ANSWERS

U4.4 Outliers

Noise and filtering · University · about 25 min

What this lesson is about

Some readings are not noisy, they are wrong. The median, and gating on what you expected.

- Watch one happen
- The median filter
- Gating

Questions

6 marks in all

1. Nine readings of 60 cm and one of 400 cm are averaged. By how many centimetres does the outlier shift the mean away from 60? [1 mark]

> **Answer:** 34. The mean is (9 x 60 + 400) / 10 = 94, so one bad reading moves the answer by (400 - 60) / 10 = 34 cm.

2. A window of five holds two maximum-range readings. What does this print? [1 mark]

```
window = [61, 400, 59, 60, 400]
median = sorted(window)[len(window) // 2]
mean = sum(window) / len(window)
print(median, mean)
```

> **Answer:**
>
> ```
> 61 196.0
> ```
>
> Sorted, the window is 59, 60, 61, 400, 400, and the middle one is 61. The mean is 196, ruined by the two outliers the median ignores.

3. How many bad readings out of five can a median of five survive with its output still one of the good readings? [1 mark]

> **Answer:** 2. With two outliers at one end, the middle (third) sorted value is still a good reading. A third outlier would reach the middle.

4. A gate is rejecting about half of all readings. What does that most likely say? [1 mark]

- ☐ **A** The expectation the readings are compared with is wrong, and the gate is hiding it
- ☐ **B** The gate is working well and protecting the estimate
- ☐ **C** The sensor noise has halved
- ☐ **D** The gate should be widened to three times the median

> **Answer:** A. A gate should reject the occasional nonsense reading. Rejecting half means the model of what to expect has drifted, which is why the lesson says to count and print rejections.

5. Which of these are real costs of a median filter? [1 mark]

Tick every answer that is true.

- ☐ **A** It rounds off genuine sharp changes as well as outliers
- ☐ **B** It needs a sort every tick, which is not free on a microcontroller at 200 Hz
- ☐ **C** A single maximum-range reading ruins its output
- ☐ **D** It only works when the noise is Gaussian

> **Answer:** A, B. The median is robust to a minority of bad readings, which is its point. It pays for that by blunting real steps and by sorting the window each tick.

6. In a Kalman filter, the validation gate is set from the covariance. What does that achieve? [1 mark]

- ☐ **A** The gate widens when the filter is genuinely unsure and tightens when it is confident
- ☐ **B** The gate stays fixed so the rejection count is comparable
- ☐ **C** Every reading is accepted once the covariance has settled
- ☐ **D** The gate rejects readings that agree too closely with the estimate

> **Answer:** A. A gate tied to the uncertainty lets surprising readings through when the filter has reason to doubt itself, and blocks them when it does not.

The task: throw out the wrong ones

Drive towards the wall with a median filter and a gate running, and stop 35 cm from it, without being fooled by the robot that crosses in front. Plot `raw` and `median`, and print `thrown out:`, the number of readings your gate rejected.

```
from bugbot import *
connect()

window = []
thrown = 0
```

The hint students can ask for: Another robot shuttles across between you and the wall. While it is in the beam the sensor measures it, not the wall: two readings far too short. Stop 35 cm from the wall itself. A median of the last five ignores two bad readings completely; without it the robot believes the wall has jumped towards it and stops early. Count the readings you refuse.

A solution

```
from bugbot import *
connect()

window = []
thrown = 0
last_good = distance()
forward(40)
for i in range(70):
    raw = distance()
    window.append(raw)
    if len(window) > 5:
        window.pop(0)
    med = sorted(window)[len(window) // 2]
    if abs(raw - med) > 25:
        thrown += 1
    else:
        last_good = med
    plot("raw", raw)
    plot("median", last_good)
    if last_good < 35:
        stop()
    wait(0.1)
stop()
print("thrown out:", thrown)
```

Any program that meets the task's checks is marked correct in the simulator; this is one way, not the only way.