🗒 🗒 Chapter 7.1 : Filter Theory & Concepts - Part II

FIR, IIR & Adaptive Filters

1. FIR Filters (Finite Impulse Response)

- Depend only on present + finite past inputs.
- Always stable, simple to reason about.
- Can be designed for **linear phase** (good for communication, radar, audio).

☞ Example: **Moving Average Filter**

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

# Moving average = simple FIR
b = np.ones(5)/5    # 5-point average
a = [1]             # no feedback

# Test signal: noise + sine
fs = 1000
t = np.linspace(0, 1, fs, endpoint=False)
x = np.sin(2*np.pi*50*t) + 0.5*np.random.randn(len(t))

y = lfilter(b, a, x)

plt.subplot(2,1,1); plt.plot(t, x); plt.title("Input: 50 Hz + noise")
plt.subplot(2,1,2); plt.plot(t, y); plt.title("Output: Smoothed (FIR moving average)")
plt.show()
```

☞ Here `b` = [1/5, 1/5, 1/5, 1/5, 1/5], `a` = [1].
This is literally “take average of last 5 samples.”

2. IIR Filters (Infinite Impulse Response)

- Depend on inputs **and** past outputs.
- More efficient → sharper cutoff for same order.
- But risk of instability.

☞ Example: **1st Order Low-Pass IIR**

```python
from scipy.signal import butter, lfilter

# Butterworth low-pass
fs = 1000
fc = 50
Wn = fc / (fs/2)
b, a = butter(1, Wn, btype='low')   # 1st order IIR

y_iir = lfilter(b, a, x)

plt.subplot(2,1,1); plt.plot(t, x); plt.title("Input: 50 Hz + noise")
plt.subplot(2,1,2); plt.plot(t, y_iir); plt.title("Output: IIR low-pass")
plt.show()
```

☞ Here `b, a` = feedback + feedforward.
You'll notice sharper filtering than moving average.

---

3. Adaptive Filters

Here coefficients **change in real time** depending on signal.
Classic example: **LMS (Least Mean Squares)** for noise cancellation.

Pseudo-code (we'll later code step-by-step):

```
Initialize weights w
For each sample:
    y[n] = w^T x[n]        # filter output
    e[n] = d[n] - y[n]     # error = desired - output
    w = w + μ * e[n] x[n] # update weights
```

Where:

- `d[n]` = desired signal (clean reference)

- `e[n]` = error used to adapt

☞ Real-world: your mic picks up music + noise. You feed noise reference, filter adapts to cancel.

---

4. Our Roadmap

- ✅ FIR with moving average
- ✅ IIR with butterworth
- 🔜 Adaptive with LMS
- Then → combine and try on **hello.iq**