



U3.5 A fix from a landmark

Odometry and drift · University · about 30 min

Name _____

Class _____

Date _____

What this lesson is about

Dead reckoning between landmarks, and a reset whenever something known comes into view.

- The oldest idea in navigation
- A fix from a tag
- Replacing and blending
- When not to trust a fix

Questions

6 marks in all

1. Tag 7 is 165 cm up the mat from the start, and the robot, facing it, reads it at 40 cm. What y does the fix give, in cm? [1 mark]

2. The estimate is $y = 120$ and three fixes in a row all say 130. The estimate is blended as in the lesson. [1 mark]
What does this print?

```
y = 120.0
for fixed in (130.0, 130.0, 130.0):
    y = 0.8 * y + 0.2 * fixed
print(round(y, 2))
```

3. Dead reckoning with a fix every so often. What does a plot of the position error against time look like? [1 mark]

- ☐ A A sawtooth: growing between fixes and dropping to nearly nothing at each one
- ☐ B A straight line that grows all run, only more slowly
- ☐ C A square root curve that levels off without any fixes
- ☐ D Flat at zero, because each fix removes the error for good

4. When is replacing the estimate with the fix, rather than blending, the right choice? [1 mark]

- ☐ A When the fix is far better than the estimate
- ☐ B Always, because a fix comes from outside the robot
- ☐ C When the fix is noisy, so that its noise is removed at once
- ☐ D When the tag is far away, since a distant tag covers more of the mat

5. Which of these are reasons in the lesson to distrust a fix?

[1 mark]

Tick every answer that is true.

- ☐ A The tag is far away
- ☐ B The tag is seen at a sharp angle
- ☐ C It is a single detection
- ☐ D The id does not match the landmark you think it is
- ☐ E The same id has been detected on two ticks in a row

6. Watching the printout, each fix makes a bigger jump in the estimate than the last. What does that tell you? [1 mark]

- ☐ A The dead reckoning error between fixes is growing, which points to a bias such as an uncalibrated gyro
- ☐ B The tag is moving
- ☐ C The fixes are getting noisier as the robot gets closer
- ☐ D The blend weight is too high

The task: a fix from a tag

Tag 7 is on the far wall, 165 cm up the mat from the start. Drive at least a metre towards it, dead reckoning as you go, take a fix when the tag is close enough to be worth having, and print `my y:` at the end.

```
from bugbot import *
connect()

set_cv("apriltag")
TAG_Y = 165.0
```

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



Do it on the robot

www.bugbotlab.com/learn/u3-5-a-fix-from-a-landmark/

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

Challenges

1. Blend the fix at 0.2 instead of replacing, and compare the error.
2. Only accept a fix when the tag is within 100 cm. Does that help or hurt?
3. Print the size of the jump at each fix. What does a growing jump tell you about your dead reckoning?

