



U6.6 Fusing a fix

What this lesson is about

Odometry prediction, a tag as the measurement, and a gain that rises as the estimate ages.

- The rule
- The sawtooth
- R for a tag
- Gating, again

Questions

6 marks in all

1. The variance for six ticks, with a tag fix on ticks 3 and 6. What does it print? [1 mark]

```
Q, R_TAG = 0.6, 9.0
p = 4.0
for tick in range(1, 7):
    p += Q
    if tick % 3 == 0:
        k = p / (p + R_TAG)
        p = (1 - k) * p
    print(tick, round(p, 2))
```

Answer:

```
1 4.6
2 5.2
3 3.53
4 4.13
5 4.73
6 3.35
```

p climbs by 0.6 per tick to 5.8, a fix cuts it to $5.8 \times 9 / 14.8 = 3.53$, then it climbs again to 5.33 and drops to 3.35. That is the sawtooth.

2. Using the lesson's range model $R_{\text{tag}} = 4.0 + 0.002 * d * d$, what is R for a tag 100 cm away? [1 mark]

Answer: $24. 4 + 0.002 \times 100 \times 100 = 4 + 20 = 24$. At 30 cm it would be 5.8, so near tags are trusted far more.

3. The gate accepts a fix when (measured - est) squared is less than $9 * (p + R_{\text{TAG}})$. With $p = 16$ and $R_{\text{TAG}} = 9$, what is the largest innovation magnitude accepted, in cm? [1 mark]

Answer: $15. 9 \times (16 + 9) = 225$, and the square root is 15 cm, which is three sigma of the predicted innovation spread of 5 cm.

4. Why is it useful that this gate widens when the filter is unsure?

[1 mark]

- ☐ A That is exactly when a surprising fix is most likely to be genuine
- ☐ B It lets the filter reject more fixes as time passes
- ☐ C It keeps the rejection count constant
- ☐ D It stops p from growing

Answer: A. A large p means dead reckoning has drifted, so a fix far from the estimate may well be right. When p is small, the same disagreement is more likely a misread tag.

5. U3.5 took a fix by replacing the estimate with the tag's value. Which weaknesses does fusing with a gain fix?

Tick every answer that is true.

- ☐ A It threw away good dead reckoning
- ☐ B It trusted the tag completely
- ☐ C It needed the robot to stop
- ☐ D It could not use the flow sensor

Answer: A, B. Replacing discards the prediction and assumes the tag has no error. The gain weights the two by their variances instead.

6. Between fixes the robot dead reckons. What does the variance plot show?

[1 mark]

- ☐ A A steady climb, dropping sharply each time a fix arrives
- ☐ B A steady fall, jumping up at each fix
- ☐ C A flat line, since flow has no noise
- ☐ D A smooth curve settling to a constant

Answer: A. Each prediction adds Q and each correction multiplies by $(1 - k)$. The sawtooth is the picture of every navigation system.

The task: fuse a tag fix

Drive at least 80 cm towards tag 4, predicting with flow and correcting once a second while the tag is in view. Plot `estimate` and `variance`, and print `my y:`. No `position()`.

```
from bugbot import *
connect()

DT, TAG_Y = 0.1, 165.0
Q, R_TAG = 0.6, 9.0
set_cv("apriltag")
est, p = 0.0, 4.0
```

The hint students can ask for: Tag 4 is 165 cm up the mat from the start. Predict with flow and grow the variance every tick; when the tag is in view, correct with 165 minus its distance, using R for a tag fix. Watch the variance sawtooth: climbing between fixes, dropping at each one.

A solution

```
from bugbot import *
connect()

DT = 0.1
TAG_Y = 165.0
R_TAG = 9.0
Q = 0.6

set_cv("apriltag")
est, p = 0.0, 4.0
forward(65)
for i in range(90):
    est += flow()[1] * DT
    p += Q
    seen = [t for t in apriltags() if t[0] == 4]
    if seen and seen[0][3] < 130:
        measured = TAG_Y - seen[0][3]
        k = p / (p + R_TAG)
        est += k * (measured - est)
        p = (1 - k) * p
    plot("estimate", est)
    plot("variance", p)
    if est > 105:
        stop()
    wait(DT)
stop()
print("my y:", round(est, 1))
```

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