



2.5 Drift and correction

Sensing · Robot club · about 20 min

What this lesson is about

Why the robot wanders, and the first closed loop.

- Watch the drift
- Open loop and closed loop
- The simplest correction
- Correcting the heading too
- Steering while moving

Questions

7 marks in all

1. Which of these programs is closed loop?

[1 mark]

- ☐ A One that checks position() after each leg and slides back to the line
- ☐ B forward(70), wait(3), stop()
- ☐ C forward(70, distance=64) on its own
- ☐ D A for loop that drives the same square every time

Answer: A. Closed loop means the sensors feed back into what the motors do. Open loop acts and hopes.

2. Why does the robot slide sideways when it drives forward? Choose all that apply.

[1 mark]

Tick every answer that is true.

- ☐ A The four feet do not push exactly equally
- ☐ B The mat is not perfectly even
- ☐ C Tiny errors add up over a long drive
- ☐ D There is a bug in forward()
- ☐ E position() resets itself as it drives

Answer: A, B, C. Every BugBot drifts a little. It is the drive and the mat, not your program.

3. After a 10 cm leg, position() says x is 2.5. What does the correcting program do?

[1 mark]

```
x, y = position()
if x > 1:
    left(40, distance=x)
elif x < -1:
    right(40, distance=-x)
```

- ☐ A Slides left 2.5 cm
- ☐ B Slides right 2.5 cm
- ☐ C Nothing, because 2.5 is too small
- ☐ D Turns left 2.5 degrees

Answer: A. x is positive, so the robot is to the right of the line. left(40, distance=x) slides it back by that much.

4. What does this program print?

[1 mark]

```
error = 355
if error > 180:
    error = error - 360
print(error)
```

Answer:

-5

A heading of 355 is really 5 degrees to the left of 0, so the error should be -5, not 355.

5. In `drive(70, -x * 15, 0)`, `x` is 2. What is the sideways value? Give a whole number.

[1 mark]

Answer: -30. Minus 2 times 15 is -30. Negative sideways is left, which pushes the robot back towards the line.

6. You change `-x * 15` to `-x * 40`. What do you expect?

[1 mark]

- ☐ A It settles closer to the line
- ☐ B It drifts further off the line
- ☐ C It drives faster
- ☐ D Nothing changes

Answer: A. A bigger gain pushes back harder for the same error, so the robot settles 0.7 cm off the line instead of 1.8. Push the gain much higher and it starts to overshoot one way, then the other.

7. Why is it worth correcting the heading as well as `x`?

[1 mark]

- ☐ A A wrong heading turns into a sideways error on the next leg
- ☐ B `heading()` is the only thing that drifts
- ☐ C Correcting `x` resets the heading
- ☐ D It is not, `x` is all that matters

Answer: A. If the robot faces a few degrees off, every forward move carries it a little sideways.

The task: straight and true

Drive about 64 cm and finish inside the small square, still facing 0 within 6 degrees. Correct as you go.

```
# the two lines every program starts with: the commands, then the robot
from bugbot import *
connect()

# drive forward at 70 for 64 cm, then stop
forward(70, distance=64)
```

The hint students can ask for: This robot leaks sideways badly. Drive about 64 cm and finish inside the small square, still facing 0. Check position() and heading() as you go and correct with small strafes and turns.

A solution

```
from bugbot import *
connect()
for step in range(6):
    forward(70, distance=10)
    x, y = position()
    if x > 1:
        left(40, distance=x)
    elif x < -1:
        right(40, distance=-x)
    err = heading()
    if err > 180:
        err -= 360
    if abs(err) > 2:
        turn_left(30, angle=err)
    forward(40, distance=4)
```

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