

# ADLR WS21 - Milestone Report Team 04

Marc Hauck  
*Robotics, Cognition, Intelligence*  
*Technical University of Munich*  
Munich, Germany  
ge65qoy@mytum.de

Baris Tura  
*Robotics, Cognition, Intelligence*  
*Technical University of Munich*  
Munich, Germany  
baris.tura@tum.de

## I. INTRODUCTION

In this project, ...

## II. EXPERIMENTS

### A. Discrete Environments

To get familiar with the stable-baselines library and the environment inherited from the Open AI gym class, we started with an empty 5x5 grid world environment with no obstacles. The agent started in the top left corner and had to find the goal in the bottom right corner by taking one of the actions UP, DOWN, RIGHT and LEFT. The reward was set to 1 at the goal and 0 else. Using a discrete observation and action space, the experiments needed only little computation time and we were able to obtain results quickly. Since it was easy for the agent to find the goal in this small environment, we extended it to a 10x10 grid world. This led to an increase of training time that was needed for the agent to succeed, because it needed more exploration time to find the sparse reward.

In a second step, we added obstacles to the environment and initially rewarded a collision with -1. Adding the obstacles led to a massive increase of computation time as they increased the dimension of the observation space depending on the number of obstacles. Additionally, it required changes of the reward function: If the negative reward of a collision was too small compared to the reward for reaching the goal, the agent would ignore the obstacle and step over it. This was still possible because we did not reset the environment upon collision. If the negative reward for colliding was too high compared to the goal reward, the agent would not move at all and stay in its corner. By greatly increasing the number of training steps and finding a good ratio between positive and negative rewards by using binary search, we were able to successfully train the agent in this environment.

### B. Continuous Environments

Next, we transformed the environment to a continuous one. To enable the agent to freely move in the continuous space, the action space was set to a two-dimensional interval from -0.2 to 0.2 for the agent's velocity in x- and y- direction. The experiments started with the sparse rewards from the discrete environment and no obstacles, but the agent was not able to find the goal reliably. As long as the positions of start and goal were fixed, the agent was able to overfit to the goal's position and find it, but if we sampled start and goal position only from

small intervals, the agent often walked past it and ended up stuck in the opposite corner. To overcome this problem, we had to change the reward function in a way that guided the agent more towards the goal. To do that, we set the reward after every step to the negative distance of the agent to the goal which led to good results with little training time.

As in the discrete environment, we then added obstacles to it and provided a negative reward for a collision. If the agent did not collide, the reward stayed the same as before. We were able to successfully guide the agent to the goal for up to three obstacles, observing an increase of training time with a higher number of obstacles again. For environments with more obstacles, the required training steps exceeded numbers that made sense for our project.

### C. Changing the Obstacle Representation

xxx

## III. CONCLUSION AND OUTLOOK

xxx

## REFERENCES

- [1] T. Jurgenson and A. Tamar, "Harnessing reinforcement learning for neural motion planning", *arXiv preprint arXiv:1906.00214*, 2019.
- [2] B. Wang, Z. Liu, Q. Li, and A. Prorok, "Mobile robot path planning in dynamic environments through globally guided reinforcement learning", *IEEE Robotics and Automation Letters*, 5(4), pp. 6932-6939, 2020.