



U6.6 Fusing a fix

State estimation · University · about 35 min

Name _____

Class _____

Date _____

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))
```

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]

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]

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

5. U3.5 took a fix by replacing the estimate with the tag's value. Which weaknesses does fusing with a gain [1 mark] 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

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

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
```

Plan your program here, then type it in and press Run.



Do it on the robot

www.bugbotlab.com/learn/u6-6-fusing-a-fix/

The simulator checks it and tells you when it passes. Nothing to install, no account.

Challenges

1. Make `R_TAG` grow with range and compare the estimate.
2. Add a three sigma gate and count the fixes it refuses.
3. Print the innovation at each fix. Does it shrink as the run goes on?