



3.4 Controlling speed

Control · Robot club · about 15 min

Name _____

Class _____

Date _____

What this lesson is about

Speed from the remaining distance, the dead band, clamping.

- The bang-bang park
- Proportional speed
- The dead band
- The pattern

Questions

6 marks in all

1. The bang-bang park drives at 60 until `distance()` is 22 or less, then stops. Where does it end? [1 mark]
☐ A Closer than 22 cm, because it coasts
☐ B Exactly 22 cm
☐ C Further than 22 cm
☐ D Touching the wall every time
2. Target 22, gain 4, and `distance()` reads 32. What speed does `speed = (distance() - target) * 4` ask for? [1 mark]

3. The plain proportional park stalls a little short of the target. Why? [1 mark]
☐ A Near the target it asks for speeds below about 15, which do not move the robot
☐ B The gain is too high
☐ C The sensor cannot see closer than 25 cm
☐ D The loop runs out of ticks too early
4. What does this program print? [1 mark]

```
for value in [4, 30, 250]:  
    print(max(16, min(60, value)))
```


5. In the clamped park, the robot overshoots and `distance()` reads 18. What does it do?

[1 mark]

```
error = distance() - target
speed = max(16, min(60, abs(error) * 4))
if error > 0:
    forward(speed)
else:
    backward(speed)
```

- ☐ A Backs up slowly, at speed 16
- ☐ B Stops, because it is past the target
- ☐ C Drives forward at speed 16
- ☐ D Backs up at speed 60

6. Put the three lines of every P controller in order.

[1 mark]

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

```
___ output = clamp(error * gain)
___ apply(output)
___ error = target - measured
```

The task: park at twenty

Stop with the front of the robot 20 to 24 cm from the wall, the green band, without touching the wall.

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

# as long as the thing ahead is further than 22 cm
while distance() > 22:
    # drive forward at 60 (keeps going until the next command)
    forward(60)
    # pause 0.1 s (the robot keeps doing what it was told)
    wait(0.1)
# all motors off
stop()
```

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



Do it on the robot

www.bugbotlab.com/learn/3-4-controlling-speed/

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

Challenges

1. Park at 40 cm, then at 10 cm, with the same loop and a different `target` .
2. Print the speed every tick and watch it fall as the robot approaches.
3. Start the robot **closer** than the target (edit the loop to drive there first) and check it backs up.