ANSWERS

U5.2 Proportional control

Feedback control · University · about 30 min

What this lesson is about

One gain, the whole trade-off, and the steady state error it cannot remove.

- What the gain means
- Turning it up
- Where the overshoot is
- The steady state error
- Choosing Kp by hand

Questions

6 marks in all

1. A proportional controller has Kp = 1.5 percent per cm. What command, in percent, does an error of 12 cm produce? [1 mark]

> **Answer:** 18. command = Kp x error = 1.5 x 12 = 18 percent.

2. The error is in centimetres and the command in percent. What are the units of Kp? [1 mark]

> **Answer:** percent per centimetre. Kp converts centimetres of error into percent of command, so it is percent per centimetre. That is why a gain does not carry over to a different robot.

3. This drive does not move below 15 percent. With Kp = 2, what is the largest error in cm that proportional control alone leaves uncorrected? [1 mark]

> **Answer:** 7.5 (accept within 0.01). The command reaches 15 percent at 15 / 2 = 7.5 cm. Any smaller error asks for a push the motors cannot deliver, so it stays.

4. In the lesson's turning cell at Kp = 4, the error trace falls fast, crosses zero, comes back and crosses again before settling. What does that say? [1 mark]

- ☐ **A** The gain is too high: fast but underdamped
- ☐ **B** The gain is too low
- ☐ **C** There is steady state error
- ☐ **D** The sensor is biased

> **Answer:** A. Overshoot and ringing are the signature of too much proportional gain for how fast the robot turns and how often the loop looks. At Kp = 1 the same turn arrives without swinging.

5. Why does a low-gain proportional controller stop short of the target and stay there? [1 mark]

- ☐ **A** It needs an error to produce a command, and the command for a small error is below the drive's dead band
- ☐ **B** The integral has wound up
- ☐ **C** The sensor filter delays the last few centimetres
- ☐ **D** The sign of the error is wrong

> **Answer:** A. Proportional control only pushes in proportion to the error. When that push falls below 15 percent nothing moves, so the error never shrinks further.

6. Put the lesson's hand-tuning recipe for Kp in order. [1 mark]

Number the lines 1 to 3 to put them in the right order.

___ `Halve that value`

___ `Start low, around 1`

___ `Double it until the robot overshoots or wobbles`

> **Answer:**
>
> ```
> Start low, around 1
> Double it until the robot overshoots or wobbles
> Halve that value
> ```
>
> Find the gain where trouble starts, then back off by half.

The task: find a gain that works

Settle 30 cm from the wall, within 5 cm, and stay there from ten seconds onwards. Print the `kp:` you chose.

```
from bugbot import *
connect()

KP = 1.0
TARGET = 30.0
```

The hint students can ask for: Too little gain and it stops short: once the command falls under 15 percent the drive does nothing. More gain gets it closer; far too much and it shuffles back and forth on the sensor noise. Start at 1, double until it misbehaves, then come back down.

A solution

```
from bugbot import *
connect()

KP = 2.5
TARGET = 30.0
for tick in range(350):
    error = distance() - TARGET
    plot("error", error)
    drive(max(-60.0, min(60.0, KP * error)), 0, 0)
    wait(0.1)
stop()
print("kp:", KP)
```

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