



3.2 Proportional control

Control · Robot club · about 15 min

Name _____

Class _____

Date _____

What this lesson is about

Correct in proportion to the error. Gain, sign, feedback.

- The robot that curves
- Steer by the error
- The sign
- Why it is called feedback

Questions

6 marks in all

1. What does a proportional controller do?

[1 mark]

- ☐ A Pushes in proportion to the error: hard when far off, gently when close
- ☐ B Goes at one fixed speed until the error is zero
- ☐ C Turns a set amount every loop
- ☐ D Waits until the error is big before doing anything

2. The gain is 4 and the heading error is 7.5 degrees. What rotation does `drive(70, 0, error * 4)` ask for? [1 mark]

3. What does this program print?

[1 mark]

```
def wrapped(h):  
    return (h + 180) % 360 - 180  
  
for heading in [10, 350]:  
    error = wrapped(0 - heading)  
    print(heading, error * 4)
```

4. You write `drive(70, 0, -error * 4)` by mistake. What happens?

[1 mark]

- ☐ A The controller runs away and the error grows
- ☐ B The robot holds the heading more gently
- ☐ C Nothing, the sign does not matter
- ☐ D The robot drives backwards

5. Why is it called a feedback loop?

[1 mark]

- ☐ A The heading comes out of the robot and goes back to the motors as a correction
- ☐ B The robot prints its heading every loop
- ☐ C It uses a `for` loop
- ☐ D The robot drives in a circle

6. Why does the P controller not need bursts and settling like the crude fix?

[1 mark]

- ☐ A Small errors give small rotations, so the correction fades away with no overshoot
- ☐ B It always drives at a low speed
- ☐ C It stops after every loop
- ☐ D It only runs once

The task: hold the heading

Drive into the green zone and finish still facing 0, within 6 degrees, on the curving robot.

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

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

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



Do it on the robot

www.bugbotlab.com/learn/3-2-proportional-control/

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

Challenges

1. Print the error every tick and watch it shrink.
2. Hold a heading of 20 instead of 0, so the robot drives a straight diagonal.
3. Replace `wait(0.1)` with `wait(0.5)`. What happens, and why?