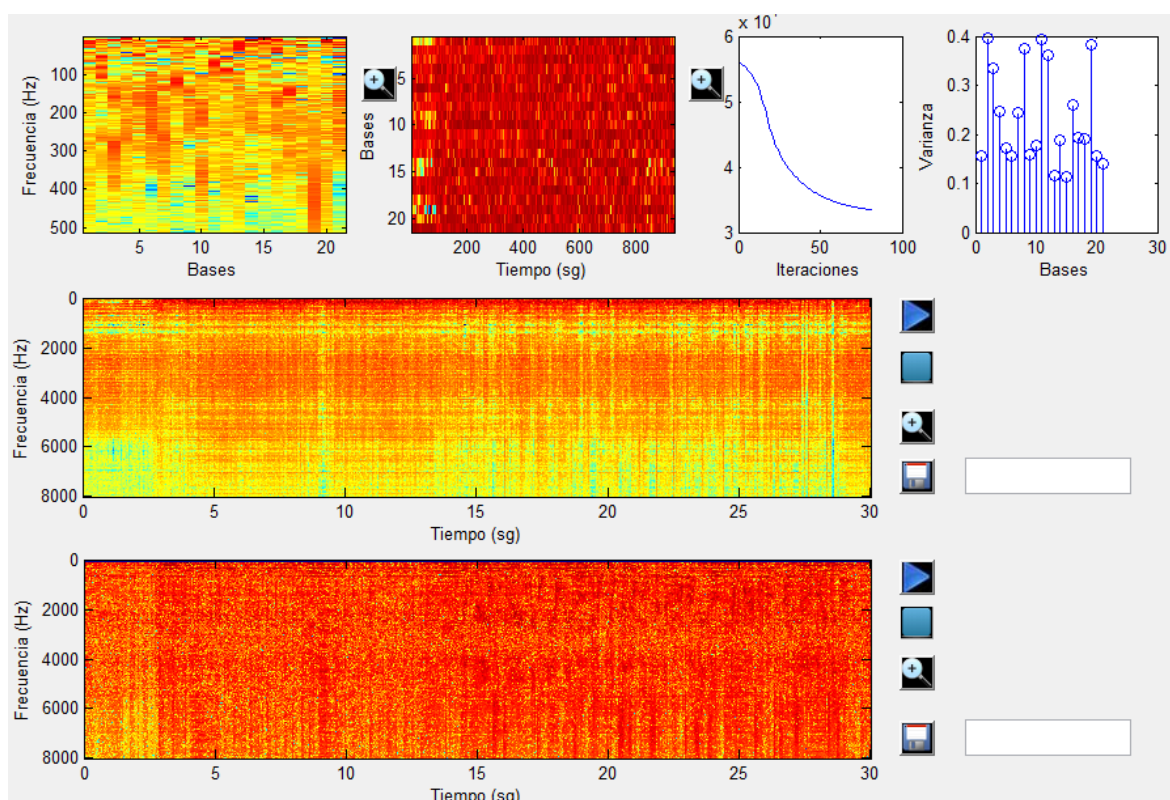


Fig. 8.4 Resultados obtenidos tras la realización del algoritmo.

Una vez realizado el algoritmo podremos comprobar los resultados obtenidos mediante las imágenes que aparecen así como comprobar la varianza temporal de cada una de las bases o la divergencia del NMF.

Tras escuchar los resultados obtenidos y comprobar su validez se da la opción de guardar estos en ficheros de audio. Para ello se deberá escribir el nombre del fichero que se quiere crear y seleccionar el botón de guardado.

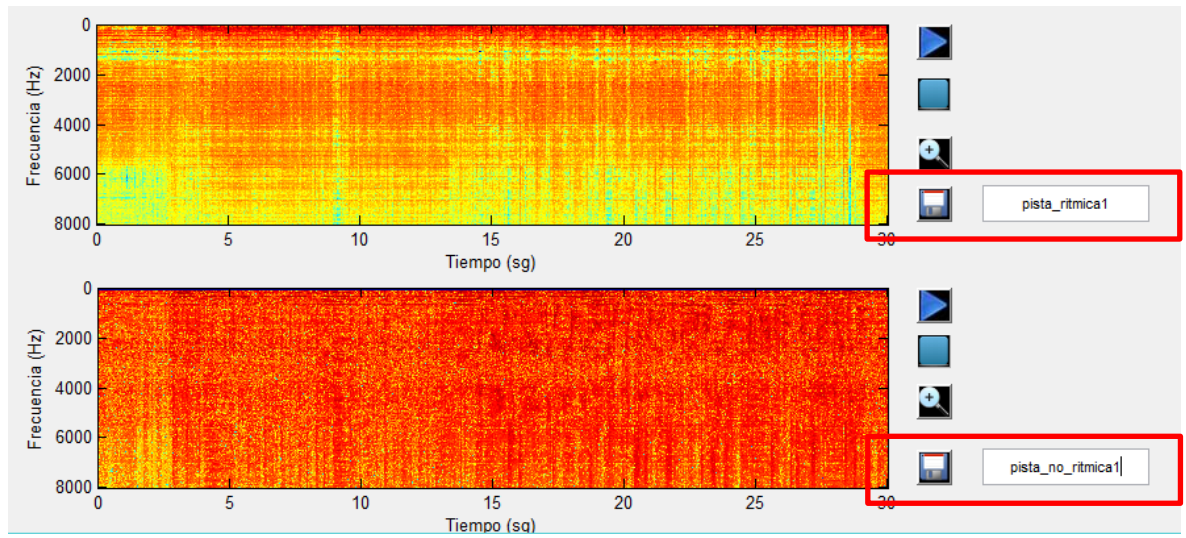


**Fig. 8.5** Proceso de guardado de las señales generadas.