Guía Laboratorio 7
Procesamiento Digital de Señales

Paula Pérez, Alejandro Escobar y Cristian Ríos

2024-1

NOTAS:

- Enviar el informe del laboratorio con el siguiente nombre: *Lab7_PDS_Apellido_Nombre.ipynb*
- Enviar junto con el informe los archivos adicionales generados y descargados. Todo esto debe ir en un archivo comprimido con el siguiente nombre: *Lab7_PDS_Apellido_Nombre.zip*
- OJO! Recuerde tener cuidado con la indentación y caracteres como el guión bajo y las llaves cuando copie y pegue el código entregado en esta guía.
- Las preguntas deberán ser resueltas en el notebook indicando sus respectivos numerales.

1. Transformada Rápida de Fourier para señales discretas

La Transformada Rápida de Fourier (Fast Fourier Transform - FFT) es una herramienta fundamental en el procesamiento digital de señales. La técnica fue propuesta en 1965 por J. W. Cooley y J. W Tukey, quienes abordaron por primera vez el problema de la programación de un algoritmo para el cálculo de series complejas.

Debe quedar claro que la FFT no es una nueva transformada sino que se trata de un algoritmo para el cálculo de la Transformada Discreta de Fourier (DFT). Su importancia radica en el hecho que elimina una gran parte de los cálculos repetitivos a los que se encuentra sometida la DFT, por lo tanto se logra un cálculo más rápido. Además, la FFT generalmente permite una mayor precisión en el cálculo de la DFT disminuyendo los errores de redondeo.

La Transformada de Fourier para la señal x[n] se define de acuerdo con la siguiente ecuación.

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{\frac{-j \cdot 2\pi k \cdot n}{N}} = \sum_{n=0}^{N-1} x[n] \cdot \mathbf{U}[k,n] \tag{1}$$

Donde $\mathbf{U} \in \mathbb{R}^{N \times K}$ representa una matriz de transformación, que se puede definir con la ecuación 2. Donde N es el número de muestras de la señal, y NFFT es el número de puntos de la Transformada de Fourier que se desea calcular (debe ser una potencia de 2).

$$\mathbf{U} = \frac{1}{\sqrt{N}} \left[\exp\left(\frac{j \cdot 2\pi k \cdot n}{N} \right) \right]_{n \in \{0,\, N-1\},\, k \in 0,\, \mathrm{NFFT}} \tag{2}$$

El número de puntos de la Transformada de Fourier define la resolución en frecuencia, donde cada índice mapea el contenido de frecuencia de la señal en un rango normalizado entre 0 y $2\pi\,$rad/seg. Cada índice también mapea la frecuencia de muestreo entre 0 y fs, por lo tanto, para encontrar la equivalencia entre la frecuencia normalizada, y la frecuencia en Hz, se debe multiplicar la frecuencia normalizada por $(fs/2)$. Luego de definir la matriz, se puede redefinir la Transformada de Fourier como una multiplicación de una matriz y un vector (columna).

$$X = \mathbf{U}^* \cdot \mathbf{x} \tag{3}$$

La siguiente función construye la matriz $\mathbf{U}$ de transformación.

```
def dftmatrix(N, Nfft):
    #construct DFT matrix
    k= np.arange(Nfft)
    if N is None: N = Nfft
    n = np.arange(N)
    U = np.matrix(np.exp(1j* 2*np.pi/Nfft *k*n[:,None]))
    return U/np.sqrt(Nfft)
```

Para calcular la Transformada de Fourier de la señal x[n] se usa la siguiente expresión.

```
X=U.H*x # donde U.H significa el complejo conjugado de la matriz U
```

2. Generación de Señales y Transformada de Fourier

1. Teniendo un sistema con $f_s = 4\,\text{kHz}$, cree un vector tiempo de 0.9 segundos y genere las siguientes señales de acuerdo con el último número de su cédula (C).

   Pares:

   - $x_1(t) = \sin(2\pi f_1 t)$
   - $x_2(t) = \sin(2\pi f_2 t)$
   - $x_3(t) = \sin(2\pi f_3 t)$
   - $x_4(t) = \sin(2\pi f_4 t)$

   Impares:

   - $x_1(t) = \cos(2\pi f_1 t)$
   - $x_2(t) = \cos(2\pi f_2 t)$
   - $x_3(t) = \cos(2\pi f_3 t)$
   - $x_4(t) = \cos(2\pi f_4 t)$

   Teniendo:
   $f_1 = (1 + C) * 16$
   $f_2 = 3 * C + 20$
   $f_3 = (3 + C) * 50 - 90$
   $f_4 = 5 * C + 80$.

2. Grafique cada señal en una sola figura.

3. Calcule la Transformada de Fourier usando **U** y Nfft = 4096 para cada señal.

4. Usando subplots, grafique parte par e impar de la Transformada de Fourier. ¿Qué se puede concluir al respecto?.

Ejemplo de una señal sinusoidal con una frecuencia de 60 Hz, $f_s = 600$ Hz, y NFFT = 512.

```
fs=600.0 # frec de muestreo
f=60.0 # frec de la senal
nfft=512 # numero de puntos de la transformada
t = np.arange(0, 0.1, 1/fs) # vector de tiempo
xt=np.sin(2*np.pi*f*t) # senal en el tiempo
xt.shape=(len(xt),1) # Convierto en vector columna para poder multiplicar con la matriz
U=dftmatrix(len(xt), nfft) # calculo la matriz de transformacion
Xf=U.H*xt[:] # calculo la fft
freq=np.hstack((np.arange(0, nfft/2-1), np.arange(-nfft/2, 1)))*fs/nfft # vector de frecuencias ...
    para poder graficar la fft en Hz
# genero graficas
plt.subplot(211)
plt.plot(t, xt)
plt.xlabel('Tiempo (seg)')
plt.subplot(212)
plt.plot(freq, np.imag(Xf))
plt.xlabel('Frecuencia (Hz)')
plt.show()
```

Los resultados del ejemplo se observan en la Figura 1.

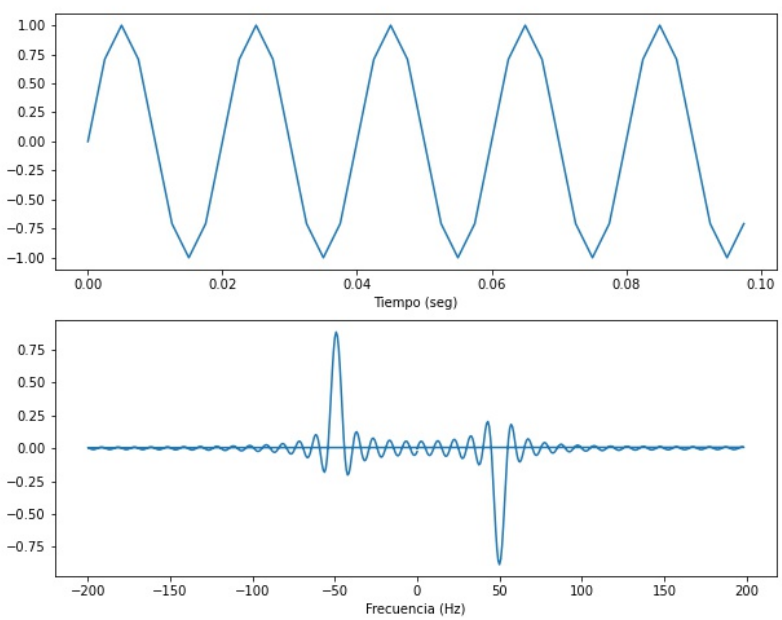

Figura 1: Ejemplo cálculo de FFT.

5. Calcule la señal $x_{\mathrm{T}}(t) = x_1(t) + x_2(t) + x_3(t) + x_4(t)$
6. Grafíquela la señal en el tiempo, parte par y parte impar de la Transformada. ¿Qué se puede concluir?
7. A dos de las cuatro señales, agréguele un offset de fase. Ejemplo:

```
x2=np.sin(2*np.pi*f2*t - np.pi)
```

8. Repita el paso 5 y 6 con la nueva señal.

3. Recuperación de la señal a partir de su Transformada de Fourier

1. Use la siguiente instrucción para calcular la Transformada inversa de Fourier en cada uno de los casos del ítem anterior.

```
inversa=np.fft.ifft(Xf.T) #Xf: Frequency-domain signal
```

2. Grafique los resultados usando subplots y compárelos. ¿Puede observar alguna diferencia?, ¿a qué se debe esto? Escriba sus conclusiones.

4. Conclusiones

Realice conclusiones generales sobre la práctica. Recuerde que las conclusiones son parte fundamental de su evaluación en el laboratorio, tómese el tiempo de pensar las conclusiones.