WORKSHEET

U2.5 Inverse kinematics

Kinematics and frames · University · about 30 min

Name _____ Class _____ Date _____

What this lesson is about

The useful direction: a velocity in the world, a heading, and the command that produces it.

- The problem
- In code
- Saturation

Questions

6 marks in all

1. The robot faces east, heading 90, and you want it to move due north at 12 cm/s. With V_MAX = 20 and V_LAT = 15, what does this print? [1 mark]

```
import math
V_MAX, V_LAT = 20.0, 15.0
h = math.radians(90)
wx, wy = 0.0, 12.0
vx = wx * math.cos(h) - wy * math.sin(h)
vy = wx * math.sin(h) + wy * math.cos(h)
print("fwd:", round(100 * vy / V_MAX), "lat:", round(100 * vx / V_LAT))
```

2. The robot is at heading 180, facing -y. You want a world velocity of 12 cm/s due east (+x), with V_LAT = 15. What lat command does the inverse kinematics give? [1 mark]

3. In which order does the inverse kinematics work? [1 mark]

- ☐ **A** Rotate the world velocity into the body frame, then divide each component by that axis's full scale
- ☐ **B** Divide the world velocity by each axis's full scale, then rotate into the body frame
- ☐ **C** Rotate with body_to_world, then divide by the full scales
- ☐ **D** Divide wx by V_LAT and wy by V_MAX, with no rotation

4. The inverse kinematics asks for fwd = 150 and lat = 60. This scales the command down the way the lesson does. What does it print? [1 mark]

```
fwd, lat, rot = 150.0, 60.0, 0.0
worst = max(abs(fwd), abs(lat), abs(rot), 100.0)
fwd, lat, rot = fwd * 100 / worst, lat * 100 / worst, rot * 100 / worst
print(fwd, lat, rot)
```

5. The same request, fwd = 150 and lat = 60, is instead clipped axis by axis to 100 and 60. What does the robot do? [1 mark]

- ☐ **A** Travels in the wrong direction as well as too slowly
- ☐ **B** Travels in the right direction, just more slowly
- ☐ **C** Does exactly what was asked, because the robot clips internally anyway
- ☐ **D** Stops, because an out-of-range command is rejected

6. Which request can a differential drive, with only a left and a right wheel speed, not satisfy? [1 mark]

- ☐ **A** A velocity straight sideways in its own frame
- ☐ **B** Driving forward while turning
- ☐ **C** Turning on the spot
- ☐ **D** Driving straight backwards

The task: north while facing east

The robot faces east. Get it into the green zone to the north without turning: heading must still be within 15 degrees of 90 at the end. Use the inverse kinematics.

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

V_MAX, V_LAT = 20.0, 15.0
```

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

Do it on the robot
www.bugbotlab.com/learn/u2-5-inverse-kinematics/
The simulator checks it and tells you when it passes. Nothing to install, no account.

Challenges

1. Add saturation handling to `world_drive()` and ask for 40 cm/s. Does the robot still go in the right direction?
2. Ask for a world velocity and a spin at the same time, and confirm the world path is unaffected by the spinning.
3. Write the inverse kinematics for a differential drive (left and right wheel speeds). Which requests become impossible?