WORKSHEET

U2.8 Driving like a car

Kinematics and frames · University · about 35 min

Name _____ Class _____ Date _____

What this lesson is about

The unicycle and the car, the velocity constraint that makes them non-holonomic, the bracket that gets a car sideways anyway, and parallel parking.

- Three robots on one mat
- The constraint is on velocity, not on position
- Getting sideways without going sideways
- Parallel parking
- Why the BugBot does not bother

The task: parallel park

The bay is to the robot's right, between two parked cars. All the distances are from the robot's starting point, facing along `+y` (heading 0). - **The bay** is 18 cm wide and 47 cm long. Across, it runs from 13 cm to 31 cm to your right; along, from 10 cm behind you to 37 cm ahead. Its middle is 22 cm to your right and 13.5 cm ahead. - **The parked cars** are 14 cm wide and 22 cm long, from 16 cm to 30 cm to your right. The one behind runs from 35 cm to 13 cm behind you; the one ahead from 42 cm to 64 cm ahead. - **The robot** is 7 cm across, and touching a car counts from its edge, not its centre. To pass, the robot's centre must finish inside the bay, facing within 15 degrees of the way it started (a final heading between 345 and 15), without touching either car, inside 40 seconds. Use `car()`. The checker ignores how you call the motors and checks that the robot never slides, never turns on the spot and never turns tighter than `R_MIN`. Any speed up to 100 percent is allowed, forwards or in reverse.

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

V_MAX, W_MAX, R_MIN = 20.0, 120.0, 25.0

def car(speed, steer, seconds):
    """speed -100..100 (negative reverses), steer -1..1 (full lock left..right). Turn rate
is tied to speed."""
    speed = max(-100, min(100, speed))
    steer = max(-1, min(1, steer))
    v = speed / 100 * V_MAX
    w = math.degrees(v / R_MIN) * steer
    drive(speed, 0, 100 * w / W_MAX)
    wait(seconds)
    stop()
    wait(0.4)

# your moves, all with car()
```

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

Do it on the robot

www.bugbotlab.com/learn/u2-8-driving-like-a-car/

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

Challenges

1. Park in the same bay in three shuffles of shorter arcs instead of two long ones: a reversing pair, then a forward pair, then a reversing pair. How much road does it use compared with the two long arcs? Then work out why doing all three pairs in reverse needs far more road than either.

2. Run the wiggle with `θ` doubled and `d` halved. The sideways shift per cycle stays about the same. Why, from the formula, and what happens to the forward part?
3. A car-like robot is told to reach a point directly to its side, 30 cm away, as fast as possible. Sketch the path. How does the fastest path change if the robot may also reverse?