

# 1. Espectrograma

## 1.1. Función para realización de espectrograma

Para la realización de los espectrogramas, se utilizó la función “spectrogram”, de la librería «scipy» de Python. La misma se basa en la implementación de transformadas de Fourier consecutivas sobre la señal a lo largo del tiempo. La sintaxis para su utilización es la siguiente:

```
scipy.signal.spectrogram(x, fs=1.0, window=('tukey', 0.25), nperseg=None, noverlap=None, nfft=None,
                        detrend='constant', return_onesided=True, scaling='density', axis=-1, mode='psd')
```

Donde los parámetros y sus efectos en detalle se describen a continuación.

- **x [array]:** arreglo con los valores que toma la señal en el tiempo (sobre los que se aplica la transformada de Fourier).
- **fs [float]:** frecuencia de sampleo de la señal  $x(t)$ . Por defecto normalizada: 1.0.
- **window [string ó tuple ó array]:** ventana temporal a utilizar. Influye tanto el tipo de ventana como como el ancho de la misma.

Respecto al tipo de ventana, si se utiliza por ejemplo una ventana cuadrada, esta posee un corte abrupto. Si en dicha ventana no entra un número entero de períodos de la señal, se produce un corte abrupto.

Esto remite en la aparición de otras componentes de alta frecuencia que antes no había, lo que se conoce como «fuga espectral» (dado que la energía de los armónicos principales se “fuga” a los otros armónicos nuevos). Otras ventanas (como la Blackman-Harris) tienen un corte suave en los extremos (tienden a cero gradualmente), lo que minimiza la fuga espectral considerablemente.

El ancho de la ventana interviene en la resolución en tiempo y en frecuencia. Si la ventana es más ancha, se obtiene mayor resolución en frecuencia, dado que si la frecuencia de la señal sufre algún cambio en el tiempo (como en una señal FM), es posible captarlo con la ventana con mayor definición. Pero en el tiempo se pierde resolución dado que se sabe con menor precisión dónde ocurre exactamente el cambio de frecuencia. Con una ventana angosta, se gana resolución en el tiempo, pero en frecuencia se podría perder el cambio que antes se lograba captar en el tiempo con una ventana ancha, por lo que se vería una sola frecuencia en lugar de dos.

- **npersec [int]:** es el largo de cada segmento. Por defecto es «None», pero si la ventana se da en formato de «string» se considera 256, y si se da como «array» es el largo del mismo.
- **noverlap [int]:** es el número de puntos a solapar entre segmentos. Por defecto es «None», que es  $npersec // 8$ . Es decir, define la separación resultante entre ventanas.
- **nfft [int]:** es el largo de la FFT utilizada. Por defecto es «None», que determina el largo igual a “npersec”.
- **detrend [string ó function ó False]:** elimina la tendencia lineal a lo largo del eje temporal. Por defecto es “constant”. Si es de tipo string, se pasa directamente a la función «detrend» (también de scipy), y si es una función, se le pasa un segmento y lo devuelve ya con el desvío correspondiente. Si es «False», no se aplica ningún desvío. Para el caso de audio, no nos es de interés.
- **return\_onesided [bool]:** si se asigna «True», se devuelve un espectro unilateral. Si es «False», el espectro será bilateral.
- **scaling ['density', 'spectrum']:** considerando a  $x(t)$  en volts [V], se elije si procesar la densidad espectral de potencia [ $V^2/Hz$ ] ó el espectro de potencia [ $V^2$ ]. Por defecto se procesa la densidad espectral. Se simboliza como  $S_{xx}$ .
- **axes [int]:** es el eje a lo largo del cual se procesa el espectrograma. Por defecto es «-1» (es decir, el último utilizado).
- **mode [str]:** define que es lo que se espera que devuelva la función, entre [“psd”, “complex”, “magnitude”, “angle”, “phase”]. “psd” es la densidad espectral de potencia; con “complex” devuelve la STFT (Short-Time Fourier Transform) compleja; “magnitude” devuelve el valor absoluto de la STFT, y “angle” y “phase” el ángulo correspondiente complejo.

Los parámetros que devuelve son los siguientes:

- **f** [**ndarray**]: arreglo de dimensión “n” con las frecuencias de sampleo.
- **t** [**ndarray**]: arreglo de dimensión “n” con los segmentos de tiempo.
- $S_{xx}$  [**ndarray**]: arreglo de dimensión “n” con el espectrograma de  $x(t)$ .

En referencia a las diferentes ventanas utilizables, además del ancho se debe considerar realizar el análisis con o sin overlap (solapamiento entre segmentos). En el caso más sencillo, no utilizar overlap podría conllevar pérdida de información: si se tiene un cambio de frecuencia de corta duración en comparación a la duración total del segmento, y se encuentra cerca del final de éste, al no utilizar segmentos solapados para la STFT podría perderse dicho evento. Para evitar dicha pérdida de información, se aplica la STFT a segmentos con un cierto grado de solapamiento. Se gana en no perder información, pero se requiere mayor tiempo de procesamiento, dado que al ser segmentos solapados, para cubrir la totalidad de la señal se necesita realizar más veces la transformación. El porcentaje de overlap se encuentra en general entre el 50 % y el 75 %, de acuerdo al paper «*On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform*» (1978, Fredric J. Harris, member IEEE).

## 1.2. Aplicación a escala de Sol Mayor (G3)

La tonalidad de *sol* mayor (en el sistema inglés abreviada como **G**) consiste en la escala mayor del sol y contiene las notas *sol*, *la*, *si*, *do*, *re*, *mi*, *fa* sostenido y *sol*. La notación **G3** corresponde a la pequeña octava, cuya frecuencia es aproximadamente 195.998Hz.

El espectrograma realizado se muestra a continuación.

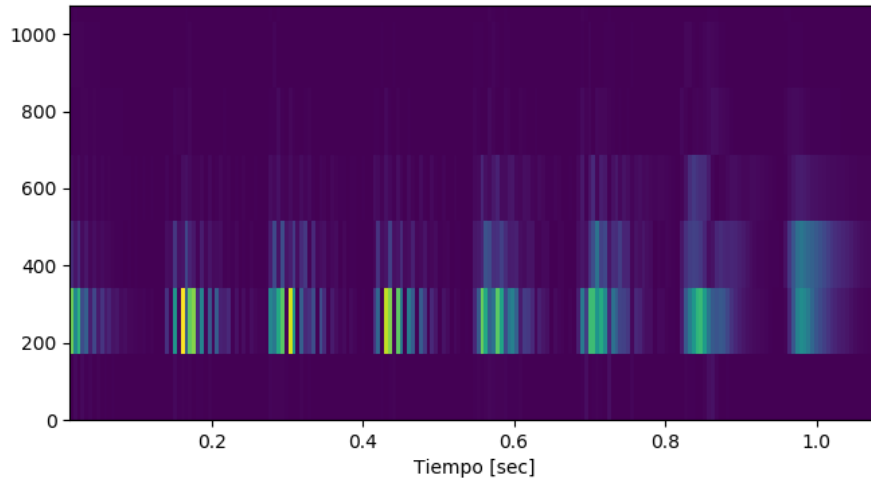


Figura 1: Espectrograma de Sol mayor

En el espectrograma obtenido sintetizando con un piano, resulta visible la frecuencia de la primera octava cercana a los 200Hz. Se utilizan 120ms por nota según lo sugerido, de manera tal que puede diferenciarse cada una por la escala creciente en frecuencia.