🧩 CASE STUDIES IN FILTER DESIGN (Digital Domain)

🧩 Case 1: Low-Pass Filter for Smoothing a Noisy Signal

🎯 Requirement:

Remove high-frequency noise >1kHz from a speech signal sampled at 8kHz.

⚙ Filter Type:

FIR Low-pass filter

🧩 Insight:

Design $h[n]$ to suppress frequencies above cutoff. Sinc-based kernel tapered with window (to control side lobes).

◈ Python:

```
from scipy.signal import firwin, lfilter, freqz
import numpy as np
import matplotlib.pyplot as plt

fs = 8000          # Sampling frequency
cutoff = 1000      # Desired cutoff in Hz
numtaps = 51       # Filter length

h = firwin(numtaps, cutoff, fs=fs)

# Plot impulse response h[n]
plt.stem(h, basefmt=" ")
plt.title("Impulse Response h[n] - Low-Pass Filter")
plt.xlabel("n"); plt.ylabel("Amplitude"); plt.grid()
plt.show()

# Frequency response
w, H = freqz(h, worN=8000, fs=fs)
plt.plot(w, 20*np.log10(np.abs(H)))
plt.title("Frequency Response")
plt.xlabel("Frequency (Hz)"); plt.ylabel("Magnitude (dB)"); plt.grid()
plt.show()
```

🔍 Result:

- $h[n]$ looks like a sinc wave, windowed to suppress ripples.
- Frequency response shows clean cutoff at 1kHz.

◈ Case 2: High-Pass Filter to Remove DC/ Slow Drift

🎯 Requirement:

Remove low-frequency drift or DC bias (e.g., baseline wander in biosignals).

⚙ Filter Type:

FIR High-pass

🧭 Insight:

Design $h[n]$ to kill 0 Hz (DC). Usually symmetrical with alternating sign pattern.

🧭 Python:

```
h = firwin(51, 300, fs=2000, pass_zero=False)  # High-pass: cutoff = 300 Hz

plt.stem(h, basefmt=" ")
plt.title("h[n] - High-Pass Filter")
plt.xlabel("n"); plt.ylabel("Amplitude"); plt.grid()
plt.show()

w, H = freqz(h, worN=8000, fs=2000)
plt.plot(w, 20*np.log10(np.abs(H)))
plt.title("Frequency Response - High Pass")
plt.xlabel("Frequency (Hz)"); plt.ylabel("Magnitude (dB)"); plt.grid()
plt.show()
```

🔍 Result:

- $h[n]$ shows positive and negative swings (to cancel slow changes).
- DC (0 Hz) is strongly attenuated.

🟰 Case 3: Band-Pass Filter for Voice Detection (300–3000 Hz)

🎯 Requirement:

Extract the speech component from a broadband signal (as in SDR or telephony).

⚙ Filter Type:

FIR Band-pass

🧭 Insight:

Keep only the frequencies between 300–3000 Hz, reject below and above.

🔷 Python:

```
h = firwin(101, [300, 3000], fs=8000, pass_zero=False)

plt.stem(h, basefmt=" ")
plt.title("h[n] - Band-Pass Filter (300–3000 Hz)")
plt.grid(); plt.xlabel("n"); plt.ylabel("Amplitude")
plt.show()

w, H = freqz(h, worN=8000, fs=8000)
plt.plot(w, 20*np.log10(np.abs(H)))
plt.title("Band-Pass Filter Response")
plt.xlabel("Frequency (Hz)"); plt.ylabel("Magnitude (dB)")
plt.grid(); plt.show()
```

🔍 Result:

- $h[n]$ is a band-limited sinc-shaped waveform.
- Frequency response passes only 300–3000 Hz — ideal for voice.

⚫ Case 4: Notch Filter to Remove 50 Hz Powerline Interference

🎯 Requirement:

Eliminate 50 Hz hum from a signal (e.g., ECG or audio).

⚙ Filter Type:

FIR Band-stop (Notch)

🔷 Insight:

Design a dip in the frequency response centered at 50 Hz.

🔍 Python:

```
h = firwin(201, [48, 52], fs=1000, pass_zero=True)

plt.stem(h, basefmt=" ")
plt.title("h[n] - Notch Filter (50 Hz)")
plt.grid(); plt.xlabel("n"); plt.ylabel("Amplitude")
plt.show()

w, H = freqz(h, worN=8000, fs=1000)
plt.plot(w, 20*np.log10(np.abs(H)))
plt.title("Notch Filter Response (50 Hz)")
plt.xlabel("Frequency (Hz)"); plt.ylabel("Magnitude (dB)")
plt.grid(); plt.show()
```

🔍 Result:

- Deep notch around 50 Hz.
- $h[n]$ is symmetric and longer — needed for sharp notching.

🔍 Case 5: Matched Filter for Known Pulse Detection

🎯 Requirement:

Detect a known signal shape (e.g., radar return, chirp).

⚙ Filter Type:

Matched filter = time-reversed version of target signal

🔍 Insight:

Maximum correlation when signal and filter align.

💠 Python:

```
pulse = np.concatenate([np.ones(10), np.zeros(90)])  # Simulated short pulse
h = pulse[::-1]  # Matched filter: time-reversed

plt.stem(h, basefmt=" ")
plt.title("h[n] - Matched Filter")
plt.grid(); plt.xlabel("n"); plt.ylabel("Amplitude")
plt.show()
```

🔍 Result:

- Output of convolution peaks when the signal aligns with pulse.
- Used in radar, sonar, signal synchronization.

✅ Summary Table

Case	Filter Type	Purpose	Key Feature of h[n]
1	Low-pass	Remove noise	Sinc-like, smooth
2	High-pass	Remove drift	Alternating signs
3	Band-pass	Extract voice	Band-limited sinc
4	Notch	Kill 50 Hz	Deep zero at 50 Hz
5	Matched	Detect shape	Time-reversed target