



Área Académica de Ingeniería Física

Física Computacional I

Tarea 3

Estudiante:
Ever Ortega Calderón

Fecha de entrega:
6 de abril de 2021

Profesores:
Álvaro Amador Jara
Jose Esteban Pérez Hidalgo

I Semestre 2021

Pseudocódigos

1. Para generar la señal ruidosa se generará dos señales con diferentes frecuencias, se sumarán y se agregará ruido por medio de una distribución normal aleatoria de numpy. Por lo que se debe definir las variables de magnitud y frecuencia para cada señal.

Las señales se generarán por medio de la función `signal` de SciPy, una de ellas será una cuadrada (`square`) y la otra será dientes de sierra (`sawtooth`) (las funciones deberán recibir el conjunto de puntos en el tiempo para lo cual se definirá una tasa de muestreo y un `deltaT` que será la separación deseada en el tiempo, su multiplicación será la cantidad de puntos a analizar y por medio de la función de Numpy `linspace` se genera el arreglo de puntos en el tiempo) y su suma la contendrá una nueva variable a la cual se le sumará el ruido el cual se creará por medio de `numpy.random.normal` y esto se guardará en otra variable la cual será la señal ruidosa para trabajar.

Por último, se graficarán estas señales, tanto la original como la de ruido para evidenciar que efectivamente se está distorsionando la señal.

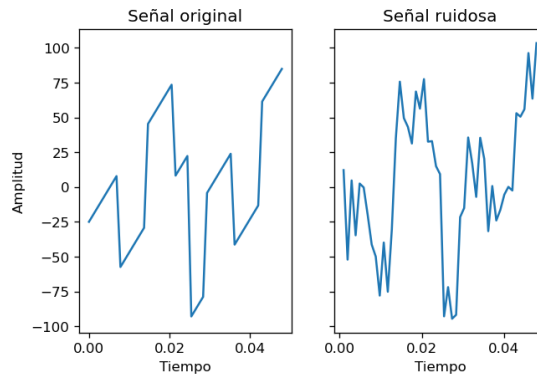
2. Para la transformada de Fourier se usa la implementación de `Scipy.fft.fft`, la cual se le aplicará a la señal ruidosa. Para la gráfica en el dominio de la frecuencia se calculan las amplitudes y los puntos que corresponderán a las frecuencias, esto de nuevo con `linspace`
3. Para filtrar la señal se crea la función `Filtrar_Señal()`, en esta función se introducirá un parámetro umbral, el cual definirá cuales frecuencias se filtran por medio de un ciclo en el cual las amplitudes que no sean mayores al umbral serán ruido y se eliminarán dándoles un valor de 0 y se graficará las amplitudes lo que mostrará la transformada en el dominio de la frecuencia, pero con el ruido, menor a umbral, eliminado.
4. Para la transformada inversa se usa la implementación `Scipy.fft.ifft` para lo cual se calcula la transformada rápida inversa de la señal ruidosa, pero filtrada, con el filtro que se hizo en el punto 3, esto dará la señal ruidosa filtrada, pero en el dominio del tiempo, con lo que se grafica los puntos en el tiempo y la transformada inversa.

Código debidamente documentado:

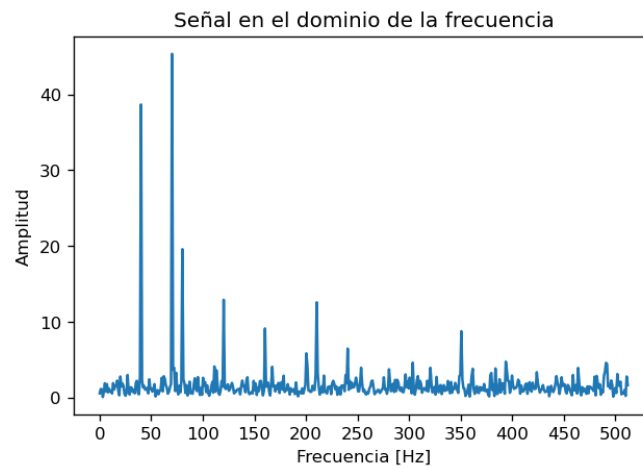
<https://github.com/ever2706/FCI-Tarea3.git>

Presentación y análisis de resultados

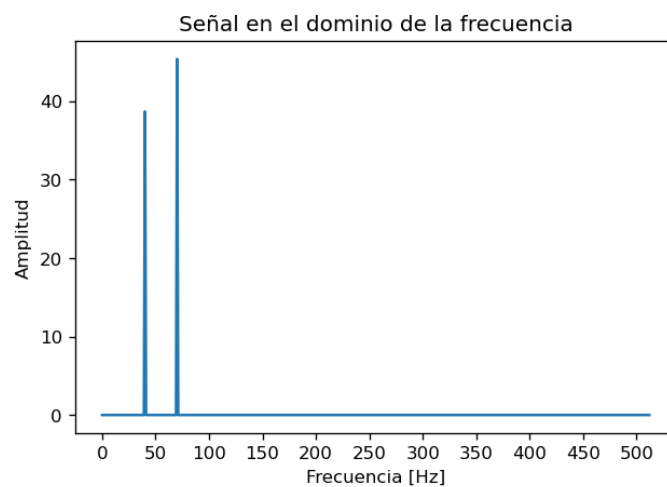
- Para cada gráfica solicitada (4 gráficas) comente brevemente lo que muestra la gráfica



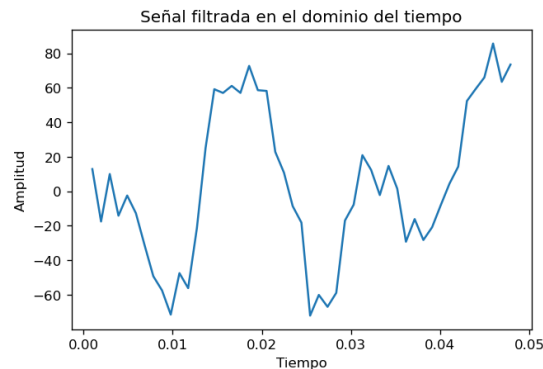
Se observa a la izquierda la señal pura, la cual se creó al sumar una señal cuadrada y una de dientes de sierra, a la derecha se ve la señal de la izquierda más un ruido aleatorio, por ello es importante señalar que esta señal ruidosa cambiará cada vez que el código se ejecute, pues el ruido es totalmente aleatorio. Ambas se ven en el dominio del tiempo.



En esta se ve la transformada de Fourier de la señal ruidosa, por lo que se ejemplifica la señal en el dominio de la frecuencia, importante señalar que las dos frecuencias más elevadas en amplitud hacen referencia a las dos frecuencias principales de las señales puras creadas y definidas al inicio.



En esta gráfica se observa el gráfico anterior menos las frecuencias consideradas como ruido, estas frecuencias se eliminaron al elegir un valor umbral que sería la referencia para considerar qué señales eran ruido, destacan las frecuencias principales de la señal pura.



En esta gráfica se aplicó la transformada rápida inversa, con lo que se tiene la función filtrada en el dominio del tiempo, si se compara con la gráfica inicial de la señal pura, se puede observar que no quedan líneas tan rectas como en la original, se obtienen picos que en la original no se dan, pero a grandes rasgos posee la misma forma que la señal original.

- ¿Cómo se obtienen las amplitudes para cada frecuencia a partir de los valores de la transformada de Fourier?

Las amplitudes o el espectro de amplitudes se puede obtener una vez que se tiene la transformada rápida de la señal, recordando que los valores de la transformada son números complejos, los cuales para nuestro caso están representados de forma cartesiana $a+bj$. Para el espectro de amplitudes se calcula para un lado de la FFT, por eso se hace para la mitad de los puntos desde cero hasta $nPuntos/2$, se calcula la magnitud de estos números complejos (importante señalar que se descarta la parte imaginaria por ende solamente se calcula el valor absoluto de la transformada, pues lo correspondiente a la parte imaginaria sería sumar un cero) y se divide entre la cantidad de puntos $nPuntos$, al final de se multiplica por 2 para tener la totalidad de puntos y así se obtiene el espectro de amplitudes.

- ¿Qué relación hay entre el ruido insertado a la señal y el umbral de filtrado?

El valor de umbral marca la referencia sobre cuales frecuencias de ruido se eliminaran de la resultante transformada de Fourier, si el valor de umbral cubre de la mayor frecuencia de ruido hacia abajo entonces todo el ruido se eliminará, en otro caso si el valor de umbral es muy alto, inclusive mayor que las frecuencias que no son ruido, entonces se filtrará todo y el resultado será una amplitud de cero para todas las frecuencias, por otro lado, si el umbral es muy bajo entonces no se filtraran todas las frecuencias de ruido. Por lo que lo ideal será que el umbral sea la mayor frecuencia de ruido y con esto se garantizará que se elimine todo el ruido y que así la gráfica sin ruido sea lo más parecida al resultado esperado. En el caso estudiado se usó un umbral de 21 el cual eliminó todas las frecuencias con una amplitud menor a 21 considerando las cómo ruido y se conservaron las frecuencias restantes.