

Guía Laboratorio Filtros

Procesamiento Digital de Señales

Hanna Isabella Colorado Martinez.

C.C:1000408557

2023-1

NOTAS:

- Enviar el informe del laboratorio con el siguiente nombre: Laboratorio_filtros_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: Lab_filtros_PDS_Apellido_Nombre.zip
- Las preguntas deberán ser resueltas en el notebook indicando sus respectivos numerales.

Introducción:

- En este laboratorio se pretende proporcionar un espacio para la práctica y experimentación en el manejo y implementación de los filtros FIR y IIR; los estudiantes trabajarán con diferentes técnicas de filtrado y aprenderán a extraer información en base a los diagramas de amplitud y espectrogramas de las señales filtradas, finalmente, se introducirá una nueva clase de filtro.

1. Carga y visualización de la señal

Nota:

Se pide para este laboratorio instalar la librería “Librosa” usando el siguiente comando: “pip install librosa”

- 1.1. Cargue, normalice y reproduzca una canción de su preferencia, cree el vector tiempo de la señal y extraiga su frecuencia de muestreo; grafique la señal en términos de la amplitud y el tiempo.
(Puede utilizar la función: **librosa.util.normalize()** consulte que es lo que hace esta función.)

Nota: si el audio es muy grande grafique y trabaje con un rango de su preferencia de la señal

- 1.2. verifique el rango de normalización de la señal.
- 1.3. Grafique el espectrograma de la señal en un rango de su preferencia, ¿Qué puede interpretar de este?

2. Diseño de Filtros con Respuesta Finita al Impulso (FIR)

Los Filtros con Respuesta Finita al Impulso (**FIR**) son un tipo de filtro digital utilizado en el procesamiento de señales. Estos filtros se caracterizan por tener una respuesta que se extiende solo durante un número finito de muestras.

El diseño de un filtro FIR implica determinar los coeficientes adecuados para lograr la respuesta deseada. Hay diferentes métodos para diseñar filtros FIR, como el método de la ventana, el método de Parks-McClellan o el método de la respuesta al impulso inversa.

Una vez diseñado el filtro FIR, podemos implementarlo en el dominio del tiempo utilizando convolución. Cada muestra de salida se calcula multiplicando los coeficientes del filtro por las muestras de entrada correspondientes y sumando los productos ponderados.

Para este laboratorio se implementará el método de método de enventanado, una técnica ampliamente utilizada en la implementación de filtros FIR que ofrece varias ventajas significativas. las ventanas pueden mitigar el efecto de oscilaciones indeseables en la respuesta en frecuencia del filtro (**fenómeno de Gibbs**), mejorando la selectividad y la precisión. Esto permite obtener una mejor supresión de frecuencias no deseadas y una mejor aproximación a la respuesta deseada.

Ventana	Ecuación
Rectangular	$w(n) = 1, \text{ para } 0 \leq n < N$ $w(n) = 0, \text{ en cualquier otro caso}$
Hann	$w(n) = 0.5 * \left(1 - \cos\left(\frac{2\pi n}{N-1}\right)\right)$ para $0 \leq n < N$
Hamming	$w(n) = 0.54 - 0.4 * \cos\left(\frac{2\pi n}{N-1}\right)$ para $0 \leq n < N$
Blackman	$w(n) = 0.42 - 0.5 * \cos\left(\frac{2\pi n}{N-1}\right)$ $+ 0.08 * \cos\left(\frac{4\pi n}{N-1}\right)$ para $0 \leq n < N$
Flattop	$w(n) = a_0 - a_1 * \cos\left(\frac{2\pi n}{N-1}\right) + a_2$ $* \cos\left(\frac{4\pi n}{N-1}\right) - a_3$ $* \cos\left(\frac{6\pi n}{N-1}\right) + a_4$ $* \cos\left(\frac{8\pi n}{N-1}\right)$

	para $0 \leq n < N$
--	---------------------

2.1. Procedimiento

2.1.1. Diseñe un filtro FIR que funcione como un pasa-bajas por el método de enventanado, con una banda de transición de 300 Hz, una frecuencia de corte de 2KHz y un ripple de 25dB. (Utilice las ventanas: hanning, blackman y flattop).

2.1.2. Grafique el espectro de amplitud de la señal antes de aplicar el filtro.

Nota: funciones implicadas.

- `plt.plot(freqs, magnitude**2)`
- `plt.plot(freqs, 20 * np.log10(magnitude)).`
- `plt.plot(freqs, magnitude)`

2.1.3. Grafique el espectro de amplitud de los tres métodos de enventanado anteriormente mencionados aplicando la transformada rápida de Fourier(FFT).

2.1.4. Basándose en las gráficas: ¿Qué se puede apreciar en los diagramas de amplitud?

2.1.5. Grafique y escuche las señales filtradas ¿Qué cambios hubo en el audio?

Nota: Escuche una por una, pero solo deje en el laboratorio un solo audio de la ventana de su preferencia, esto debido a temas del tamaño del archivo y el Github.

2.1.6. Grafique los espectrogramas resultantes de la señal después de ser filtrada por cada una de las ventanas, compárelos y diga: ¿Qué puede inferir de estos?

2.1.7. ¿Qué efecto se tendría en el filtrado de la señal si se modifica el ripple?

2.1.8. Utilice la ventana hamming y con una frecuencia de corte de 1kHz, escuche la señal y diga que cambios pueden apreciarse.

2.1.9. Consulte acerca de la ventana: “flattop” y diga si considera que los resultados fueron concluyentes con el funcionamiento de esta ventana.

2.1.10. Utilizando la ventana Hamming, implemente un filtro pasa-altas; filtre y escuche la señal ¿Qué puede escuchar?

2.1.11. Grafique sus espectros de amplitud y su espectrograma.

2.1.12. ¿Qué puede apreciar en los espectros de amplitud?

3. Diseño de Filtros con Respuesta Infinita al Impulso (IIR)

Los filtros IIR son un tipo de filtro digital ampliamente utilizado en el procesamiento de señales. A diferencia de los filtros FIR (Finite Impulse Response), los filtros IIR tienen una respuesta de impulso que se extiende infinitamente en el tiempo.

Son conocidos por su capacidad para proporcionar una respuesta de frecuencia más compleja y flexible en comparación con los filtros FIR debido a que los filtros IIR tienen la capacidad de utilizar realimentación en su estructura, lo que les permite tener polos y ceros en su función de transferencia.

La respuesta en frecuencia de un filtro IIR puede tener características como picos y valles, lo que permite una mayor capacidad de ajuste y adaptación a las necesidades específicas de la señal de entrada. Hay que tener en cuenta que los filtros IIR pueden ser más susceptibles a problemas de inestabilidad y oscilaciones no deseadas.

3.1. Procedimiento

- 3.1.1. Diseñe un filtro IIR pasa-bajas de tipo Butterworth, con una frecuencia de corte de 2kHz en un rango de su preferencia.
- 3.1.2. Escuche la señal filtrada; en comparación con el filtro pasa-bajas FIR ¿encuentra alguna diferencia en el audio?
- 3.1.3. Grafique los espectros de amplitud y potencia de la señal filtrada, en comparación con el filtro FIR anteriormente implementado ¿Qué puede resaltar?
- 3.1.4. Grafique el espectrograma de la señal resultante, teniendo en cuenta los resultados vistos en la gráfica del espectro de amplitud ¿Qué puede decir de este?
- 3.1.5. Por medio de un filtro pasa-bandas intente separar uno de los instrumentos presentes en el audio que escogió, consulte las frecuencias correspondientes de este instrumento y diga si fue posible separarlo.
- 3.1.6. Grafique el espectrograma de la señal resultante.

4. Aplicación filtro de distorsión.

Los filtros de distorsión son componentes son usados normalmente en el ámbito de la música y la producción de sonido. Se utilizan para alterar y modificar el timbre y la calidad del sonido de una señal de audio, añadiendo efectos de distorsión característicos.

Existen diferentes tipos de filtros de distorsión, algunos ejemplos incluyen el filtro de distorsión de sobremarcha (overdrive), el filtro de distorsión de fuzz y el filtro de distorsión de saturación (saturation). Estos filtros pueden modificar tanto las frecuencias altas como las bajas de una señal de audio.

4.1. Implementación:

- En este ejemplo, utilizamos la función `np.tanh()` para aplicar una distorsión no lineal a la señal en el rango especificado. Puede guiarse del siguiente código para implementar el filtro de distorsión.

```
def filtro_distorcion(señal_entrada, inicio, final, gain=1.0):
    # Aplicar el filtro de distorsión utilizando la función wave
    distorted_signal = np.copy(señal_entrada)
    distorted_range = distorted_signal[inicio:final]
    distorted_range = np.tanh(distorted_range * gain) #la tanh aplica distorsión no lineal a las
    señal.
    distorted_signal[inicio:final] = distorted_range

    return distorted_signal
```

4.2. Procedimiento:

- 4.2.1. Tome la señal anteriormente filtrada con la ventana hamming de los filtros FIR y aplique el filtro de distorsión.
- 4.2.2. Grafique el espectrograma de la señal filtrada y escúchela, ¿Qué diferencias encuentra con respecto a los resultados del filtro anterior?

5. Conclusiones:

- Realice conclusiones generales sobre la práctica. Recuerde que las conclusiones son parte fundamental de su evaluación en el laboratorio

