🎯 Matched Filter: Summary for DSP/ SDR & IQ Applications

📄 What Is a Matched Filter?

A **matched filter** is a signal processing technique used to **detect a known signal pattern** (template) embedded in a longer signal, often corrupted by noise. It's a **correlation-based detector** that maximizes the signal-to-noise ratio (SNR) for that pattern.

🔍 Core Idea

> Slide the time-reversed version of a known signal across the input. Wherever there is a strong match, a **peak** appears in the output.

📐 Mathematically:

If known signal (template) is:

$$s[n] = [a, b, c]$$

Then matched filter is:

$$h[n] = [c, b, a] \quad (\text{time-reversed})$$

Output:

$$y[n] = x[n] * h[n] = \sum x[k] \cdot h[n - k]$$

⚙ How to Use It

1. **Define your known pattern**: `s = [1, 2, 1]`
2. **Reverse it**: `h = s[::-1]`
3. **Convolve with input signal**: `y = np.convolve(x, h, mode='full')`
4. **Locate peaks in** `y[n]` → pattern is detected
5. **Find pattern start index** using:

$$\text{start} = \text{peak index} - (L - 1)$$

🛠 Python Example

```
import numpy as np
import matplotlib.pyplot as plt

s = np.array([1, 2, 1])  # known pattern
x = np.array([0, 0, 1, 2, 1, 0, 0])  # received signal
h = s[::-1]  # matched filter
y = np.convolve(x, h, mode='full')

peak_idx = np.argmax(y)
start_idx = peak_idx - (len(h) - 1)

print("Peak at:", peak_idx)
print("Pattern starts at:", start_idx)
```

📊 What You'll See

- A **clear peak** at the point where the known pattern is detected.
- The **start of the pattern** occurs `len(h) - 1` samples **before the peak**.

🔍 Physical & SDR Interpretation

Concept	Meaning in SDR
Peak in `y[n]`	Pattern detected (e.g., preamble, sync signal)
Start index	Where IQ data of interest begins
Time reversal	Ensures energy is accumulated from front to back
SNR boost	Enhances detection of low-amplitude or buried signals
Multiple peaks	Multiple occurrences (e.g., packets, pulses)

🔍 Real-World SDR Applications

Application	Use of Matched Filter
Preamble detection	Detect sync sequence before packet
Burst detection	Find transient signals in wideband IQ
Radar pulse detection	Echo matching to transmitted pulse
Symbol synchronization	Locate exact symbol start
Spread spectrum (CDMA)	Match spreading codes

✅ Key Points to Remember

- Matched filter = time-reversed template.
- Output peak shows strongest correlation.
- Start of pattern = `peak index - (len(template) - 1)`
- It **maximizes SNR** for that specific signal pattern.
- Essential for any SDR receiver looking for known patterns in IQ streams.

🔄 Next Steps to Master It

Step	What to Do
1️⃣	Add **Gaussian noise** to `x[n]` and repeat detection
2️⃣	Try **multiple pattern instances** in a long signal
3️⃣	Use real-world patterns: e.g., BPSK symbols, sync preambles
4️⃣	Build a **threshold detector**: only trigger when `y[n] > T`
5️⃣	Extend to **complex IQ data** (real + imaginary part)

Would you like this:

- As a **PDF or Markdown export**?
- With **code blocks for each case** (cleaned up)?
- Connected to a **small SDR lab** (e.g., detect a chirp or tone burst in IQ)?

You've just mastered one of the most foundational DSP concepts — and one that directly translates to SDR front-end design and standard development.