WORKSHEET

U10.7 Project: drive the route

BugBotLab

Following a trajectory · University · about 50 min

Name _____ Class _____ Date _____

What this lesson is about

A full route through waypoints, marked on how closely it was followed and how long it took.

- The route
- The structure
- Slowing down for what matters
- Where to look when it fails
- What comes next

Questions

6 marks in all

1. What does this print for the project's route? [1 mark]

```
import math
PATH = [(30.0, 30.0), (30.0, 150.0), (120.0, 150.0), (120.0, 60.0), (170.0, 60.0)]
cum = [0.0]
for (ax, ay), (bx, by) in zip(PATH, PATH[1:]):
    cum.append(cum[-1] + math.hypot(bx - ax, by - ay))
print(cum)
print(round(cum[-1] / 13.0, 1))
```

2. The speed rule is cruise x max(0.45, 1 - off / 14) with a cruise of 13 cm/s. What speed does it give when the robot is 10 cm off the path, in cm/s to 2 decimal places? [1 mark]

3. Why is slowing down when off the path self correcting? [1 mark]

- ☐ **A** Drifting wide costs speed, which gives the correction time to work, which brings the speed back
- ☐ **B** Lower speed shortens the look-ahead distance
- ☐ **C** The dead band disappears at low speed
- ☐ **D** Slowing down removes the drive lag

4. `off path` is small on the straights but spikes at every corner, and the robot clips the crate. Where should you look? [1 mark]

- ☐ **A** The look-ahead cutting the corners: reduce L, slow down for the bend, or both
- ☐ **B** The cross-track sign convention
- ☐ **C** The cruise speed is too low to finish in time
- ☐ **D** The path itself goes through the crate

5. A follower that points its velocity at the look-ahead point stays within 1.3 cm of this route at L = 3. A classmate's robot turns to face the look-ahead point and drives forwards only, and it weaves along the straights. What is the likely cause? [1 mark]

- ☐ **A** Steering through the heading with a look-ahead too small for the speed
- ☐ **B** The look-ahead is too large
- ☐ **C** The corner speed rule is too cautious
- ☐ **D** The route is longer than 350 cm

6. The corner rule reduces speed in proportion to the angle between the path direction at the projection and at the look-ahead point. What idea is that? [1 mark]

- ☐ **A** The preview idea of pure pursuit, applied to the throttle instead of the steering
- ☐ **B** Feedforward of the reference speed
- ☐ **C** A dead band inverse on the speed
- ☐ **D** Time scaling from the tracking error

The task: drive the route

Follow the tape from (30, 30) round to the green finish, within 10 cm of it for at least 85 percent of the run, without touching the crate, inside 90 seconds. Plot `off path` and `speed`, and print `drove:`, how far the robot actually travelled.

```
from bugbot import *
import math
connect()

DT = 0.1
START = (30.0, 30.0)
PATH = [(30.0, 30.0), (30.0, 150.0), (120.0, 150.0), (120.0, 60.0), (170.0, 60.0)]
CRUISE, LOOK = 13.0, 16.0
```

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

Do it on the robot

www.bugbotlab.com/learn/u10-7-project-drive-the-route/

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

Challenges

1. Round the corners in the path itself with short arcs, and follow that instead. Does the run get faster, tighter, or both?

2. Run it with a fixed speed and no corner rule, and compare the worst `off path` value.
3. Put the route on a clock as in U10.6, and report how many seconds behind its own plan the robot finished.