

Técnicas Digitales III

Trabajo práctico: Filtrado digital FIR

1) Filtro Moving Average con señales senoidales en Python

a) Genere una señal senoidal con frecuencia fundamental $f_n = 100$ Hz. Elija una frecuencia de muestreo adecuada.

b) Agregue ruido gaussiano a la señal senoidal tal que la relación señal-ruido entre la señal senoidal y la señal con ruido sea de 15 dB.

c) Calcule el valor máximo del orden del filtro (N_{\max}) $f_{co} = 2 f_n$.

d) Aplique filtrado del tipo moving average a la señal con ruido para un filtro MA con dimensión igual $N = N_{\max}$. Utilice la función `lfilter()` de `scipy.signal`:

```
from scipy.signal import lfilter
y = lfilter(b, a, x)
```

e) Grafique la respuesta en frecuencia y fase del filtro MA. Use la función `freqz()` de `scipy.signal`:

```
from scipy.signal import freqz
import matplotlib.pyplot as plt

w, h = freqz(b, a, worN=n, fs=2 * np.pi)

plt.plot(w, 20 * np.log10(abs(h)))
plt.title('Respuesta en frecuencia')
plt.xlabel('Frecuencia [radianes/muestra]')
plt.ylabel('Amplitud [dB]')
plt.grid()
plt.show()
```

f) Grafique las señales en el dominio del tiempo sin ruido, con ruido y filtrada, y compare las tres.

g) Grafique la respuesta en frecuencia de las señales original y filtrada y compare.

```
import numpy as np

frequencies = np.fft.fftfreq(len(x), d=sample_spacing)
import matplotlib.pyplot as plt

plt.plot(frequencies, np.abs(Y))
plt.xlabel('Frecuencia [Hz]')
plt.ylabel('Amplitud')
plt.grid()
plt.show()
```

Donde `sample_spacing` es el espaciado entre muestras en la señal de tiempo (es decir, el inverso de la frecuencia de muestreo).

h) Repita los puntos d) a g) para $N = N_{\text{max}} / 2$ y $N = N_{\text{max}} * 10$.

2) Filtro Moving Average con señales de audio

a) Cargue el archivo de audio provisto llamado Tchaikovsky.mat.

```
import scipy.io

data = scipy.io.loadmat('Tchaikovsky.mat')

# Suponiendo que la matriz se llama 'Tchaikovsky' dentro del archivo .mat
signal = data['Tchaikovsky']
```

Se cargará la matriz `signal` con dos canales (estéreo). La variable $F_s = 44100$. Elija 1 de los 2 canales disponibles.

b) Agregue ruido gaussiano a esta señal tal que la relación señal-ruido entre la señal y la señal con ruido sea de 50 dB.

c) Calcule el valor máximo de N (N_{max}), con las frecuencias $f_s = F_s$ y $f_c = 11050$ Hz.

d) Aplique filtrado del tipo moving average a la señal con ruido para un filtro MA con dimensión igual $N = N_{\text{max}}$. Utilice la función `lfilter()` de `scipy.signal`.

e) Utilice la biblioteca `sounddevice` para reproducir las señales sin ruido, con ruido y filtrada.

```
import sounddevice as sd

# Asumiendo que y es tu matriz (o vector) de audio y Fs es la
# frecuencia de muestreo
sd.play(y, Fs)
sd.wait() # Espera a que termine la reproducción
```

f) Grafique la respuesta en frecuencia de las señales original y filtrada y compare.

h) Repita los puntos de d) a f) para $N = N_{\text{max}} / 2$ y $N = N_{\text{max}} * 10$.

3) Filtrado por ventanas con señales de audio

a) Use la biblioteca `scipy.signal` para diseñar un filtro:

- ☐ Pasa-banda.
- ☐ Ventana Blackman, orden 10.
- ☐ Frecuencias de corte de 300 Hz y 3400 Hz (canal telefónico), con formato punto flotante, precisión doble (valor por defecto).
- ☐ Frecuencia de muestreo 44100 Hz.

```
from scipy.signal import firwin

numtaps = 10 # número de coeficientes del filtro
fir_coefficients = firwin(numtaps, [f1, f2], window='blackman',
pass_zero=False)
```

Para visualizar la respuesta en frecuencia del filtro diseñado.

```
from scipy.signal import freqz
import matplotlib.pyplot as plt

w, h = freqz(fir_coefficients, worN=8000)
plt.plot(0.5*w/np.pi, np.abs(h))
```

b) Aumente el orden del filtro a 50. ¿Se modifica la respuesta en frecuencia del filtro?.

c) Utilice como señal de entrada el archivo `Tchaikovsky.mat`. Aplique a la señal de interés el filtro diseñado en el punto b) haciendo:

```
from scipy.signal import convolve

# Convolución entre la primera columna de signal y fir_coefficients
fir_output = convolve(matrix[:, 0], fir_coefficients, mode='same')
```

d) Grafique los espectros de la señal original (`signal`) y filtrada (`fir_output`).

e) Examine ambas gráficas. ¿Qué diferencia observa entre ambas señales?.

4) Filtración de una Onda Senoidal con un Filtro FIR

- a) Generar una onda senoidal de 50 Hz con una frecuencia de muestreo de 1000 Hz y una duración de 1 segundo.
- b) Agregar ruido blanco gaussiano a la señal.
- c) Diseñar un filtro FIR pasa-bajo utilizando la ventana de Hamming.
- d) Aplicar el filtro a la señal ruidosa.
- e) Graficar la señal original, la señal con ruido, y la señal filtrada.