

Guía Laboratorio
Procesamiento Digital de Señales
Salomón Santiago García
C.C: 1001003575
2023-1

NOTAS:

- Enviar el informe del laboratorio con el siguiente nombre: Lab_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 guion bajo y las llaves cuando copie y pegue el código entregado en esta guía.
- Las preguntas deberían ser resueltas en el notebook indicando sus respectivos numerales

1. Transformadas 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.

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).

1.2.Carga, visualización y extracción de los segmentos del audio

1. Cargue, normalice, grafique y escuche la señal compartida (audio5.wav).
2. Use la función mostrada a continuación y segmente su señal con un tamaño de ventana de 50ms.

```
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)
```

1.3.Aplicación de la STFT

1. Usando la siguiente función puede calcular la STFT al segmento del ítem anterior:

```
def potspec(X, size, n_padded_min=0):
    # Zero padding para la proxima potencia de 2
    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]
    return np.abs(Y) ** 2, n_padded
```

2. Grafique la representación tiempo-frecuencia con el segmento del ítem anterior con un tamaño de NFFT de 512. Puede ayudarse de las siguientes líneas de código:

```
espectro,nfft=potspec(x,size,512)
espectro=np.flipud(10*np.log10(espectro).T)

plt.figure(figsize=(20,10))
plt.title('Espectrograma')
plt.imshow(espectro,aspect='auto', extent=[0, len(archivo)/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 de audio? Explique.

4. Calcule y grafique la STFT para valores de NFFT equivalentes a 206, 1024 y 2048. ¿Qué diferencia encuentra al variar este tamaño, si es notorio? ¿A su criterio cual es el tamaño de NFFT que tiene una mejor resolución tiempo-frecuencia? Explique.

2. Filtrado de música con filtros FIR

De manera general, existen 4 familias principales de instrumentos musicales: viento, cuerda, percusión y los eléctricos. Aunque se consideran más categorías, estas son las más usadas. Ejemplos de instrumentos en estas categorías son:

- Viento: saxofón, flauta, clarinete, trompeta, oboe.
- Cuerda: guitarra, arpa, violín, piano de cuerdas percutidas, tiple.
- Percusión: timbal, tambor, platillos, bombo, piano.
- Instrumentos eléctricos: bajo eléctrico, guitarra eléctrica, sintetizador.

1. Con el audio del numeral anterior, normalice la señal en amplitud, elimine su nivel DC y cree su vector de tiempo dependiendo de su frecuencia de muestreo. Vuelva a reproducirlo e indique que instrumentos se logran escuchar.
2. Diseñe un filtro FIR pasa-bajas donde la frecuencia de corte a emplear la escogerá el estudiante a partir de los espectrogramas realizados en el inicio de la guía. Utilice una banda de transición de 200 Hz y un ripple de 60 dB. Para el diseño del filtro se puede apoyar en el siguiente código:

```
from scipy.signal import kaiserord, lfilter, firwin, freqz

nyq_rate = fs / 2.0
roll_off = 200.0
cutoff_hz = 7000.0
width = roll_off/nyq_rate
ripple_db = 60.0                                     #The desired attenuation in the stop band, in dB.
N, _ = kaiserord(ripple_db, width)                    # Compute the order and Kaiser parameter for the FIR filter.
taps = firwin(N, cutoff_hz/nyq_rate, pass_zero='lowpass')
w, h = signal.freqz(taps, [1], worN=2000)
plt.plot(nyq_rate*w/np.pi, np.abs(h))
```

3. Pase la señal de audio por el filtro diseñado y escúchelo. ¿Qué instrumentos logró escuchar? ¿Alguno dejó de sonar? ¿Explique?
4. Si ahora realiza el mismo ejercicio de los ítems 2 y 3 para frecuencias de corte de 400 Hz y 20000 Hz ¿Se pudo separar otro instrumento para este caso? Explique que ocurre en cada frecuencia.
5. Explique en las conclusiones que información terminó brindando el espectrograma y que aspectos hay que tener en cuenta para diseñar un filtro a partir de este

3. 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.