



U7.7 Project: the kidnapped robot

Localisation · University · about 50 min

Name _____

Class _____

Date _____

What this lesson is about

No idea where it starts. Work it out, then drive somewhere on purpose.

- Two phases
- What to watch on the chart
- The structure
- Readings while turning
- What comes next

Questions

7 marks in all

1. What does this program print?

[1 mark]

```
import math
SIGMA = 3.5
particles = [40.0, 42.0, 44.0, 150.0, 152.0]
measured = 157.0
w = [math.exp(-(((200 - y) - measured) ** 2) / (2 * SIGMA * SIGMA)) + 1e-12 for y in
particles]
total = sum(w)
weights = [v / total for v in w]
print(round(sum(v * y for v, y in zip(weights, particles)), 1))
```

2. The depth sensor faces the far wall, so its readings pin down y . The robot turns ninety degrees to the left. What does a cloud weighed against the new readings pin down?

[1 mark]

- ☐ A x
- ☐ B y again, more precisely
- ☐ C The heading
- ☐ D Both x and y at once

3. In phase one the robot faces the far wall, and the cloud finds y but stays a stripe across the mat. What cuts the stripe down to a spot?

[1 mark]

- ☐ A A reading from a different direction, such as the left wall once a ninety degree turn has finished
- ☐ B Shuffling sideways and reading the far wall again
- ☐ C Averaging more readings of the far wall
- ☐ D Using a smaller sigma

4. A project filter weighs its x cloud against every reading, including those taken during the ninety degree turn towards the left wall. It ends 58 cm from the robot with a spread of 5 cm. Why? [1 mark]
- ☐ A The readings during the turn were at a slant, so the cloud collapsed where they seemed to point, and once no guess was near the right x it could not come back
 - ☐ B The turn made the odometry heading drift by 2 to 3 degrees
 - ☐ C Five cm of spread is too much jitter
 - ☐ D The left wall is further away than the far wall
5. During the run the estimate suddenly jumps by 60 cm. What has most likely happened? [1 mark]
- ☐ A The cloud had two clusters and has just resampled onto one of them
 - ☐ B The motion noise is too large
 - ☐ C The effective sample size has risen to N
 - ☐ D The robot has driven into a wall
6. In phase two the spread starts climbing steadily. What does that mean? [1 mark]
- ☐ A The measurements have stopped being informative and the filter is effectively dead reckoning
 - ☐ B The filter is converging on the true position
 - ☐ C Too many particles are being injected
 - ☐ D The resampling threshold is too low
7. Which defence specifically lets the filter recover if the robot is picked up and moved mid-run? [1 mark]
- ☐ A Injecting a small percentage of particles scattered over the whole mat every tick
 - ☐ B Adding jitter to each resampled copy
 - ☐ C Resampling only when neff is below $N/2$
 - ☐ D Using a larger number of particles at the start

The task: the kidnapped robot

Work out where the robot woke up, print it as `found:`, and then drive into the green corner. `position()` is not allowed anywhere.

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

DT, N, SIGMA = 0.1, 400, 3.5
particles = [random.uniform(0, 200) for i in range(N)]
```

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



Do it on the robot

www.bugbotlab.com/learn/u7-7-project-the-kidnapped-robot/

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

Challenges

1. Shuffle 20 cm sideways during phase one, still facing the far wall, and keep weighing. Does the cloud collapse any faster, or learn anything about `x`? Why not?
2. Inject 2 percent random particles every tick, and half way through the run, pick the robot up in your head: set the true position somewhere else by driving it there with the filter off. Does it recover?
3. Track heading as well as position in each particle. What breaks, and what does it cost?