

Exercise: Novelty Detection

1. Robot with deterministic fixed behavior

Take your robot simulation from the previous exercise or write a simple simulation, where a robot of circular form with four distance sensors tries to explore some 2D world (which you could paint e.g. in a drawing program). Implement a simple collision avoidance behavior where the robot uses different states in order to avoid collision with non-free world cells. Make sure, that the robot behavior is fully deterministic (i.e., do not use random numbers). Then observe what happens. Can the robot always escape local regions or does it get stuck in some local region due its deterministic behavior as shown here?

<https://www.youtube.com/watch?v=WUggP5YKdgs>

Paint a 2D test world such that you can also observe a similar case as shown in the video.

2. Novelty Detection

Now think about an idea how you could detect that the robot got stuck in some region.

Implement such a “*novelty detection*” mechanism for your robot that allows the robot to recognize whether some part of the world has been visited before or not. Do not assume a 2D position sensor for the robot, i.e., the robot has to recognize whether it has been in some region of the world before nor not just based on its sensor values.

Then, whenever it drives around in an already known part of the 2D world for a longer time, let it make some non-deterministic (random) movement (e.g. turning by some random angle), in order to give it a chance to leave this part of the world while its deterministic behavior would result in staying there forever.