Guía Laboratorio 8
Procesamiento Digital de Señales

Paula Pérez, Alejandro Escobar y Cristian Ríos

2024-1

NOTAS:

- Enviar el informe del laboratorio con el siguiente nombre: *Lab8_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: *Lab8_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 de Fourier de tiempo corto - Espectrogramas

1.1. Introducción

La transformada de Fourier de tiempo corto (Short-Time Fourier Transform, STFT) es un método de procesamiento de señales que nos permite obtener una representación tiempo-frecuencia denominada espectrograma que es usada para determinar el contenido de frecuencias armónicas y de fase en secciones locales de una señal. El cálculo de la STFT consiste en tomar un determinado número de muestras por medio de una ventana temporal, luego se halla el contenido de frecuencia (Ω) de la ventana y se representan en una gráfica de dos dimensiones (tiempo-frecuencia). En el caso de señales de audio la información a transformar es dividida en tramas (que usualmente se solapan unas con otras para reducir irregularidades en la frontera) y a cada una de estas se le realiza una transformación de Fourier a partir de la Ecuación 1.

$$X_m(\Omega) = \sum_{n=-\infty}^{\infty} x(n) f(n-m) e^{-j\Omega n} \tag{1}$$

En este caso, $x(n)$ representa la señal de audio y $f(n)$ la ventana utilizada. El índice de tiempo discreto m es normalmente considerado como un tiempo *lento* y usualmente no se expresa con tan alta resolución como con el tiempo n.
Finalmente, la respuesta en magnitud de la STFT es conocida como un espectrograma, el cual es una matriz que muestra la variación del espectro y la energía de la señal para cada una de las tramas a lo largo del tiempo, es común representar la energía de la señal en decibelios (dB) a partir de la Ecuación 2 para una mejor visualización en el espacio de color.

$$\text{Espectrograma}\{x(t)\} = 10 \log_{10} |X_m(\Omega)| \tag{2}$$

1.2. Carga y visualización de la señal

1. Cargue, normalice, grafique y escuche las señales compartidas ($senal1.wav$ y $senal2.wav$).

1.3. Extracción de segmentos

También llamado enventanado y es el proceso mediante el cual se segmenta una señal en N ventanas para luego aplicarle la STFT. Use la función mostrada a continuación y segmente su señal con un tamaño de ventana de 40 ms. **Para este punto elija solo una de las dos señales.**

```
def extraer_ventanas(signal, size, fs):

    #Tamano de paso
    step=int(0.010*fs)


    n_seg = int((len(signal) - size) / step)

    # extraer segmentos
    windows = [signal[i * step : i * step + size]
               for i in range(n_seg)]

    # stack (cada fila es una ventana)
    return np.vstack(windows)
```

1.4. Aplicación de ventanas

Existen diferentes tipos de ventanas como Hamming, Hanning, Blackman, a través de las cuales se puede establecer el grado de resolución tanto de tiempo como de frecuencia que se desee. Si la ventana es muy angosta analizaremos una porción muy pequeña de la señal lo que nos permite tener una buena resolución en tiempo, pero una mala resolución en frecuencia, ya que conoceremos solo una mínima fracción del espectro total de la señal. Por otro lado, si la ventana es muy ancha tendremos una buena resolución en frecuencia pero una mala resolución en tiempo.

1. Consulte como se construye y se aplica una ventana Hamming a una señal. ¿Qué tamaño de ventana debe usar?
2. Aplique la ventana construida para cada uno de los segmentos extraídos anteriormente.
3. Usando un subplot, grafique un segmento antes y después de aplicarle la ventana Hanning. ¿Observa alguna diferencia? Explique.

1.5. Aplicación de la STFT

1. Usando la siguiente función calcule la STFT a cada uno de los segmentos del ítem anterior, para esto use un tamaño de NFFT de 512.

```
def potspec(X, size, n_padded_min=0):

    # Zero padding
    if n_padded_min==0:
        n_padded = max(n_padded_min, int(2 ** np.ceil(np.log(size) / np.log(2))))
    else:
        n_padded = n_padded_min

    # Transformada de Fourier
    Y = np.fft.fft(X, n=n_padded)
    Y = np.absolute(Y)

    # non-redundant part
    m = int(n_padded / 2) + 1
    Y = Y[:, :m]
    Img=Y.imag
    Real=Y.real
    spec= np.sqrt(Real**2+ Img**2)

    return spec, n_padded
```

2. Grafique la representación tiempo-frecuencia obtenida en el ítem anterior. Puede ayudarse de las siguientes líneas de código.

```
espectro,nfft=potspec(frames,size,1024)
espectro=np.flipud(10*np.log10(espectro).T)
plt.figure(figsize=(20,10))
plt.title('Espectrograma')
plt.imshow(espectro,aspect='auto', extent=[0, len(signal)/fs, 0,  fs//2])
plt.ylabel('Frequency [Hz]', fontsize=18)
plt.xlabel('Time [sec]', fontsize=18)
plt.show()
```

3. ¿Qué puede concluir del espectrograma? ¿En qué intervalos de frecuencia está definida la señal? Explique.

4. Calcule y grafique la STFT para valores de NFFT equivalentes a 64, 1024 y 4096. ¿Qué diferencia encuentra al variar este tamaño? ¿A su criterio cuál es el tamaño de NFFT que tiene una mejor resolución tiempo-frecuencia? Explique.

5. Con la mejor resolución del numeral anterior, grafique las dos señales. Que se puede concluir?

6. Grafique el espectrograma de ambas señales a una NFFT de 512. Que puede concluir? En que se diferencian?

2. 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.