TRABAJO PRÁCTICO 2

FUNDAMENTOS MATEMÁTICOS DEL PROCESAMIENTO DE SEÑALES

23 Diciembre 2021

Este trabajo práctico se realizará en Python 3 y se aconseja fuertemente utilizar el entorno de programación Anaconda e implementar las diferentes secciones en Jupyter Notebooks, con comentario y notas en cada ítem.

Se espera que el código entregado pueda ser descargado desde github y ejecutado (figuras reproducibles) y que cada resultado este acompañado de una breve discusión en formato *markdown*, el cual permite escribir ecuaciones dentro de Jupyter.

Utilizaremos los siguientes métodos y librerias:

```
import numpy as np
import pylab as pl
import scipy as sp
import scipy.signal
import scipy.io.wavfile
```

1 Convolución discreta

Una señal finita de N muestras se escribe como un vector

$$x = \begin{bmatrix} x[0] \\ x[1] \\ \vdots \\ x[N-1] \end{bmatrix} \in \mathbb{C}^N$$

Usando la convolución circular, un operador lineal y homogéneo en el tiempo se puede expresar como:

$$y = x \circledast h = C_h^o x \,,$$

donde $C_h^o \in \mathcal{M}_{\mathbb{C}}(N, N)$ es una matriz de convolución parametrizada por el vector h, la respuesta impulsional del filtro.

Debido a los efectos de los bordes, la convolución circular $\circledast$ es poco utilizada en algoritmos de la vida real. Se ocupa en su reemplazo la convolución discreta donde las dos señales son "rellenadas" (*padding* en inglés) con zeros fuera del intervalo de muestreo:

$$\tilde{x}[n] = \begin{cases} x[n] & \text{si } 0 \leq n < N \\ 0 & \text{si no} \end{cases} \,, \qquad \tilde{x} \in \mathbb{C}^{2N-1}$$

$$\tilde{h}[n] = \begin{cases} h[n] & \text{si } 0 \leq n < N \\ 0 & \text{si no} \end{cases} \,, \qquad \tilde{h} \in \mathbb{C}^{2N-1} \,.$$

Escribimos luego

$$y = x \star h = C_h x$$

donde $C_h \in \mathcal{M}_{\mathbb{C}}(2N-1, N)$.

1. Dada una respuesta impulsional h, escriba las matrices C_h y C_h^o. Pruebe que C_h es una matriz de Toeplitz y C_h^o es una matriz circulante.
2. Ahora estudiaremos los efectos de borde de la convolución circular.

2 Transformadas de Fourier Discreta (DFT) y Rápida (FFT)

La DFT es un operador lineal que se puede representar como una matriz $F_N = [e_0, e_1, ..., e_{N-1}]^T \in \mathcal{M}_{\mathbb{C}}(N, N)$ donde $e_k \in \mathbb{C}^N$ es la k-ésima función de la base de Fourier:

$$(F_N)_{k,p} = e_k[p] = e^{i\frac{2\pi(p-1)}{N}(k-1)} .$$

2.1 Funciones básicas y transformada discreta de Fourier (DFT)

1. Calcule las funciones de base de Fourier e_k para todos los índices k para una señal de tamaño $N = 32$. Trazar sus partes reales e imaginarias en una figura.
2. Calcule F_N la matriz de transformada discreta de Fourier (DFT) para $N = 1, 2, 4$ y 32. Para $N = 32$ visualícela las partes reales e imaginarias como una imagen. ¿Qué contienen las líneas de la matriz F_N?
3. Para una señal de largo $N = 1024$ tal que

$$x[n] = \cos(2\pi f_0 n)$$

   con $f_0 = \frac{k}{N}$ para $k < \frac{N}{2}$ un entero positivo, calcule la matriz F_N y aplíquela a la señal x. Visualice luego la magnitud del resultado en el dominio frecuencial.
4. Cambie f_0 a un valor grande tal que $k > N$. ¿Qué sucede con el espectro?

2.2 Transformada de Fourier Rápida (FFT)

1. Mida el tiempo de cálculo de la DFT con y sin el cálculo previo de la matriz de Fourier F_N para $N = 1024$
2. Usando el método `np.fft.fft` mida el tiempo del cálculo de la FFT para la misma señal y compárelo con los dos tiempos anteriores.
3. Calcule y almacene el tiempo computacional de la DFT, la DFT con la matriz precalculada y la FFT para $N = 2k$ muestreados logarítmicamente de $k = 2$ a $k = 12$. Grafique en gráfico log vs log los diferentes tiempos computacionales en función de N. Discuta lo observado.

3 Interpretación de las señales

Para todas las señales que se describen a continuación, siga los siguientes pasos:

1. Cargue la señal en la memoria y almacene tanto la señal x como la frecuencia de muestreo f_s.
2. Trace la señal en el tiempo con el eje x adecuado (tiempo en segundos).
3. Grafique la magnitud de la señal FFT con la frecuencia real correspondiente centrada alrededor de 0.
4. Interprete y discuta las propiedades de la señal en frecuencia utilizando la información proporcionada sobre cada señal en las descripciones a continuación. Cuando sea posible, recupere parámetros físicos como la constante de tiempo de las frecuencias fundamentales de las señales cuando sea posible. Puede hacer zoom en parte de un gráfico utilizando `pl.xlim ([xmin, xmax])`.

Lista de señales y su nombre de archivo correspondiente. Algunas se han guardado en los trabajos prácticos anteriores y las demás se pueden descargar desde el repositorio github del curso.

- A4.wav contiene la nota MIDI m = 69.
- A4clip.wav contiene la nota MIDI m = 69 con el recorte correspondiente a un efecto de saturación.
- seq.wav contiene la secuencia de notas generada en el TP N° 2.
- chirp.wav contiene una señal de modulación de frecuencia de chirp. ¿Cuáles son las frecuencias instantáneas en esta señal? ¿Cuál es el soporte de su espectro?
- uku.wav y uku2.wav contienen una nota "mal" interpretada en un ukelele. ¿ Cuales son las notas tocadas? ¿Cuáles son sus números MIDI correspondientes?

- drum.wav es una grabación de una ejecución de batería que contiene un bombo y un platillo, correspondientes respectivamente a una señal de baja frecuencia y alta frecuencia.
- stairway.wav y stairwayb.wav contiene 10 segundos del comienzo de una canción conocida. El segundo archivo ha sido dañado por ruido. Haga un zoom en las frecuencias bajas y encuentre el modo. ¿Cuál es la nota midi correspondiente a este modo que es la nota más tocada en la secuencia? ¿Cuál es el soporte en frecuencia del ruido agregado?
- ecg.npz es la grabación de un Electrocardiograma. ¿Puede ver el promedio de latidos por minuto en el espectro? ¿Dónde está situado el ruido en el dominio de la frecuencia?
- conso.npz es el registro del uso eléctrico en Watts del edificio del Centro de Novación Drahi-X con un período de muestreo de 1 min durante aproximadamente 4 semanas. Recupere los días y las semanas a partir de la señal. En el dominio de la frecuencia, haga zoom en las frecuencias bajas, en particular en aquellas correspondientes con periodicidad de 1 día y 1 semana.

4 Filtraje Digital

En esta sección estudiaremos varios filtros digitales y los aplicaremos a las señales.

4.1 Filtraje Ideal

1. Cargue la señal en el archivo “stairwayb.wav”. Intentaremos atenuar el ruido presente en el audio cortando toda la banda de frecuencia donde hay ruido.
2. Calcule la FFT de la señal y grafique su magnitud en el dominio de Fourier. Seleccione una frecuencia de corte f_c para un filtro de paso bajo ideal.
3. Aplique un filtro ideal “low-pass” con una frecuencia de corte f_c. Escuche la señal filtrada. Tenga en cuenta que guardar un archivo wav en formato flotante recorta los valores entre -1 y 1, por lo que la señal se debe escalar correctamente para evitar la saturación.
4. Utilice un filtro ideal para seleccionar solo la nota con la frecuencia más baja en la señal “seq.wav”. Escuche la señal filtrada para comprobar que solo queda una nota.
5. Calcule la transformada de Fourier inversa de la respuesta de frecuencia ideal del filtro de paso bajo para obtener su respuesta de impulso de convolución circular. Haga lo mismo con el filtro de paso alto ideal. Compruebe ambos que la frecuencia 0 se pasa y se corta respectivamente calculando la ganancia estática.

4.2 Diseño de filtros digitales

En aplicaciones de la “vida real”, a menudo es necesario diseñar filtros causales. Esto se hace estimando coeficientes de un filtro FIR o IIR, de orden finito, que aproxima sistemas de tiempo continuo como los filtros de Butterworth de Chebychev.

1. Calcule los coeficientes de un filtro Butterworth FIR discreto para una frecuencia de corte normalizada de $f_c = 0.2$ (para una frecuencia de muestreo de 1) de orden $n = 2$.
2. Implemente una función `def freqResp (a, b, f)` que devuelve la respuesta de frecuencia a un filtro IIR a, b para obtener una lista de frecuencias f. Grafique la respuesta de frecuencia para el filtro Butterworth de órdenes $n = 1, 2, 3, 4$.
3. Aplique el filtro a la señal “stairwayb.wav”. ¿Cual es el equivalente frecuencia de corte en Hz? ¿En qué orden el filtro es lo suficientemente fuerte como para atenuar bien el ruido?
4. Calcule los coeficientes de un filtro Chebychev FIR discreto de tipo 1 para una frecuencia de corte normalizada de $f_c = 0.2$ (para una frecuencia de muestreo de 1), de orden $n = 2$ y permitiendo ondulaciones de 1dB en el paso de banda. Grafique la respuesta de frecuencia de ambos filtros Buterworth y Chebychev del mismo orden en el misma gráfico.
5. Aplique el filtro Chebychev a la señal con ruido “stairwayb.wav”. ¿Qué sucede con un orden de $n = 50$? ¿Ocurre lo mismo con el filtro Butterworth de igual orden?

4.3 Separación de fuentes y eliminación de ruido

1. Diseñe filtros IIR o ideales que permitan una separación grosera de las diferentes fuentes en las señales drum.wav y seq.wav.

2. Diseñe filtros para las señales ecg.npz y conso.npz para apreciar mejor la señal primaria (respectivamente los latidos del corazón y el uso de energía).