

1. 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. Infiuye 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 [str ó function ó False]**: NO SE BIEN QUE HACEE
- **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]**: VER BIEN QUE SERIA DICE QUE ES EL EJE SOBRE EL QUE HACE EL ESPECTRO O ALGO POR EL ESTILO
- **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 $x(t)$.