ANSWERS

U3.3 How the error grows

Odometry and drift · University · about 30 min

What this lesson is about

Noise walks, bias marches, scale stretches, and a heading error rotates everything after it.

- Noise: a random walk
- Bias: a march
- Scale: a stretch
- Heading error: the multiplier
- The ranking

Questions

6 marks in all

1. The noise error in a dead reckoning run grows as a random walk. If the run is made twice as long, by what factor does that error grow? Give two decimal places. [1 mark]

> **Answer:** 1.41 (accept within 0.01). A random walk grows with the square root of the number of steps, so doubling the run multiplies it by the square root of 2, about 1.41.

2. Which of these errors grow in direct proportion to the time or distance driven, rather than with its square root? [1 mark]

Tick every answer that is true.

- ☐ **A** Heading error from a gyro bias
- ☐ **B** Position error from a flow scale error
- ☐ **C** Error from zero-mean noise on flow()

> **Answer:** A, B. A bias and a scale error lean the same way every step, so they add up in a straight line. Zero-mean noise partly cancels and grows only with the square root.

3. Gyro rates in deg/s are read while the robot stands perfectly still. What does this print? [1 mark]

```
rates = [0.3, 0.5, 0.2, 0.4, 0.1, 0.3]
mean = sum(rates) / len(rates)
spread = (sum((r - mean) ** 2 for r in rates) / len(rates)) ** 0.5
print("bias:", round(mean, 2))
print("noise:", round(spread, 2))
print("after 60 s:", round(mean * 60, 1))
```

> **Answer:**
>
> ```
> bias: 0.3
> noise: 0.13
> after 60 s: 18.0
> ```
>
> The mean is 1.8 / 6 = 0.3 deg/s, which is the bias. The squared deviations sum to 0.1, so the spread is the square root of 0.1 / 6, about 0.13. The bias alone is 0.3 × 60 = 18 degrees after a minute.

4. Using the lesson's rule that driving d with a heading error of e radians puts you about d × e to one side, this works out the sideways error over 10 metres. What does it print? [1 mark]

```
import math
for deg in (1, 3, 10):
    print(deg, round(1000 * math.radians(deg), 1))
```

Answer:

```
1 17.5
3 52.4
10 174.5
```

One degree is 0.01745 radians, so 1000 cm × 0.01745 = 17.5 cm. Three degrees gives 52.4 cm and ten gives 174.5 cm, from an error that costs nothing while the robot stands still.

5. The flow sensor reads 4 percent high. How many centimetres too far does dead reckoning put the robot after a 10 m straight run? [1 mark]

Answer: 40. A scale error is proportional to distance: 4 percent of 1000 cm is 40 cm.

6. For this sort of ground robot over a run of a minute or two, which error source does the lesson rank as the most damaging? [1 mark]

- ☐ **A** Heading bias
- ☐ **B** Scale error on the translation
- ☐ **C** Slip of the drive on the mat
- ☐ **D** Noise on the readings

Answer: A. Heading bias grows linearly and every degree of it is multiplied by the distance driven afterwards. Noise, at the other end of the ranking, mostly averages away.

The task: measure the gyro bias

Without moving the robot at all, print this gyro's bias as `bias: 0.31`, in degrees per second.

```
from bugbot import *
connect()

rates = []
```

The hint students can ask for: Stand still and read imu()[1] many times. The average of enough readings is the bias; the spread around it is the noise. Sixty readings is plenty and takes six seconds.

A solution

```
from bugbot import *
connect()

rates = []
for i in range(60):
    rates.append(imu()[1])
    wait(0.1)
print("bias:", round(sum(rates) / len(rates), 2))
```

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