

tp1_ejercicio_4

June 13, 2024

0.0.1 Ejercicio 4

Tenemos el siguiente filtro

$$y[n] = \alpha y[n-1] + (1-\alpha)x[n]$$

con $|\alpha| < 1$.

Ejercicio 4, parte a Sabemos que la transformada zeta de una ecuación en diferencias general de la siguiente forma,

$$y[n] = -\sum_{k=1}^N \left(\frac{a_k}{a_0}\right) y[n-k] + \sum_{k=0}^M \left(\frac{b_k}{a_0}\right) x[n-k]$$

es la siguiente

$$H(z) = \frac{\sum_{k=0}^M b_k z^{-k}}{\sum_{k=0}^N a_k z^{-k}}$$

Siendo que la respuesta en frecuencia no es más que la evaluación de la transformada zeta en el círculo unitario, esta es

$$H(e^{j\omega}) = \frac{\sum_{k=0}^M b_k e^{-j\omega k}}{\sum_{k=0}^N a_k e^{-j\omega k}}$$

En nuestro caso, vamos que $N = 1$, $M = 0$, $a_0 = 1$, $a_1 = -\alpha$ y $b_0 = 1 - \alpha$.

La respuesta en frecuencia entonces es

$$H(e^{j\omega}) = \frac{1 - \alpha}{1 - \alpha e^{-j\omega}}$$

0.0.2 Ejercicio 4, parte b

Aplicamos una entrada $x[n] = \delta[n]$. Asumiendo $y[-1] = 0$, tenemos

$$\begin{aligned}n = 0, y[0] &= \alpha y[-1] + (1 - \alpha)\delta[0] = 1 - \alpha \\n = 1, y[1] &= \alpha y[0] + (1 - \alpha)\delta[1] = \alpha(1 - \alpha) \\n = 2, y[2] &= \alpha y[1] + (1 - \alpha)\delta[2] = \alpha^2(1 - \alpha)\end{aligned}$$

De manera general, vemos que la respuesta al impulso es

$$\alpha^n(1 - \alpha)u[n]$$

0.0.3 Ejercicio 4, parte c

Sabemos que la respuesta al impulso es la antitransformada de la respuesta en frecuencia.

Podemos encontrar la antitransformada basándonos en el par

$$a^n u[n] \Leftrightarrow \frac{1}{1 - ae^{-j\omega}}$$

Entonces,

$$(1 - \alpha)\alpha^n u[n] \Leftrightarrow \frac{(1 - \alpha)}{1 - \alpha e^{-j\omega}}$$

La respuesta al impulso es, entonces

$$y[n] = (1 - \alpha)\alpha^n u[n]$$

Ejercicio 4, parte d

```
[1]: import numpy as np
import plotly.graph_objs as go
import plotly.subplots as sp

# Definir la función lambda para la respuesta en frecuencia
H = lambda alpha, omega: (1-alpha) / (1-alpha* np.exp(-1j * omega) )

alpha = 0.95
omega = np.linspace(-np.pi, np.pi, 1000)
H_alpha = H(alpha, omega)

magnitud = np.abs(H_alpha)
fase = np.angle(H_alpha)

# Generar la señal cuadrada
N_fft = 1024
```

```

impulse_response = lambda N, alpha: (1 - alpha) * alpha ** np.arange(N)
signal = impulse_response(N_fft, alpha)

# Calcular la FFT de la señal
fft_signal = np.fft.fftshift(np.fft.fft(signal))
omega_fft = np.fft.fftshift(np.fft.fftfreq(len(signal))) * 2 * np.pi

magnitud_fft = np.abs(fft_signal)
fase_fft = np.angle(fft_signal)

# Crear la figura de comparación
fig = sp.make_subplots(rows=2, cols=1, subplot_titles=('Comparación de Magnitud', 'Comparación de Fase'))

fig.add_trace(go.Scatter(x=omega, y=magnitud, name='Teórico Magnitud'), row=1, col=1)
fig.add_trace(go.Scatter(x=omega_fft, y=magnitud_fft, name='FFT Magnitud', mode='lines'), row=1, col=1)
fig.add_trace(go.Scatter(x=omega, y=fase, name='Teórico Fase'), row=2, col=1)
fig.add_trace(go.Scatter(x=omega_fft, y=fase_fft, name='FFT Fase', mode='lines'), row=2, col=1)

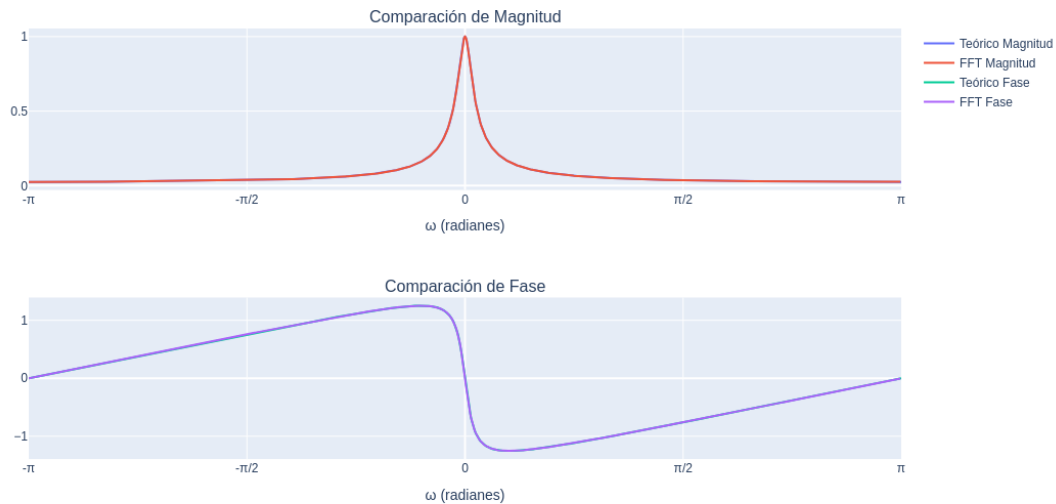
# Configurar los ticks del eje X en unidades de pi
pi_ticks = [-np.pi, -np.pi/2, 0, np.pi/2, np.pi]
pi_tick_labels = ['- ', '- /2', '0', ' /2', ' ']

fig.update_xaxes(
    tickvals=pi_ticks,
    ticktext=pi_tick_labels,
    title_text=' (radianes)',
    row=1, col=1
)
fig.update_xaxes(
    tickvals=pi_ticks,
    ticktext=pi_tick_labels,
    title_text=' (radianes)',
    row=2, col=1
)

fig.update_layout(height=600, width=800, title_text="Comparación de Respuesta en Frecuencia")
fig.show()

```

Comparación de Respuesta en Frecuencia



Ejercicio 4, parte e

¿A partir de qué valor de n (positivo), la respuesta se hace 0 (desde una perspectiva matemática)? ¿La salida actual depende solo de las entradas actuales y anteriores, o de algo más?

Desde un punto de vista estrictamente matemático, la respuesta nunca se hace 0, es decir, tiene una duración **infinita**. La salida depende de las entradas actuales y las anteriores, y en teoría, de **todas** las entradas pasadas, es decir, de toda la historia. De todas maneras, el peso de cada muestra decae exponencialmente con su distancia temporal.

¿Qué sucede en el caso del filtro de promedios móviles del ejercicio anterior? Si al filtro recursivo de este ejercicio se lo denomina IIR, ¿Cómo se denomina al filtro no-recursivo del punto anterior?

En el caso del filtro del ejercicio anterior, era claro a partir de que n se volvía 0 la respuesta, ya que esta era una señal cuadrada. En este caso el filtro se denomina FIR

Ejercicio 4, parte f

```
[2]: omega = np.linspace(-np.pi, np.pi, 1000)
alpha_values = np.linspace(0.5, 0.99, 5)

fig = sp.make_subplots(rows=1, cols=1, subplot_titles=('Magnitud Teórica'))

colors = ['blue', 'green', 'red', 'purple', 'orange']
for i, alpha in enumerate(alpha_values):
    H_alpha = H(alpha, omega)
    magnitud = np.abs(H_alpha)
```

```

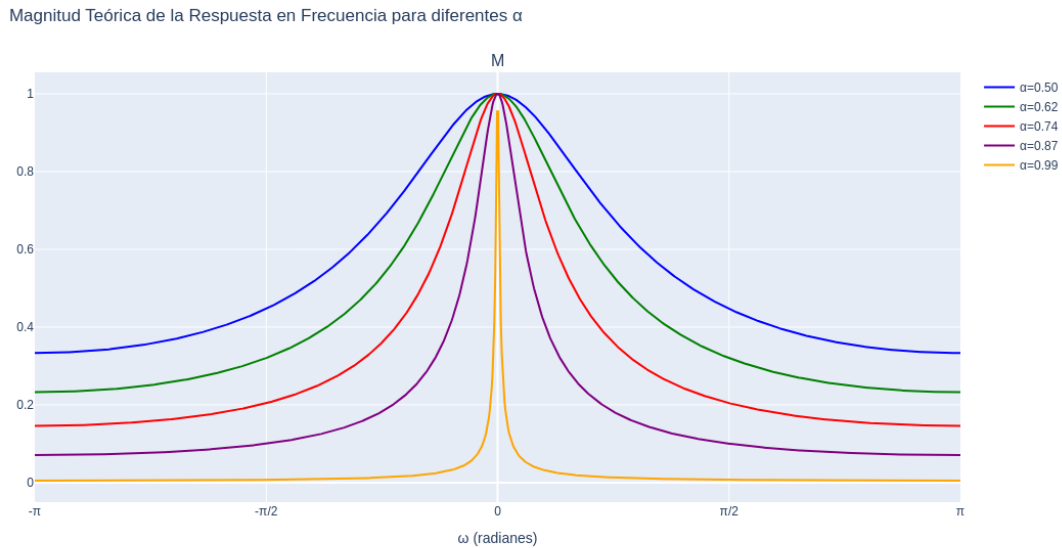
fig.add_trace(go.Scatter(x=omega, y=magnitud, mode='lines', name=f'α={alpha:
↪.2f}', line=dict(color=colors[i])), row=1, col=1)

# Configurar los ticks del eje X en unidades de pi
pi_ticks = [-np.pi, -np.pi/2, 0, np.pi/2, np.pi]
pi_tick_labels = ['- ', '- /2', '0', ' /2', ' ']

fig.update_xaxes(
    tickvals=pi_ticks,
    ticktext=pi_tick_labels,
    title_text=' (radianes)',
    row=1, col=1
)

fig.update_layout(height=600, width=800, title_text="Magnitud Teórica de la
↪Respuesta en Frecuencia para diferentes ")
fig.show()

```



Ejercicio 4, parte g Comparamos los anchos de banda a 3dB de los filtros. Este ancho de banda lo encontramos buscando la frecuencia en la que la magnitud cae en $\frac{1}{\sqrt{2}}$. Iterando, encontramos que

- Filtro IIR: ancho de banda con $\alpha = 0.95$: 0.0534 rad
- Filtro FIR: ancho de banda con $M = 55$: 0.05 rad

Vemos que para alcanzar el desempeño del IIR con $\alpha = 0.95$ necesitamos un promediador móvil de $M = 55$ muestras. En cuanto a complejidad, el IIR necesita un elemento de memoria y una

multiplicación por ciclo, mientras que el FIR requiere 55 de ambas.

Es claro que el IIR es la solución de menor complejidad. Qué filtro es preferido dependerá de las restricciones de desempeño. Es de interés resaltar que el IIR presenta fase no-lineal, lo que puede ser problemático.

Si la única restricción de diseño fuera la complejidad, el IIR es preferido.