



U1.3 Truth and belief

The robot as a system · University · about 25 min

Name _____

Class _____

Date _____

What this lesson is about

position() is the lab's overhead camera. odometry() is what the robot has, and it drifts.

- Watch belief come apart from truth
- Why dead reckoning drifts
- Turning the noise off

Questions

6 marks in all

1. Why is using position() as the input to a controller treated as cheating? [1 mark]
 - ☐ A It is the lab's overhead measurement of the robot, which a real robot does not have once it leaves the room
 - ☐ B It is less accurate than odometry()
 - ☐ C It reports the position in the body frame, not the world frame
 - ☐ D It updates too slowly to be used inside a control loop
2. A gyro reads 0.4 degrees per second too high. Integrated for 90 seconds, how many degrees of heading error does that bias alone produce? [1 mark]

3. How do the two kinds of sensor error grow when dead reckoning integrates them? [1 mark]
 - ☐ A Zero-mean noise grows with the square root of time; bias grows linearly with time
 - ☐ B Noise grows linearly with time; bias grows with the square root of time
 - ☐ C Both grow linearly with time
 - ☐ D Noise averages away completely; only bias grows
4. Driving a square, the heading error gets a little bigger at every corner. Why does that matter more than a position error of the same size? [1 mark]
 - ☐ A A heading error rotates every step taken after it, so a small error at a corner becomes a large position error down the next straight
 - ☐ B The gyro bias only acts while the robot is turning
 - ☐ C The flow sensor is noisier while turning than while driving straight
 - ☐ D odometry() restarts its count at each corner
5. Driving the same square with set_noise(0), odometry() and position() agree exactly, yet the robot still has not come back to where it started. What does that show? [1 mark]
 - ☐ A set_noise() changes the sensors, not the drive: the drive still curves, and the robot now measures the curve exactly
 - ☐ B set_noise(0) also turns off the drive's coupling, so the square is perfect
 - ☐ C odometry() stops working when the noise is off
 - ☐ D The gyro bias is still switched on

6. At the end of a run, `odometry()` gives (12.0, 118.5) and `position()` gives (15.0, 114.5). What does this print? [1 mark]

```
import math
ox, oy = 12.0, 118.5
tx, ty = 15.0, 114.5
print("error:", round(math.hypot(ox - tx, oy - ty), 1))
```

The task: how far out is it?

Drive at least 120 cm, with at least one turn in it. At the end, print how far `odometry()` is from `position()`, as `error: 4.3`.

```
from bugbot import *
connect()

forward(70, distance=50)
```

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



Do it on the robot

www.bugbotlab.com/learn/u1-3-truth-and-belief/

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

Challenges

1. Do the same run with `set_noise(0)`, `set_noise(1)` and `set_noise(3)`. Tabulate the error.
2. Drive 200 cm in a straight line, then 200 cm as four 50 cm legs with three turns. Which ends further out, and why?
3. Print the heading error separately from the position error. Which one would you rather fix?