



U9.1 The configuration space

Planning · University · about 30 min

Name _____

Class _____

Date _____

What this lesson is about

Grow the obstacles by the robot's radius and the robot becomes a point, which is the whole reason planning is tractable.

- Grow the obstacles instead
- How much to grow by
- Seeing it
- What the free space looks like

Questions

7 marks in all

1. A robot has a 10 cm square chassis and can turn freely. By how much, in cm to two decimal places, must the obstacles be grown so the robot is safe as a point at every heading? [1 mark]

2. The simulator models the 7 cm square BugBot as a 3.5 cm circle. Why is that not safe on a real robot? [1 mark]
 - ☐ A The circle fits inside the square, so a corner of the chassis can reach an obstacle the circle says is clear
 - ☐ B The circle is too large, so it closes gaps the robot could fit through
 - ☐ C A circle cannot represent the heading, so the plan cannot turn
 - ☐ D 3.5 cm is smaller than one grid cell
3. What is the price of planning with the 4.95 cm circle around the square chassis? [1 mark]
 - ☐ A It occasionally refuses a gap the robot could have squeezed through straight on
 - ☐ B It occasionally plans a route that clips an obstacle
 - ☐ C It makes the configuration space four dimensional
 - ☐ D It makes planning time grow with the heading resolution
4. For a holonomic robot with a circular footprint, why does the heading drop out of the configuration space? [1 mark]
 - ☐ A A circle occupies the same area whichever way it is facing, so collisions do not depend on heading
 - ☐ B A holonomic robot never changes its heading
 - ☐ C The heading is always measured separately by the IMU
 - ☐ D Planners cannot handle more than two dimensions

5. What does this program print?

[1 mark]

```
BLOCK = (80.0, 90.0, 40.0, 16.0)
MARGIN = 10.0
bx, by, bw, bh = BLOCK
print(bw + 2 * MARGIN)
free = 0
for x in range(0, 200, 10):
    if not (bx - MARGIN <= x <= bx + bw + MARGIN):
        free += 1
print(free)
```

6. The start and the goal lie in different connected components of the free space. What will a correct planner do?

[1 mark]

- ☐ A Report that there is no route
- ☐ B Find a route through the narrowest passage
- ☐ C Return the route with the fewest collisions
- ☐ D Keep searching until it finds a route

7. A planner mysteriously fails on a map where a route looks possible by eye. What does the page say to look for first?

[1 mark]

- ☐ A A narrow passage only slightly wider than the inflated robot
- ☐ B A heuristic that underestimates
- ☐ C Too few obstacles in the map
- ☐ D A cell size that is too small

The task: grow the obstacle, shrink the robot

Standing still, print `inflated:` , the width of the block in centimetres after growing it by the margin on both sides, and `free lanes:` , how many of the twenty lanes at `x = 0, 10, 20 ... 190` a point robot could drive straight up from `y = 30` to `y = 170` without entering the grown block.

```
from bugbot import *
connect()

BLOCK = (80.0, 90.0, 40.0, 16.0)      # x, y, width, height on the mat
MARGIN = 6.0                          # the 4.95 cm around the square, plus a little
```

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



Do it on the robot

www.bugbotlab.com/learn/u9-1-configuration-space/

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

Challenges

1. Work out the smallest margin at which one more lane becomes blocked, and the largest at which one more becomes free.
2. The block is 16 cm deep. How wide would a gap have to be for the inflated robot to fit through it with 2 cm to spare either side?
3. The BugBot is square, not round. Describe a gap it could drive through that the circular approximation refuses.