technische universität dortmund

computer science 12

Autonomous Racing Using Reinforcement Learning in ROS2

Adem Jabri

Department of Computer Science, Chair 12 (DAES)
TU Dortmund University, Germany

15. Juli 2024

technische universität dortmund

computer science 12

Outline

technische universität dortmund

computer science 12

Autonomous Driving

Figure: Autonomous driving.

- Autonomous driving technology enables vehicles or any kind of mobile machines to navigate without human intervention.
- The autonomous vehicules utilize a combination of sensors, cameras, and advanced computational algorithms in order to achieve this task.

technische universität dortmund

computer science 12

Traditional approach

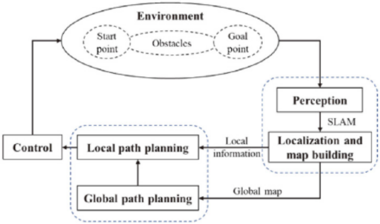


Figure: Traditional robot navigation framework.

- **Time-Intensive Map Handling:** The process of establishing and updating the obstacle map in SLAM is time-consuming.
- **Dependence on Sensor Density:** The performance of SLAM algorithms heavily relies on the accuracy of the laser sensor data.

RL as a possible replacement

- Reinforcement Learning (RL) is preferred over other deep learning methods due to its reduced dependency on large volumes of labeled data.
- This black box system will develop its capability through a training process based on trial and error.

technische universität dortmund
computer science 12

How will it work with RL?

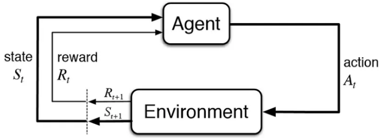


Figure: The action-reward feedback loop of a RL model.

- The agent learns to execute tasks by taking actions and observing the outcomes.
- Desirable actions are reinforced through rewards, making them more likely to be repeated.
- Actions that lead to unfavorable outcomes are given negative rewards, or "punishments".
- PPO Algorithm was used.

technische universität dortmund

computer science 12

Outline

technische universität dortmund

computer science 12

Objective

- This thesis explores the integration of ROS 2 with Reinforcement Learning to develop an autonomous robot navigation system capable of safely and efficiently navigating any racetrack, even when training an agent on only one given racetrack.

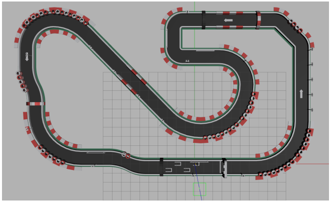


Figure: Simulated racetrack used for training the autonomous agent.

technische universität dortmund

computer science 12

Outline

technische universität dortmund

computer science 12

Software Components

- **Gymnasium:** serves as an API for RL environments.
- **Stable Baselines3:** builds upon Gymnasium by providing high-quality implementations of advanced RL algorithms.
- **Gazebo:** used to create simulations of both the environment and robot.
- **ROS 2:** connects this components with each other.

technische universität dortmund

computer science 12

System Architecture

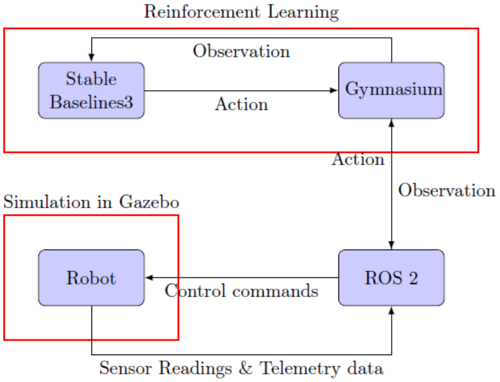


Figure: System architecture.

technische universität dortmund

computer science 12

Outline

technische universität dortmund

computer science 12

Challenges

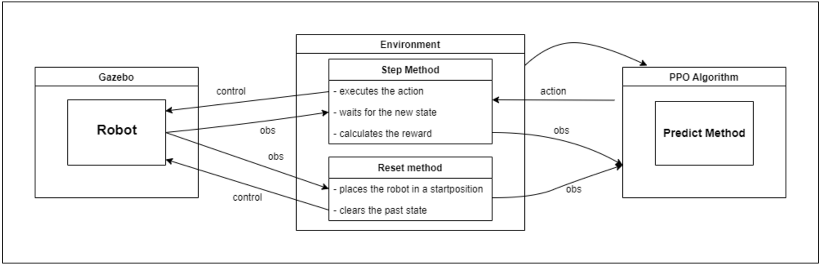
1. Train the agent to reach the targets as fast as possible
2. Avoid overfitting, so that the robot will be able to navigate other non seen racetracks
3. Efficient training
4. Safe training

technische universität dortmund

computer science 12

Outline

Robot Control Dynamics



Initialisation of the environment

Start and target positions

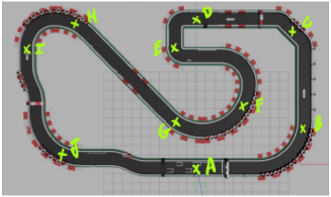


Figure: Used waypoints in the racetrack.

- These positions are used to randomize the starting points at the beginning of each episode.
- The robot was spawned also randomly in both directions.
- During the first phase of the training the target position was the waypoint directly after the start position.

technische universität dortmund

computer science 12

Initialisation of the environment
Action and Observation Space

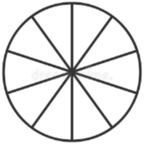


Figure: Sampled LiDAR readings.

- The Observation Space includes 10 signals from LiDAR readings and includes also the linear speed and the angular velocity.
- The action space was defined as a continuous space.

technische universität dortmund

computer science 12

Reward function

- Designing a reward function in Reinforcement Learning is often considered more of an art than a science, as there is no definitive guide or rules to follow.
- The reward function was designed based on the objectives of this thesis:
  1. Navigation toward a target:
     - **Target Reached Reward**
     - **Progress Reward**
  2. Rapidity and efficiency:
     - **Time Penalty**
     - **Steering Penalty**
     - **Speed Reward**
  3. Safety:
     - **Collision avoidance score**

technische universität dortmund

computer science 12

During the Training

- During every training session, the model parameters were saved every 3000 steps.
- After each session, new choices and strategies were implemented to address the problems observed during the previous session. The choices and strategies made include:
  1. Reward shaping
  2. PPO model Adjustment
  3. Deciding at which waypoints the robot will be spawned each time.
  4. Adjusting the weights of some waypoints to increase the probability that the robot will be spawned at that position.

technische universität dortmund

computer science 12

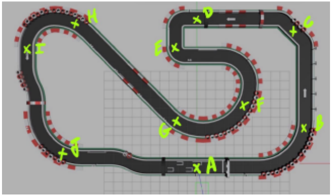
First training

First generation

- PPO model configuration:

```python
# PPO model configuration
self.model = PPO("MultiInputPolicy", self.env, verbose=1,
                 tensorboard_log=self.log_dir, batch_size=64, n_steps=2048,
                 n_epochs=10, learning_rate=0.001, ent_coef=0.01,
                 clip_range=0.1, gamma=0.99, gae_lambda=0.95,
                 policy_kwargs= {'net_arch': [400, 300]})
```

- The robot was trained on the simplest segment of the racetrack:



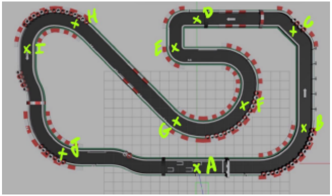
- Good results were observed.

technische universität dortmund

computer science 12

First training
Second generation

- PPO model configuration:
  - The learning rate was decreased.
  - The entropy coefficient was increased.
  - The discount factor gamma was decreased to 0,98.
- The robot started to learn in more diffucult segments:

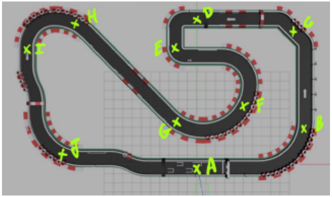


- Very bad results were observed as the agent encountered the well-known catastrophic forgetting problem.

technische universität dortmund

computer science 12

Second training
First generation



- The agent is now trained from scratch once again.
- Same PPO model configuration as first training was used.
- The start positions now include all the waypoints except D and E.
- More weight has been given to the waypoints I, H, G, and F.

Second training

First generation

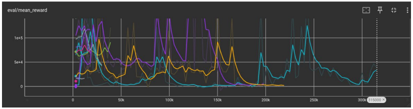


Figure: Evaluation of Mean Reward of the second training.

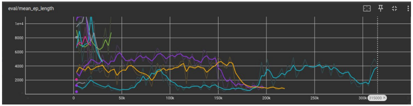
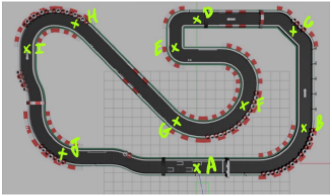


Figure: Evaluation of Mean Episode Length of the second training.

- Better results were observed. But the robot still make mistakes.
- For improvement, the model saved after 135k steps was used.

Second training

Second generation



- PPO model configuration:
  - The learning rate was decreased.
  - The discount factor gamma was decreased to 0,6.
  - The batch size and the n_steps were increased to 264 and 4096.
- The safety threshhold was decreased and the punishement for surpassing this threshhold was also increased.
- The steering penalty was more decreased.
- The waypoints B and G are no more considered as waypoints.
- Same weights has been given to the waypoints I, H and F.

technische universität dortmund

computer science 12

Second training

Second generation

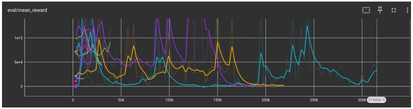


Figure: Evaluation of Mean Reward of the second training.

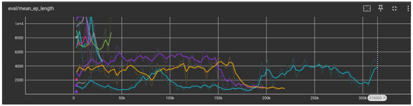


Figure: Evaluation of Mean Episode Length of the second training.

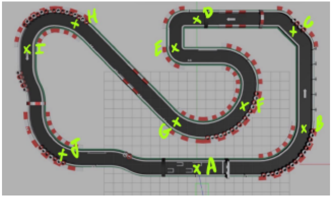
- The behavior of the robot during the training was perfect.
- The robot was tested for the first time on the entire racetrack:
  - The robot failed to navigate from D to E.
  - The linear velocity was not surpassing 0,6 m/s.
- For improvement, the model saved after 66k steps was used.

technische universität dortmund

computer science 12

Second training

Last generation



- Same PPO model configuration was used.
- The punishement for surpassing the safety threshhold was increased to -200.
- The progress reward was decreased.
- A speed reward was added.
- The start positions now include only C and F, with the respective target positions being G and B.
- More weight has been given to the start position at C.

technische universität dortmund

computer science 12

Outline

video

Generalization check

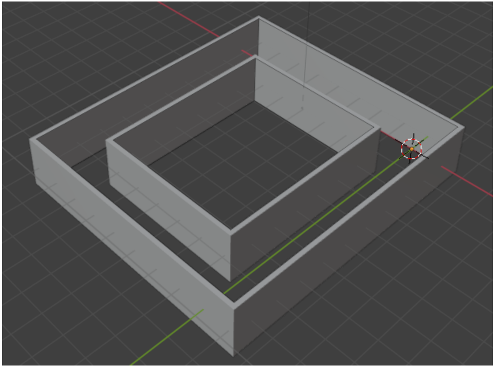


Figure: The new racetrack to test the agent's generalization.

Generalization check

Video

technische universität dortmund

computer science 12

Outline

technische universität dortmund

computer science 12

Future work

- Try increasing the complexity of the scenario by adding other robots during the training. This will enhance the agent's decision-making capabilities when dealing with dynamic obstacles.
- This might necessitate:
  1. the integration of a greater number of LiDAR readings
  2. more precise reward shaping
  3. complexer network architecture for the policy

Resources

- Reinforcement Learning: An Introduction
  https://ieeexplore.ieee.org/document/712192
- Proximal Policy Optimization Algorithms
  https://doi.org/10.48550/arXiv.1707.06347
- Lidar SLAM: The Ultimate Guide to Simultaneous Localization and Mapping
  https://www.wevolver.com/category/autonomous-vehicles
- Learning Navigation Behaviors End-to-End with AutoRL
  https://arxiv.org/abs/1809.10124
- Gymnasium Documentation
  https://gymnasium.farama.org/tutorials/gymnasium$_b asics/$
- Stable Baselines
  https://stable-baselines.readthedocs.io/en/master/

Thank you for your attention!

Questions?