



U10.3 Cross-track error

Following a trajectory · University · about 35 min

Name _____

Class _____

Date _____

What this lesson is about

Splitting the error into along and across the path, and a controller for the half that matters.

- The path frame
- Controlling it
- Choosing the gain
- The dead band, again

Questions

7 marks in all

1. A leg runs from (0, 0) to (30, 40) and the robot is at (10, 20). What does this print? [1 mark]

```
import math
ax, ay, bx, by = 0.0, 0.0, 30.0, 40.0
x, y = 10.0, 20.0
length = math.hypot(bx - ax, by - ay)
ux, uy = (bx - ax) / length, (by - ay) / length
nx, ny = -uy, ux
along = (x - ax) * ux + (y - ay) * uy
cross = (x - ax) * nx + (y - ay) * ny
print(round(along, 2), round(cross, 2))
```

2. The line runs from (60, 30) to (60, 170) and the robot, facing +y, is at (70, 50). With correction = clamp(-0.9 x cross, 12) along the left normal, and drive()'s sideways term positive to the robot's right at 15 cm/s for 100, what sideways term should be sent? [1 mark]
- ☐ A -60, because the correction of +9 cm/s is along the left normal (-1, 0), which is the robot's left
- ☐ B +60, the correction of +9 cm/s converted to percent and sent straight in
- ☐ C +9, the correction in cm/s
- ☐ D -80, the clamped maximum
3. The holonomic cross-track loop has gain $k = 0.9$ per second and the robot moves along the path at 11 cm/s. Over how many centimetres of travel is one time constant of the correction spread? Give 1 decimal place. [1 mark]
4. The robot sits 1 cm off the path with $k = 0.9$. What percentage command does the correction ask for, given 100 percent is 15 cm/s sideways? [1 mark]

5. The holonomic cross-track loop is first order and cannot oscillate on its own. Why can it still wobble in practice? [1 mark]

- ☐ A Delays: the 0.25 s drive lag, the loop period and the dead band
- ☐ B Adding a sideways velocity makes the loop second order
- ☐ C The cross-track error is signed
- ☐ D The along-track error feeds into the cross-track error

6. What happens as the cross-track gain k is raised well above 0.9? [1 mark]

Tick every answer that is true.

- ☐ A The correction saturates at the available sideways speed, so the response stops getting faster
- ☐ B The phase lost to the 0.25 s lag starts to matter and the approach becomes a wobble
- ☐ C The loop becomes second order, like a car's steering
- ☐ D The time constant $1/k$ gets longer

7. Why does the Stanley controller for a car divide the cross-track term by the speed? [1 mark]

- ☐ A At higher speed the geometry closes a given error sooner, so the same error needs a gentler steering correction
- ☐ B Faster cars have larger cross-track errors, so the term needs shrinking
- ☐ C It converts the error from centimetres to degrees
- ☐ D It stops the steering saturating at low speed

The task: drive the cross-track error to zero

The taped line runs from (60, 30) to (60, 170). The robot starts at (85, 30), which is 25 cm to the right of it. Get onto the line, drive along it to the far end, plot `cross track` the whole way, and stop near (60, 163).

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

DT = 0.1
START = (85.0, 30.0)
AX, AY, BX, BY = 60.0, 30.0, 60.0, 170.0
```

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



Do it on the robot

www.bugbotlab.com/learn/u10-3-cross-track-error/

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

Challenges

1. Try `k` at 0.2 and at 4.0 and describe both. Which one is the lag showing?
2. Plot the along-track value as well and check it climbs steadily while the cross-track collapses.
3. Remove the dead band inverse. What is the smallest cross-track error the robot can actually correct?