WORKSHEET

U12.6 The reality gap, and honest evaluation

Learning, and the capstone · University · about 40 min

BugBotLab

Name _____ Class _____ Date _____

What this lesson is about

Why a policy tuned in one world fails in another, and how to report what it does without lying.

- Why a tuned policy breaks
- Domain randomisation
- What else closes the gap
- Honest evaluation

Questions

6 marks in all

1. A braking policy is run five times with the noise turned up. What does this print for the mean and spread of the gaps it left? [1 mark]

```
gaps = [31.0, 27.5, 36.2, 29.1, 22.4]
mean = sum(gaps) / len(gaps)
spread = (sum((g - mean) ** 2 for g in gaps) / len(gaps)) ** 0.5
print(round(mean, 1), round(spread, 1), round(max(abs(g - 30.0) for g in gaps), 1))
```

2. A braking threshold tuned with no sensor noise stops the robot too early once noise is added. Why? [1 mark]

- ☐ **A** The first time a noisy reading dips past the threshold counts as a crossing, before the true distance gets there
- ☐ **B** Noise makes the average reading smaller
- ☐ **C** The robot drives faster with noise on
- ☐ **D** The threshold was tuned on the wrong mat

3. A policy is tuned across three noise levels at once instead of one. What is the trade? [1 mark]

- ☐ **A** It is worse in the quiet world than a policy tuned there, in exchange for performance you can predict across worlds
- ☐ **B** It is better in every world
- ☐ **C** It needs no evaluation afterwards
- ☐ **D** It removes the need for feedback

4. Why is the first successful run a biased sample of how well a policy works? [1 mark]

- ☐ **A** You stop looking when it works, so the run you report is more likely a good one than a typical one
- ☐ **B** The first run is always slower
- ☐ **C** The battery is fuller on the first run
- ☐ **D** It is not biased, only noisy

5. Which are honest ways to report a learned policy? [1 mark]

Tick every answer that is true.

- ☐ **A** Tune at one noise level and report at another
- ☐ **B** Report the number of runs and include the failures
- ☐ **C** Compare it with the obvious hand-written feedback controller
- ☐ **D** Report the worst case when the worst case is what matters
- ☐ **E** Drop runs where the robot hit the wall as outliers
- ☐ **F** Report the best of ten runs

6. If you can choose only one defence against the reality gap, which does the lesson recommend? [1 mark]

- ☐ **A** Prefer feedback, because a closed loop tolerates a model that is 20 percent wrong
- ☐ **B** Randomise the simulator more widely
- ☐ **C** Tune for longer in the lab
- ☐ **D** Use a larger hypothesis class

The task: tune it, then stress it

Tune the reading at which the robot starts braking, so that it ends up in the green band, 30 cm from the wall. Plot `score` as you tune. Then turn the noise up with `set_noise()`, run the tuned policy three times, and print `mean:` and `spread:` of the gaps it left. Put the noise back where it was for the run you hand in. Start every trial from the same distance, and start it well clear of the threshold you are testing. A trial that begins two centimetres above the braking point triggers on the first unlucky reading and scores nonsense, and the search will believe it.

```
from bugbot import *
connect()

DT = 0.1
CRUISE = 85
GAP = 30.0
START_GAP = 55.0
```

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

Do it on the robot

www.bugbotlab.com/learn/u12-6-the-reality-gap/

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

Challenges

1. Tune at `set_noise(0)` and evaluate at `set_noise(3)`. How much worse is it than tuning with the noise on?
2. Tune against the mean score across three noise levels at once. What does it cost you in the quiet world?
3. Replace the learned threshold with a proportional controller on the gap error and compare both on mean and spread. Which would you ship?