

Analizador espectral de 5 bandas implementado en Python/Scipy - Nicolás Padula

Analizador espectral de 5 bandas implementado en Python/Scipy

Padula, Nicolás - nicolasvpadula@gmail.com

UTN-FRSF

Matemática Superior - Trabajo Práctico N°1

Introducción

Análisis espectral

El análisis de señales o datos en términos de magnitudes referentes al espectro de la misma (frecuencia, energía, etc.) pueden aportar información clave para el desarrollo de las actividades que comprenden las distintas áreas de la ciencia y la tecnología. Esta perspectiva de análisis se conoce como análisis espectral.

En particular, un *analizador espectral* es un dispositivo que efectúa la medición de la magnitud de una señal de entrada a lo largo del eje de frecuencias del dispositivo en cuestión. Específicamente, el requerimiento del trabajo es el de desarrollar un analizador espectral en 5 bandas de frecuencias, entre 200 y 2000hz, de donde se deduce que la magnitud de las mediciones obtenidas en cada una de estas bandas representa la energía total de una señal de entrada entre los extremos de dicha banda.

Puede definirse a la energía de una señal $f(t)$ como:

$$\int_{-\infty}^{\infty} |f(t)|^2 dt \quad (1)$$

Existen diversas alternativas para la construcción de un analizador espectral. La implementación física del mismo suele realizarse a través de *filtros pasabanda* cuya respuesta de frecuencia permite el paso de bandas similares a las bandas en las cuales se quiere medir la energía.

Planteando el problema desde una perspectiva puramente matemática, es sabido que la representación en el espacio frecuencial de una señal en el espacio temporal $f(t)$ es $F(iw)$ (Oppenheim et al. 1997), donde:

$$F(iw) = \int_{-\infty}^{\infty} f(t) e^{-iwt} dt \quad (2)$$

Es decir, $F(iw)$ es la transformada continua de Fourier de $f(t)$. Haciendo uso de la relación de Parseval (Oppenheim et al. 1997) , sabemos que:

$$\int_{-\infty}^{\infty} |f(t)|^2 dt = \int_{-\infty}^{\infty} |F(iw)|^2 dw \quad (3)$$

Lo cual significa que la energía de la señal, es igual a la energía de la representación de la señal en el espacio frecuencial.

Haciendo uso de esta propiedad y volviendo a la definición de nuestro problema, si nuestro objetivo es encontrar la energía E entre dos frecuencias w_0, w_1 , entonces:

$$E = \int_{w_0}^{w_1} |F(iw)|^2 dw \quad (4)$$

Valiéndonos de este razonamiento, y considerando que la implementación del trabajo se hará a través de software, se decidió resolver el trabajo bajo este enfoque más abstracto, evitando las particularidades que surgen en el diseño de un sistema lineal basado en filtros, como el diseño de los mismos, el cálculo de los distintos parametros, la obtención de las frecuencias de corte, etc.

En función de todo lo expresado anteriormente, el presente trabajo propone la implementación de un analizador espectral de 5 bandas entre 200 y 2000hz desarrollado en Python, haciendo uso del framework de computación científica Scipy.

Desarrollo del trabajo

Considerando lo expuesto en la sección anterior, se establecieron tres tareas a realizar para el desarrollo del trabajo:

- 1 - Transformar la señal de entrada
- 2- Calcular la energía en las 5 bandas dentro del rango de frecuencias 200-2000hz
- 3- Graficar los resultados obtenidos.

Se exploraron diversas alternativas para llevar a cabo el trabajo (Scilab, Xcos, Matlab, SymPy, Scipy) pero debido al conocimiento previo del lenguaje Python del autor, se decidió hacer uso de herramientas basadas en Python para la implementación de las mismas. Inicialmente se decidió llevar a cabo estas tareas de manera simbólica a través de SymPy (SymPy Development Team, 2016) , pero debido a ciertas particularidades en el módulo de transformadas integrales, el cálculo de las mismas resulta erróneo en algunos casos¹. Sabiendo esto, se decidió implementar el trabajo de manera numérica, utilizando Scipy (Jones et al., 2001).

Se creó un arreglo t , que representará los puntos muestreados de tiempo, usando la función *linspace* perteneciente al módulo Numpy (Van der Walt et al., 2011). El número de muestras elegido fue de 4000, para poder representar los 2000hz, y el espaciado entre las muestras se estableció en $1/4000$.

Se utilizó el módulo *scipy.signal* para generar diversas señales de prueba a partir de las muestras de tiempo definidas.

1

<http://stackoverflow.com/questions/21202244/fourier-transform-a-trig-function-in-sympy-returns-unexpected-result>
<https://github.com/sympy/sympy/issues/2803>

La transformación de las señales se realizó mediante el módulo *scipy.fftpack*, el cual implementa el algoritmo Fast Fourier Transform (FFT) para transformar las señales. Las frecuencias muestreadas para estas señales transformadas se obtuvieron mediante la función *fftfreq*. Cabe destacar que dado que la frecuencia del analizador es medida en hz, la transformada de Fourier $F(f)$ se define como²:

$$F(f) = \int_{-\infty}^{\infty} f(t) e^{-i2\pi ft} dt \quad (5)$$

Finalmente, se partitionaron los arreglos de frecuencias y los puntos de $F(iw)$ asociados a dichas frecuencias en 5 bandas divididas por puntos de cortes igualmente espaciados entre 200 y 2000hz. Se calculó la energía de cada banda como se especificó en la ecuación 4. Finalmente, esta energía se representó en decibeles, sumándole una constante que funcione como piso o potencia mínima medible para el dispositivo. Por último, se graficó estas magnitudes obtenidas con respecto a las frecuencias utilizando *matplotlib* (Hunter, 2007).

El código completo del analizador puede verse en el archivo adjunto *TP1.py*.

Se realizaron pruebas con señales conocidas, definidas anteriormente a través de *scipy.signal*. Los resultados obtenidos se presentan a continuación.

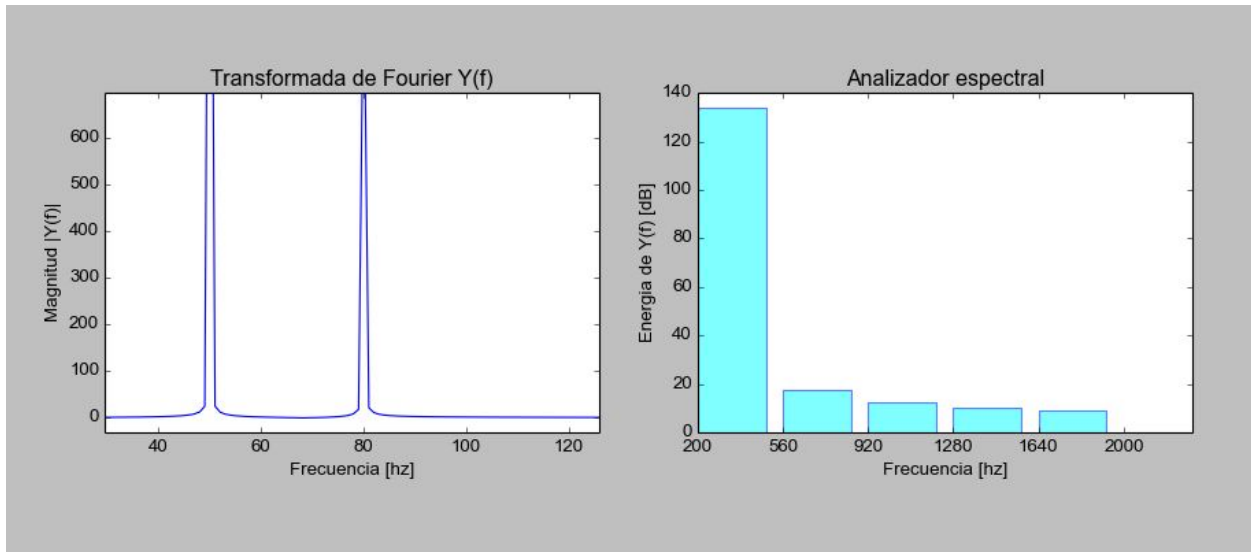
Resultados

A continuación se presentan algunas señales probadas, en los gráficos se muestran las transformadas de cada señal, y el gráfico del analizador espectral.

² <https://www.physicsforums.com/threads/need-help-understanding-fourier-transform-in-hz-vs-radians.377327/>

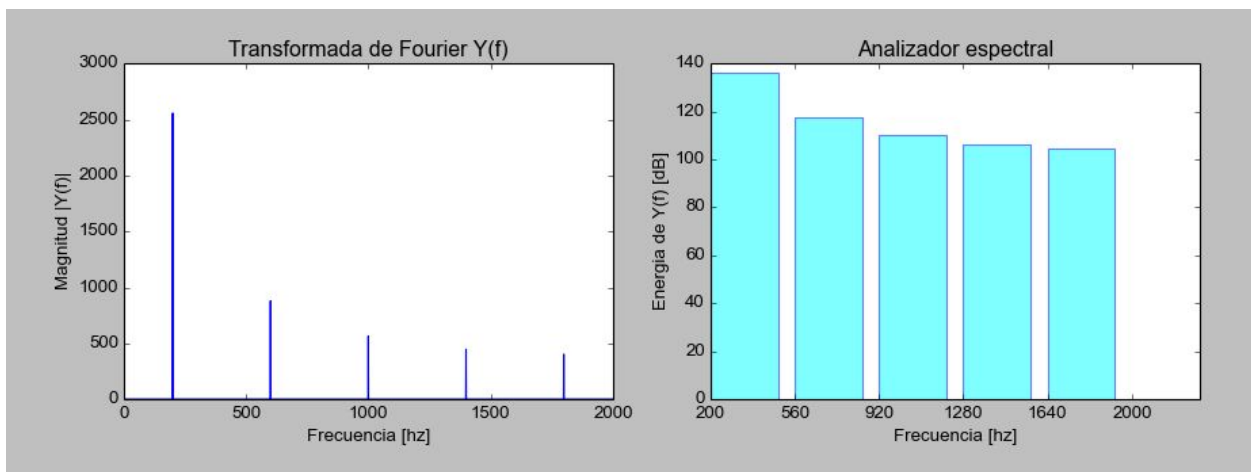
Señal 1

$y = \text{numpy.sin}(50.0 * 2.0 * \text{numpy.pi} * t) + 0.5 * \text{numpy.sin}(80.0 * 2.0 * \text{numpy.pi} * t)$



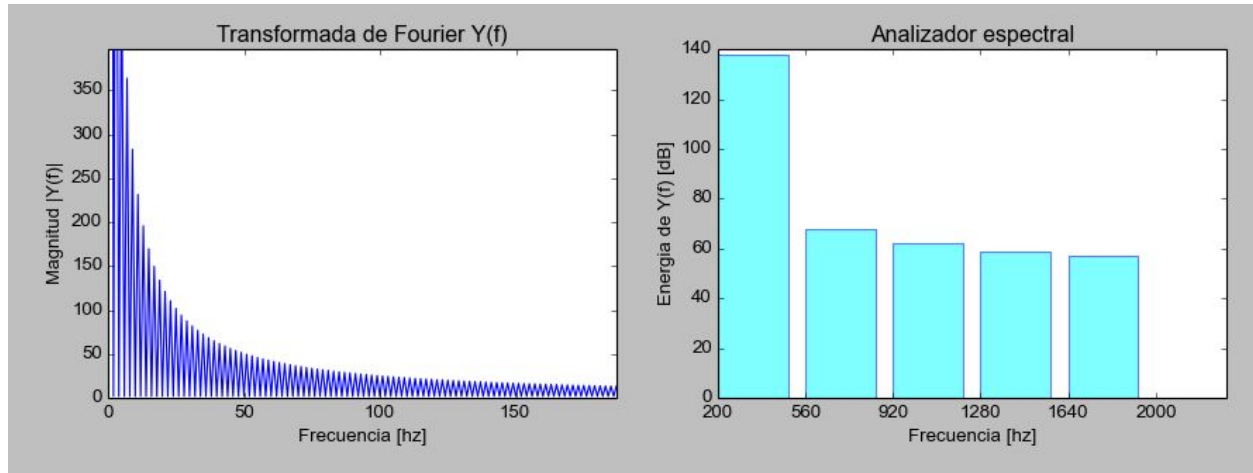
Señal 2

$y = \text{signal.square}(200 * 2.0 * \text{pi} * t)$



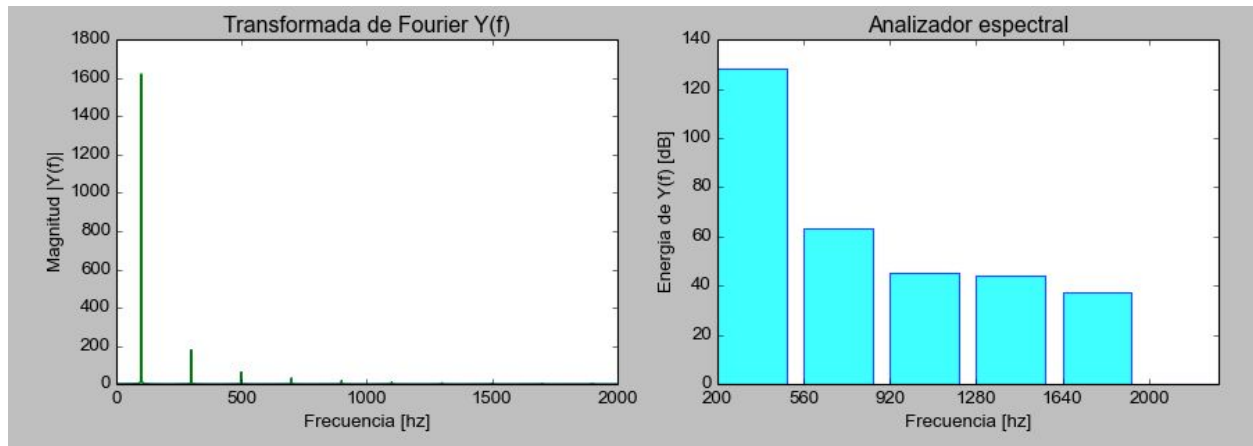
Señal 3

`y = signal.square(2.0*pi*t)`



Señal 4

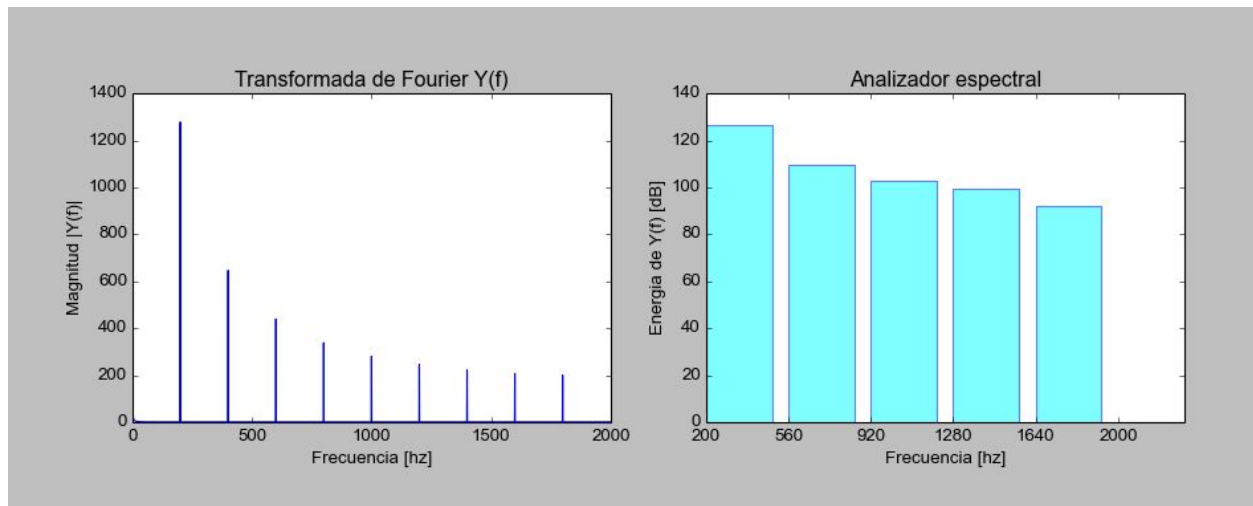
`y = scipy.signal.sawtooth(100 * 2.0*pi*t, 0.5)`



Señal 5

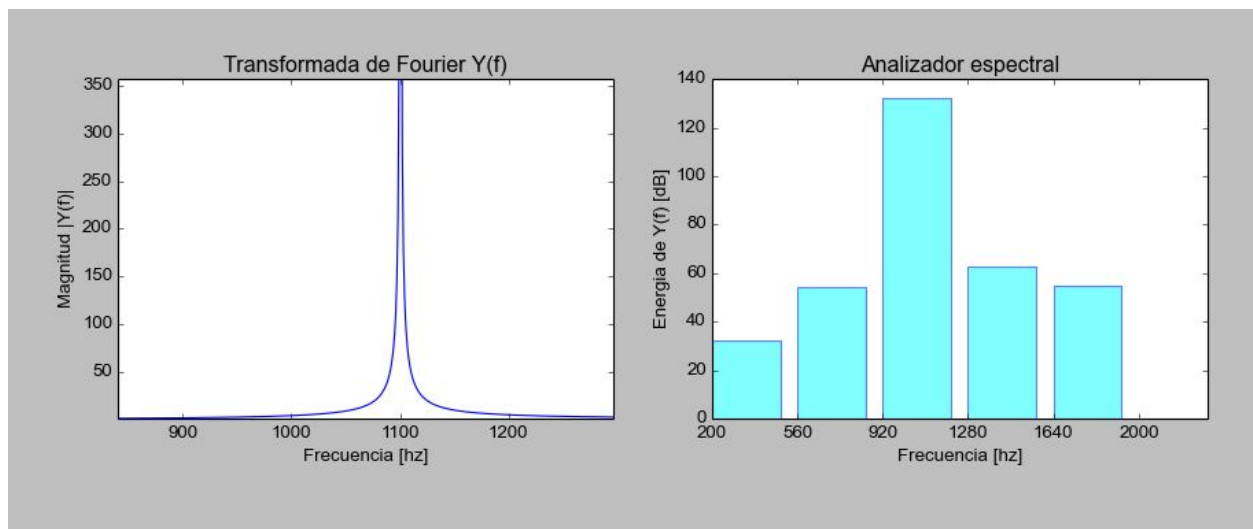
`y = scipy.signal.sawtooth(200 * 2.0*pi*t)`

Analizador espectral de 5 bandas implementado en Python/Scipy - Nicolás Padula



Señal 6

$$y = \text{scipy.sin}(1100 * 2 * \pi * t)$$



Discusión y conclusiones

Los resultados confirman que el analizador espectral representa la energía de la señal con cierto grado de precisión. Mayor fidelidad puede lograrse seleccionando un muestreo y frecuencia adecuadas para una buena visualización de las señales a estudiar. Esto nos lleva a

Analizador espectral de 5 bandas implementado en Python/Scipy - Nicolás Padula

concluir que un buen conocimiento de muestreo y del manejo de las señales discretas es crucial para obtener resultados precisos si se hace uso de un enfoque numérico.

Referencias

John D. Hunter. *Matplotlib: A 2D Graphics Environment, Computing in Science & Engineering*, 9, 90-95 (2007), DOI:10.1109/MCSE.2007.55

Jones E, Oliphant E, Peterson P, et al. *SciPy: Open Source Scientific Tools for Python*, 2001-, <http://www.scipy.org>.

Oppenheim, A. V., Willsky, A. S., & Nawab, S. H. (1997). *Signals and systems*. Prentice Hall.

Stéfan van der Walt, S. Chris Colbert and Gaël Varoquaux. *The NumPy Array: A Structure for Efficient Numerical Computation, Computing in Science & Engineering*, 13, 22-30 (2011), DOI:10.1109/MCSE.2011.37

SymPy Development Team (2016). *SymPy: Python library for symbolic mathematics*. URL <http://www.sympy.org>.