ANSWERS

3.3 Tuning the gain

BugBotLab

Control · Robot club · about 20 min

What this lesson is about

Sluggish, good, oscillating. Loop speed and driving speed.

- Too little
- Too much
- Why the loop speed matters
- Measure the sweep
- Speed and gain together

Questions

7 marks in all

1. With a gain of 1 on the curving robot, what happens? [1 mark]

- ☐ **A** The correction is too weak: the error grows and the robot ends well off the line
- ☐ **B** The robot snakes from side to side
- ☐ **C** The robot holds the heading perfectly
- ☐ **D** The robot spins on the spot

> **Answer:** A. A controller that is too weak is called sluggish. It never even crosses the target.

2. With a gain of 40, the robot snakes. What is that called? [1 mark]

- ☐ **A** Oscillation
- ☐ **B** Coasting
- ☐ **C** Drift
- ☐ **D** The dead band

> **Answer:** A. A tiny error becomes a huge rotation, which overshoots, which becomes a huge rotation the other way.

3. Rotation is capped at 100. With a gain of 40, above what heading error is the controller just bang-bang again? Give the number of degrees. [1 mark]

> **Answer:** 2.5. 40 times 2.5 is 100. Any error bigger than that asks for more than 100, so it gets the full 100 every time.

4. What does this program print? [1 mark]

```
errors = [5, 2, -1, -3, 1, 4, -2]
worst, flips, last = 0, 0, 0
for error in errors:
    worst = max(worst, abs(error))
    if error * last < 0:
        flips += 1
    last = error
print("worst", worst, "and", flips, "sign changes")
```

Answer:

```
worst 5 and 3 sign changes
```

A sign change is when this error and the last one have opposite signs, so they multiply to a negative. That happens at -1, 1 and -2. The biggest size is 5.

5. Why is a high gain with a slow loop the worst combination? [1 mark]

- ☐ **A** A big correction keeps running for too long before the next look
- ☐ **B** A slow loop makes the gain smaller
- ☐ **C** The robot cannot drive while waiting
- ☐ **D** The sensor stops working

Answer: A. Between looks the robot keeps doing what you last said. A big push for 0.3 seconds is a big overshoot.

6. In a sweep of gains, where is the good gain? [1 mark]

- ☐ **A** Just before the sign changes start climbing
- ☐ **B** The highest gain you can type
- ☐ **C** The lowest gain, 1
- ☐ **D** Wherever the worst error is biggest

Answer: A. Raising the gain shrinks the worst error, until the robot starts crossing the target back and forth.

7. A gain works well at speed 50. What might happen at speed 100? [1 mark]

- ☐ **A** It may be sluggish, because the fault acts faster
- ☐ **B** It will oscillate more, because speed makes the gain bigger
- ☐ **C** Nothing, the gain does not care about speed
- ☐ **D** The robot will drive backwards

Answer: A. This robot turns about twice as hard at 100. The same gain has twice as much fault to fight.

The task: hold the heading, fast

The zone is narrower and you have six seconds. The starter's gain of 40 wobbles and wastes time, and its speed is too low. Tune both.

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

def wrapped(h):
    return (h + 180) % 360 - 180

while position()[1] < 72:
    error = wrapped(0 - heading())
    # forward, sideways, rotation: -100 to 100 each, until the next command
    drive(60, 0, error * 40)
    # pause 0.05 s (the robot keeps doing what it was told)
    wait(0.05)
# all motors off
stop()
```

The hint students can ask for: Same robot, but the zone is narrower and you have 6 seconds. The starter's gain is far too high: it wobbles and wastes time. Tune the gain and the speed.

A solution

```
from bugbot import *
connect()
def wrapped(h):
    return (h + 180) % 360 - 180
while position()[1] < 72:
    error = wrapped(heading())
    drive(100, 0, -error * 4)
    wait(0.05)
stop()
```

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