⚙ Hypothetical DSP System: Stage-by-Stage Convolution

We'll model a signal going through 4 stages:

1. **Pre-emphasis / high-pass** (removes DC or slow variations)
2. **Smoothing / low-pass** (reduces noise)
3. **Matched filter** (detects a known pattern)
4. **Channel model** (distortion due to echo/fading)

Each stage has a different $h[n]$ and purpose.

🔹 Stage 1: Pre-emphasis / High-pass Filter

Goal:

Remove slow-varying or DC components.

Impulse Response:

$$h_1[n] = [1, -1]$$

This is a **first-difference filter**.

Input:

$$x[n] = [1, 1, 2, 4, 7]$$

Output $y_1[n] = x[n] * h_1[n]$:

$$
\begin{aligned}
y_1[0] &= 1 \times 1 = 1 \\
y_1[1] &= 1 \times (-1) + 1 \times 1 = 0 \\
y_1[2] &= 2 \times 1 + 1 \times (-1) = 1 \\
y_1[3] &= 4 \times 1 + 2 \times (-1) = 2 \\
y_1[4] &= 7 \times 1 + 4 \times (-1) = 3 \\
y_1[5] &= 7 \times (-1) = -7
\end{aligned}
$$

✅ **Effect**: Converts slowly increasing signal into its *rate of change*. Enhances fast transitions.

🔷 Stage 2: Smoothing / Low-pass Filter

Goal:

Reduce fast noise or jitter.

Impulse Response:

$$h_2[n] = \frac{1}{3}[1, 1, 1]$$

This is a **moving average filter**.

Input:

Let’s feed it the output of Stage 1:

$$x[n] = [1, 0, 1, 2, 3, -7]$$

Convolution:

$$y_2[n] = \frac{1}{3}\left[1, (1+0), (1+0+1), (0+1+2), (1+2+3), (2+3-7), (3-7), (-7)\right] \Rightarrow [0.3$$

✅ **Effect**: Smooths fluctuations, suppresses noise.

🔹 Stage 3: Matched Filter

Goal:

Detect a known signal (e.g., drone chirp: [2, 4, 2])

Impulse Response:

$$h_3[n] = [2, 4, 2] \quad \text{(flipped template)}$$

Input:

Assume signal embedded in noise:

$$x[n] = [0, 1, 2, 4, 2, 0, 0]$$

Convolution Output:

(We’ll compute only key value)

- At $n = 4$:

$$y_3[4] = 2 \times 2 + 4 \times 4 + 2 \times 2 = 4 + 16 + 4 = 24 \quad \text{(peak match)}$$

✅ **Effect**: Picks up known signal pattern → **maximum when match occurs**

🔹 Stage 4: Channel Model (Echo/Multipath)

Goal:

Simulate how signal is **distorted in air** (e.g., multipath echo)

Impulse Response:

$$h_4[n] = [1, 0, 0.5]$$

This means: direct path (1), echo after 2 units (0.5 strength)

Input:

$$x[n] = [1, 2, 0, 0]$$

Convolution:

$$\begin{aligned} y[0] &= 1 \times 1 = 1 \\ y[1] &= 2 \times 1 = 2 \\ y[2] &= 0 \times 1 + 2 \times 0 + 1 \times 0.5 = 0.5 \\ y[3] &= 0 \times 1 + 0 \times 0 + 2 \times 0.5 = 1.0 \\ y[4] &= 0 \times 1 + 0 \times 0.5 = 0 \end{aligned} \Rightarrow [1, 2, 0.5, 1.0, 0]$$

✅ **Effect**: Simulates real-world distortion due to echo or channel multipath.

✅ Summary Table

Stage	$h[n]$	Purpose	Result
1. Pre-emphasis	[1, -1]	Highlight fast changes	Edge/transition detector
2. Smoothing	$\frac{1}{3}[1, 1, 1]$	Noise reduction	Low-pass behavior
3. Matched filter	[2, 4, 2]	Detect known shape	Peak on match
4. Channel model	[1, 0, 0.5]	Simulate echo	Distorted version of input

---

Would you like to now:

- 🔁 Pick one stage and change $x[n]$ to see its effect?
- 🛠 Combine all stages into a pipeline and simulate it fully?
- 📊 Plot all of them visually using Python?

Let’s proceed step-by-step!