



U2.7 Project: a shape in the world

Kinematics and frames · University · about 40 min

Name _____

Class _____

Date _____

What this lesson is about

Drive a square in world coordinates while the robot spins, which only a holonomic drive can do.

- The structure
- What comes next

Questions

6 marks in all

1. Put the structure of the spinning square in order.

[1 mark]

Number the lines 1 to 6 to put them in the right order.

- ___ apply the inverse kinematics with the heading you have now
- ___ set the speed in proportion to the gap, capped
- ___ add the rotation term, all the way round
- ___ for each corner of the square:
- ___ until close to that corner:
- ___ work out the vector to the corner, in the world

2. world_drive is asked for 10 cm/s due east at three headings the spinning robot passes through. With $V_MAX = 20$ and $V_LAT = 15$, what does this print (heading, fwd, lat)?

[1 mark]

```
import math
V_MAX, V_LAT = 20.0, 15.0
for h_deg in (0, 90, 180):
    a = math.radians(h_deg)
    wx, wy = 10.0, 0.0
    vx = wx * math.cos(a) - wy * math.sin(a)
    vy = wx * math.sin(a) + wy * math.cos(a)
    print(h_deg, round(100 * vy / V_MAX), round(100 * vx / V_LAT))
```

3. The robot spins as it should, but instead of heading east its path curls round in circles. What is the likely bug?

[1 mark]

- ☐ A The heading is read once at the top instead of inside the loop on every tick
- ☐ B The spin rate is too fast for the motors
- ☐ C The rotation matrix has the wrong sign
- ☐ D V_MAX and V_LAT are the wrong way round

4. Along one side of the spinning square, which of these change from tick to tick?

[1 mark]

Tick every answer that is true.

- ☐ A The heading
- ☐ B The body frame command sent to drive()
- ☐ C The world direction of travel being asked for
- ☐ D The coordinates of the corner being aimed at

5. world_drive(10, 0, 25) is called every 0.1 s for 40 ticks. Ideally, how far east does the robot travel, in cm?[1 mark]

6. The whole square is driven with odometry() instead of position(). Why is the fourth corner the least square? [1 mark]

- ☐ A Dead reckoning error accumulates, and heading error rotates every step after it, so the estimate is worst at the end
- ☐ B The inverse kinematics is less accurate for the last side
- ☐ C odometry() resets at every corner
- ☐ D The fourth side is driven westwards, where the sideways full scale is smaller

The task: a square while spinning

Visit the four corner zones in order, east then north then west then home, with a rotation term running all the way round.

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

V_MAX, V_LAT = 20.0, 15.0
# the corners, in cm from where the robot started
CORNERS = [(40, 0), (40, 60), (-20, 60), (-20, 0)]
```

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



Do it on the robot

www.bugbotlab.com/learn/u2-7-project-a-shape/

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

Challenges

1. Plot the world path by printing x and y each tick, and check the corners are square.
2. Make the robot face the corner it is heading for, instead of spinning. Which is easier to write, and which is easier to watch?
3. Do the whole square with `odometry()` instead of `position()` . How square is it by the fourth corner?