



U1.2 A command is a request

The robot as a system · University · about 25 min

Name _____

Class _____

Date _____

What this lesson is about

Open loop on real hardware: gain that is not 1, a dead band, a lag, and a robot that curves.

- What actually happens
- Measuring instead of assuming
- Why this matters

Questions

6 marks in all

- What does `forward(60)` actually ask for? [1 mark]
 - ☐ A The motors driven at 60 percent
 - ☐ B A speed of 60 cm/s
 - ☐ C Exactly 60 percent of the catalogue top speed of 20 cm/s, which is 12 cm/s
 - ☐ D A move of 60 cm
- Asked to drive 60 cm forward, the robot finishes to one side of where it started and a few degrees off heading. Which of the four effects in the lesson explains that? [1 mark]
 - ☐ A Coupling
 - ☐ B Dead band
 - ☐ C Lag
 - ☐ D Gain
- A robot's measured forward speed is 7 cm/s at command 40 and 17 cm/s at command 80. Using the straight line through those two points, what command gives 12 cm/s? [1 mark]
- For the same line (7 cm/s at command 40, 17 cm/s at command 80), at what command does the line predict zero speed? [1 mark]
- The measurement program waits 1.0 s after each `forward()` before it starts averaging `flow()`. Why? [1 mark]
 - ☐ A The speed rises over about a quarter of a second, so readings taken during the rise would underestimate the settled speed
 - ☐ B `flow()` only produces a new reading once a second
 - ☐ C The deadman needs a second to reset before readings are valid
 - ☐ D It lets the battery voltage recover after the previous step
- Which description of open loop control matches the lesson? [1 mark]
 - ☐ A Acting on the command and hoping, which only works on a stiff, repeatable, unloaded machine
 - ☐ B Measuring the result and correcting as you go
 - ☐ C Keeping an estimate of the state rather than a commanded position
 - ☐ D Running the loop without any `wait()` so it goes as fast as possible

The task: how fast is this robot?

Measure this robot's top forward speed in centimetres per second and print it as `top speed: 18.4`.
Measure it, do not look it up.

```
from bugbot import *  
connect()  
  
forward(100)  
wait(1.0)
```

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



Do it on the robot

www.bugbotlab.com/learn/u1-2-commands-are-requests/

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

Challenges

1. Measure the sideways top speed and the top turn rate the same way.
2. Find the dead band: the smallest command that moves the robot at all.
3. Fit a straight line to the table above (speed against command) and use it to predict the command for 12 cm/s. Test the prediction.