WORKSHEET

4.3 Visual servoing

Vision · Robot club · about 20 min

Name _____ Class _____ Date _____

What this lesson is about

Steer from cx, speed from distance: docking at a tag.

- Two errors from one tag
- Steer while driving
- Speed from distance
- When the tag disappears

Questions

6 marks in all

1. A tag gives two errors at once. Which pairing does the docking loop use? [1 mark]

- ☐ **A** `cx` sets the rotation, `distance` sets the forward speed
- ☐ **B** `cx` sets the forward speed, `distance` sets the rotation
- ☐ **C** `cy` sets the rotation, `id` sets the speed
- ☐ **D** `distance` sets both

2. What does this program print? [1 mark]

```
stop_at = 13
for dist in [40, 20, 14]:
    speed = max(20, min(70, (dist - stop_at) * 3))
    print(speed)
```

3. With `stop_at = 13` and `speed = max(20, min(70, (dist - stop_at) * 3))`, what speed does the robot use when the tag is 25 cm away? [1 mark]

4. Why does the speed have a lower limit of 20, the `max(20, ...)`? [1 mark]

- ☐ **A** So the robot does not crawl to a halt before it reaches the stopping distance
- ☐ **B** So it never drives faster than 20
- ☐ **C** So it can reverse if it gets too close
- ☐ **D** Because the camera cannot see below 20 cm

5. Near the tag, the camera loses it for one frame and the loop does `if not tags: break`. Why is that a poor choice? [1 mark]

- ☐ **A** The robot stops dead even though the tag is almost certainly still there
- ☐ **B** `break` restarts the loop from the top
- ☐ **C** It makes the robot spin forever
- ☐ **D** It is an error to break out of a while loop

6. Put one tick of the visual servoing loop in order. [1 mark]

Number the lines 1 to 4 to put them in the right order.

___ `Read the tags with `apriltags()``

___ `Work out the bearing error and the distance error`

___ `Send one `drive` with a speed and a rotation`

___ `Wait 0.1 s, then go round again`

The task: dock at the marker

Drive to marker 1 and stop about 12 cm in front of it, square to it: the card faces straight down the mat, so that means heading 0, within 12 degrees. The docking loop above arrives at an angle and ends facing 17, so square up at the end. The starter drives at a fixed speed and only stops once it has lost sight of the tag, by then on top of the card.

```
# the two lines every program starts with: the commands, then the robot
from bugbot import *
connect()

def bearing_of(cx):
    return (cx - 160) * 120 / 320

# camera: the tag detector
set_cv("apriltag")
# again and again, for ever
while True:
    # every tag in view: [id, cx, cy, distance], nearest first
    tags = apriltags()
    if not tags:
        # leave the loop
        break
    # forward, sideways, rotation: -100 to 100 each, until the next command
    drive(50, 0, bearing_of(tags[0][1]) * 3)
    # pause 0.1 s (the robot keeps doing what it was told)
    wait(0.1)
# all motors off
stop()
```

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

Do it on the robot

www.bugbotlab.com/learn/4-3-visual-servoing/

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

Challenges

1. Dock at 20 cm, then at 8 cm. What is the closest the camera can still read the tag?
2. Add a sideways controller so the robot ends up square in front of the tag, not just facing it, without a turn at the end. Hint: the tag's bearing and the robot's heading together tell you which side you are on.
3. Time the dock. Make it faster without touching the tag.