



## DISEÑO DE FILTROS DIGITALES

### 1. OBJETIVOS

#### 1.1 General

Diseñar, aplicar y analizar filtros digitales en Python.

#### 1.2 Objetivos Específicos

- Diseñar filtros digitales.
- Analizar los comandos de Python para aplicar filtros.

### 2. MATERIALES Y EQUIPOS

Para la realización de esta práctica se requiere del paquete computacional de Python básico.

### 3. MARCO TEORICO

Los filtros digitales permiten la manipulación de características frecuenciales de señales digitales debidamente muestreadas. Frente a los filtros analógicos cuentan con las siguientes ventajas:

- **Respuesta dinámica:** El ancho de banda está limitado por la frecuencia de muestreo de la señal en lugar del comportamiento eléctrico de los amplificadores operacionales.
- **Intervalo dinámico:** El número de bits usados durante la cuantización de la señal son los que determinan el rango mínimo de valores que producirá el filtro.
- **Conmutabilidad:** Es posible almacenar en un sistema digital los parámetros del filtro y aplicarlo a múltiples señales.
- **Adaptabilidad:** Se pueden construir en software o hardware.
- **Ausencia de problemas por los componentes:** Al no usar componentes físicos no hay problemas con las tolerancias de estos, ni con el manejo de impedancia o el acople de etapas.
- **Complejidad:** Dadas las capacidades de cálculo actuales se pueden implementar filtros con órdenes imposibles de alcanzar con sistemas analógicos.

Los tipos de filtros digitales básicos se distinguen de acuerdo con su respuesta al impulso. Si se parte de la ecuación en diferencias:

$$y_k = a_1 y_{k-1} + a_2 y_{k-2} + \dots + a_n y_{k-n} + b_0 x_k + b_1 x_{k-1} + \dots + b_m x_{k-m} \quad (1)$$

Se tiene que para el caso en que todos los  $a_n$  son cero se obtiene un *filtro de respuesta finita al impulso (FIR) o filtro no recursivo*. Un filtro FIR de orden  $m$  está dado como:

$$y_k = b_0 x_k + b_1 x_{k-1} + \dots + b_m x_{k-m} \quad (2)$$

Con función de transferencia:

$$H(z) = b_0 + z^{-1}b_1 + \dots + z^{-m}b_m \quad (3)$$

Por otro lado, cuando hay valores  $a_n$  diferentes de cero se obtiene un *filtro de respuesta infinita al impulso (IIR) o filtro recursivo*. En este caso:

$$H(z) = \frac{b_0 + z^{-1}b_1 + \dots + z^{-m}b_m}{1 + z^{-1}a_1 + \dots + z^{-n}a_n} \quad (4)$$

Se debe tener en cuenta que:

- Los *filtros IIR* producen en general distorsión de fase, es decir la fase no es lineal con la frecuencia.
- Los *filtros FIR* se pueden calcular a partir de los *filtros IIR*, no se cumple en el caso contrario.
- Los *filtros FIR* son de fase lineal.
- El orden de un *filtro IIR* es mucho menor que uno *FIR* para unas especificaciones dadas.
- Los *filtros FIR* son siempre estables.

## 4. PROCEDIMIENTO

*SciPy* es un entorno de procesamiento de datos y creación de prototipos de sistemas para Python que contiene módulos para optimización, álgebra lineal, integración, interpolación, FFT, procesamiento de señales e imágenes, y otras tareas para la ciencia e ingeniería.

El módulo de Procesamiento de Señales es importado así:

```
import scipy.signal as signal
```

Dentro de este módulo se encuentran comandos para la convolución, diseño de filtros, filtrado, Wavelets, análisis espectral, entre otros.

### 4.1. Creación de la señal

Proponga una secuencia sinusoidal  $x_e(n)$ , con  $0 \leq n \leq 1000$ ,  $F_0 = 0,01\text{Hz}$  y  $F_s = 25\text{Hz}$ . Asuma que esta señal es una señal biológica, la cual se origina por algún proceso electrofisiológico, y que se registrará sobre la superficie de la piel mediante un sistema electrónico. El viaje de la señal hipotética  $x_e(n)$  a través de los diversos tejidos genera

una atenuación en la amplitud y un retraso temporal. El efecto del sistema electrónico de registro sobre la señal de interés es la adición de ruido gaussiano a la señal. La señal registrada finalmente es de la forma:

$$y_e(n) = \alpha x_e(n - D) + w(n)$$

Donde  $\alpha$  representa la amplitud,  $D$  es un retraso debido a la conducción a través de los diversos tejidos y  $w(n)$  representa el ruido generado por la electrónica de los sistemas de adquisición.

- Genere las secuencias ficticias  $x_e(n)$  y  $y_e(n)$ , implemente  $w(n)$  como una secuencia de ruido Gaussiano con media cero y varianza  $\sigma^2 = 0,01$ , establezca  $\alpha = 0,8$  y un retraso  $D = 15$  muestras. Grafique ambas señales. Nota: Para generar el ruido gaussiano use la función de numpy: `np.random.normal`.

## 4.2. Diseño de un filtro

La función `firwin` permite diseñar un filtro FIR utilizando el método de ventanas, donde los principales argumentos de entrada son  $N$ : Orden del filtro y  $W_n$ : Frecuencia de corte. Para filtros digitales  $W_n$  está normalizado entre 0 y 1, donde 1 es la frecuencia de Nyquist que es igual a la mitad de la frecuencia de muestreo de la señal.  $W_n$  es igual a la frecuencia de corte del filtro dividido la frecuencia de Nyquist.

- Diseñe un filtro FIR pasa bajas para la señal  $y_e(n)$ , con una frecuencia de corte de 1.25 muestras/s y orden 30.
- ¿Qué cambios hay en los argumentos de la función para implementar un filtro pasa banda y pasa altas? Pruebe con una frecuencia de corte adicional de 7.5 muestras/s.

Nota: Resultado esperado diseño de 3 filtros FIR (pasa bajas, pasa altas y pasa banda)

La función `iirfilter` permite diseñar un filtro IIR.

- Diseñe un filtro IIR en las mismas condiciones de los puntos anteriores. Compare cambios.

Nota: Resultado esperado diseño de 3 filtros IIR (pasa bajas, pasa altas y pasa banda)

## 4.3. Filtrado

Las funciones `lfilter` y `filtfilt` filtran una secuencia de información  $x(n)$  usando un filtro IIR o FIR.

- Pruebe dichos comandos utilizando la información entregada por el filtro pasa bajas FIR implementado anteriormente y la señal  $y_e(n)$  ¿Por qué se utiliza sólo el numerador? ¿A qué hace referencia el numerador? ¿Evidencia cambios en el uso de las funciones de filtrado? ¿Cómo se puede explicar la falta de coincidencia? ¿Es adecuado el orden del filtro usado? Analice.

- Pruebe dichos comandos utilizando la información entregada por el filtro pasa bajas IIR implementado anteriormente y la señal  $y_e(n)$  ¿Por qué se utiliza el numerador y denominador? **¿Es necesario cambiar el orden del filtro? En tal caso hágalo y analice.**

#### 4.4 Aplicación

Cargue la señal *señal\_filtros.txt* adquirida a una frecuencia de muestreo de 500 Hz. Extraiga y trabaje con el **canal 1** de la señal.

- Realice un análisis frecuencial con la función que permite calcular el periodograma de Welch de una señal ¿Hay presencia de ruido eléctrico? Analice.
- Diseñe un filtro FIR que permita eliminar el ruido de la señal. Grafique. Luego verifique nuevamente el comportamiento en frecuencia (periodograma de Welch de la señal filtrada) ¿Hay cambios? Analice.
- Grafique un segmento de las señales antes y después de filtrar, analice los cambios.

### 5. INFORME

Realice todos los procedimientos en un Colab, responda las preguntas planteadas y anexe conclusiones y referencias. Por favor ser muy ORDENADOS, incluir enunciados para yo saber qué están haciendo en cada parte.

### 6. ARCHIVOS ADJUNTOS

Archivo de texto con la señal a filtrar: *senalfiltros.txt*

### 7. BIBLIOGRAFIA

Oliphant, T. E. (2007). SciPy: Open source scientific tools for Python. Computing in Science and Engineering, 9, 10–20. <http://doi.org/10.1109/MCSE.2007.58>

Tong, S., & Thakor, N. V. (2009). Quantitative EEG analysis methods and clinical applications. Artech House.