technische universität dortmund

computer science 12

Autonomous Racing Using Reinforcement Learning in ROS2

Adem Jabri

Department of Computer Science, Chair 12
TU Dortmund University, Germany

22. Januar 2024

Outline

Motivation

Figure: Autonomous robot systems.

- The key challenge in this field is efficient path following, especially in dynamic and unpredictable environments.
- This thesis explores the use of Reinforcement Learning in ROS2 to address these challenges, offering a novel approach to autonomous navigation.

Traditional approach

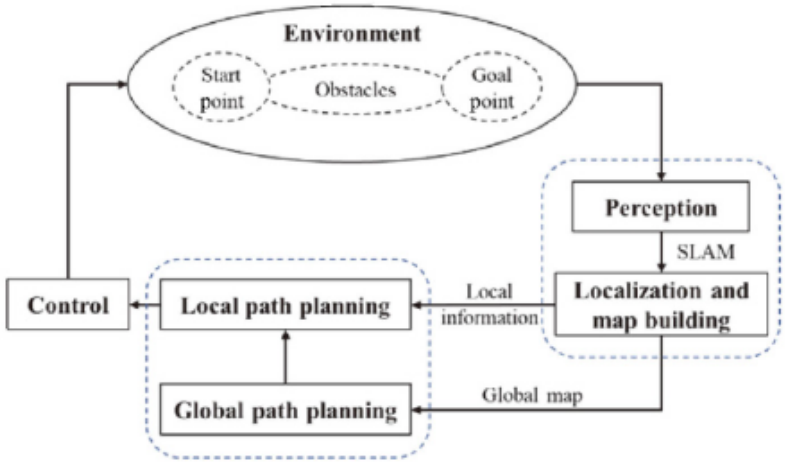


Figure: Traditional robot navigation framework.

- **Sensitivity to Conditions:** SLAM can be prone to failure due to changes in the environment or the sensor's conditions.
- **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.

Traditional approach

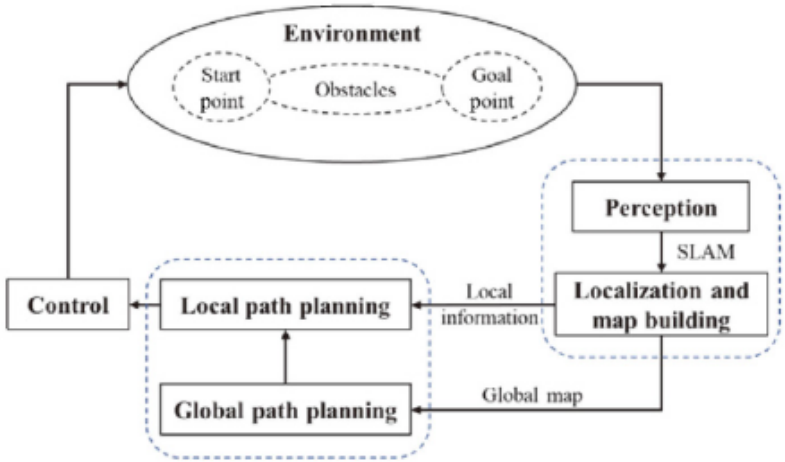


Figure: Traditional robot navigation framework.

- **Sensitivity to Conditions:** SLAM can be prone to failure due to changes in the environment or the sensor's conditions.
- **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.

How will it work with Reinforcement learning?

What is Reinforcement Learing?

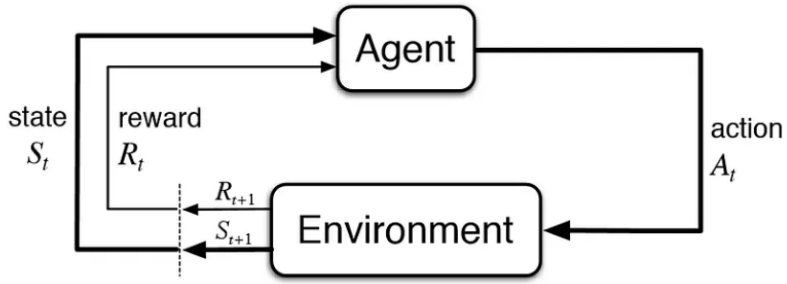


Figure: The action-reward feedback loop of a RL model.

- 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".
- The goal of reinforcement learning is to find a suitable action model that would maximize the total cumulative reward of the agent.

How will it work with Reinforcement learning?

What is Reinforcement Learing?

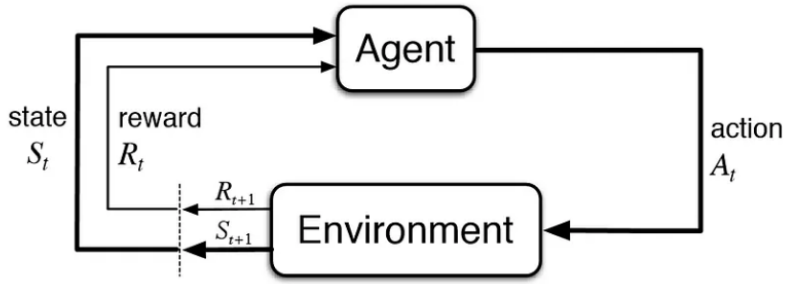


Figure: The action-reward feedback loop of a RL model.

- 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".
- The goal of reinforcement learning is to find a suitable action model that would maximize the total cumulative reward of the agent.

How will it work with Reinforcement learning?

- **Interaction-Based Learning:** The RL method enables the robot to learn the optimal navigation policy through direct interaction with the environment.
- **Mapless Navigation:** Unlike SLAM, RL doesn't require pre-existing maps, allowing the robot to navigate without detailed environmental data.
- **Adaptive Policy Development:** Through its learning process, RL develops a navigation policy that guides the robot to its target position adaptively.
- **Strong Learning Ability:** RL can efficiently process complex data and learn from diverse scenarios, improving over time.
- **Sensor Independence:** RL reduces reliance on high sensor accuracy, which is beneficial in environments where sensor data may be unreliable or noisy.

Outline

Objective

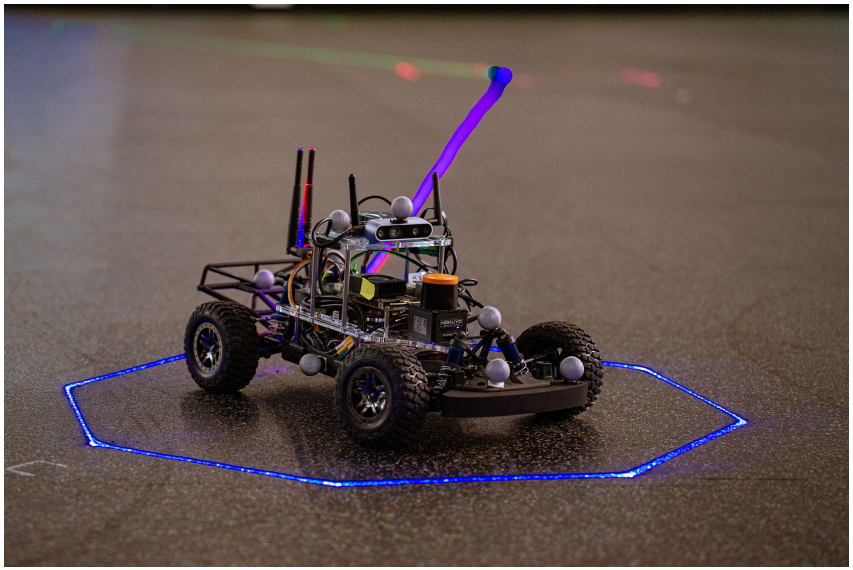


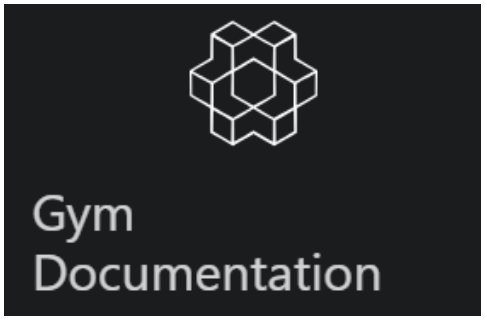
Figure: The autonomous robot.

- Develop an autonomous robot navigation system that can **safely** and **efficiently** navigate **any environment** using a reinforcement learning algorithm and the ROS2 Navigation2 stack.

Outline

Outline

Step 1:



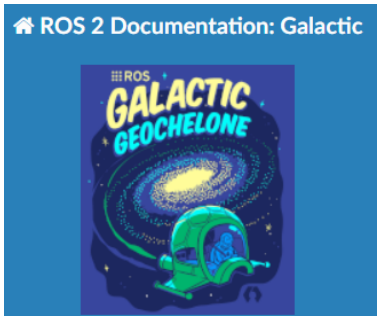


Figure: Integrated tools.

- Implement a Reinforcement Learning algorithm to find the optimal policy for guiding the robot to its target position through interaction with the environment.

Step 2:

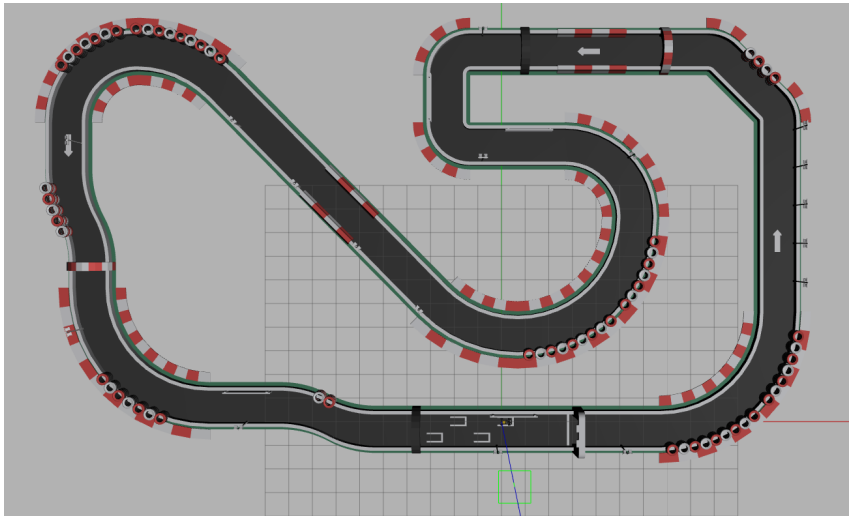


Figure: Racetrack in Gazebo Simulator.

- Create a simulation environment using ROS2 and Gazebo, with a robot that is equipped with a LIDAR sensor and a reinforcement learning algorithm.

Step 3:

Figure: Time to work hard.

- Train the reinforcement learning algorithm to navigate the environment.

Step 4:

Figure: Time to see what you learned.

- Test the robot's ability to navigate the environment safely and efficiently.

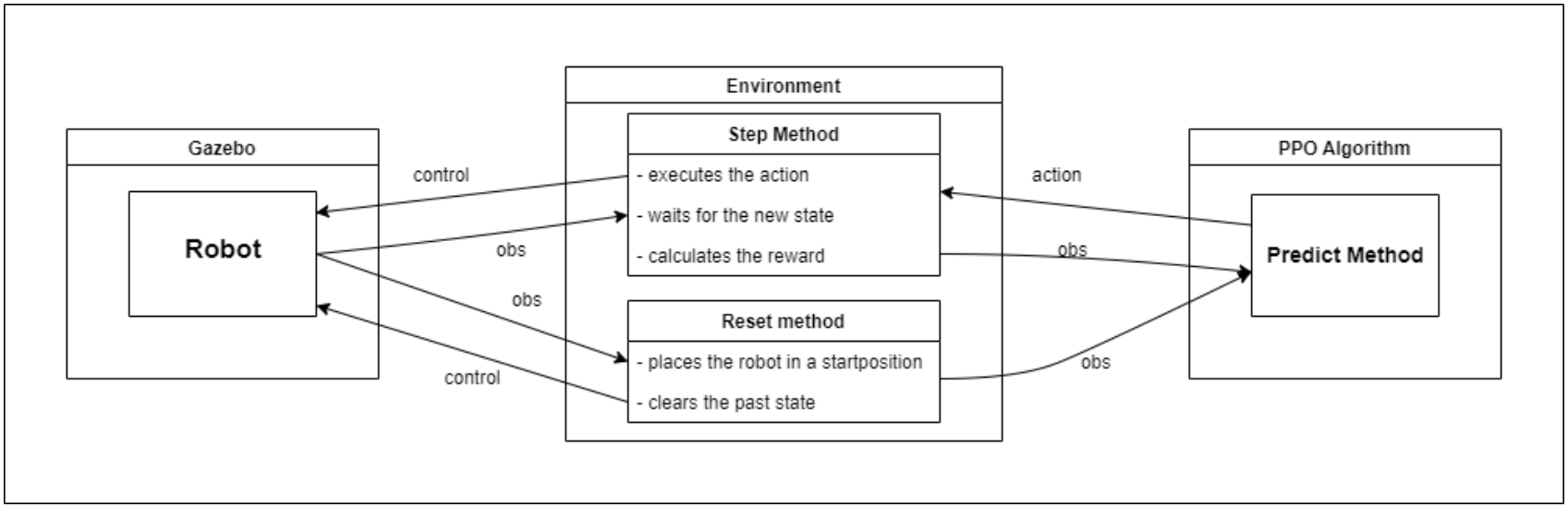
Step 5:

Figure: Truth time.

- Compare the system with the classical navigation algorithms for ROS 2, such as the Navigation2 stack.

Outline

RL based navigation system



Proximal Policy Optimization (PPO)

- A policy describes the behavior of an agent given a state.
- PPO estimates how much better an action is compared to the average action in a given state.
- PPO can be applied to a wide range of environments, both discrete and continuous action spaces.
- It uses a **clipped surrogate objective function**, which prevents too large policy updates, enhancing training stability.
- This objective function provides a balance between **exploration** (trying new things) and **exploitation** (using what is known).

Proximal Policy Optimization (PPO)

- A policy describes the behavior of an agent given a state.
- PPO estimates how much better an action is compared to the average action in a given state.
- PPO can be applied to a wide range of environments, both discrete and continuous action spaces.
- It uses a **clipped surrogate objective function**, which prevents too large policy updates, enhancing training stability.
- This objective function provides a balance between **exploration** (trying new things) and **exploitation** (using what is known).

Outline

Efficient Training

Figure: A Race against Time.

- Reinforcement Learning algorithms demand significant computational time.
- Training in a simulated environment makes it faster.
- Implement a reward function that penalizes time to promote quicker navigation.
- Simplify sensor data processing by focusing on the nearest obstacle per sensor section for efficiency.

Efficient Training

Figure: A Race against Time.

- Reinforcement Learning algorithms demand significant computational time.
- Training in a simulated environment makes it faster.
- Implement a reward function that penalizes time to promote quicker navigation.
- Simplify sensor data processing by focusing on the nearest obstacle per sensor section for efficiency.

Safe Training

- Training in a simulated environment avoids any real-world accidents.
- Including penalties for unsafe actions in the reward function discourages such behaviors.
- Define a safety threshold distance from obstacles, beyond which the robot is considered to have crashed and incorporate it into the reward function.

Overcoming Specialization

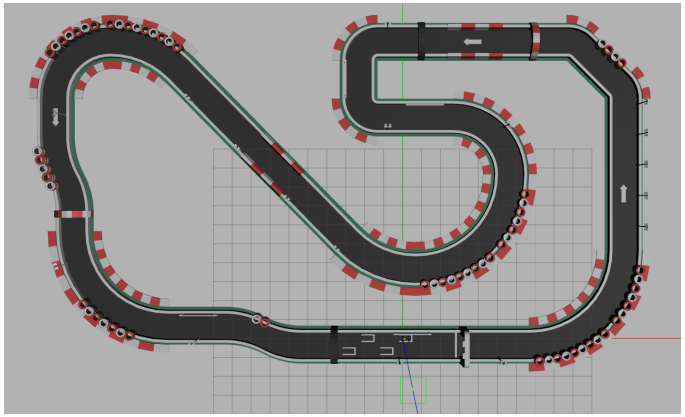


Figure: Race Track in Gazebo Simulator.

- The model, if overfitted, may navigate effectively only in the specific environment it was trained on, lacking the ability to generalize to new unseen environments.
- Implementing domain randomization to promote generalization.
- Increase the complexity of the environment to progressively challenge the model.

Overcoming Specialization

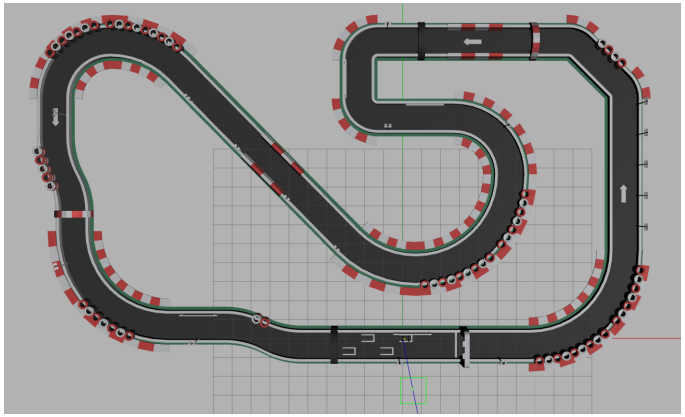


Figure: Race Track in Gazebo Simulator.

- The model, if overfitted, may navigate effectively only in the specific environment it was trained on, lacking the ability to generalize to new unseen environments.
- Implementing domain randomization to promote generalization.
- Increase the complexity of the environment to progressively challenge the model.

Overcoming Specialization

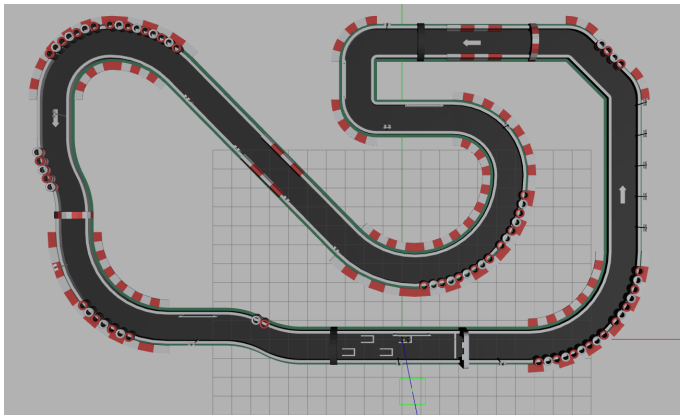


Figure: Original Racetrack.

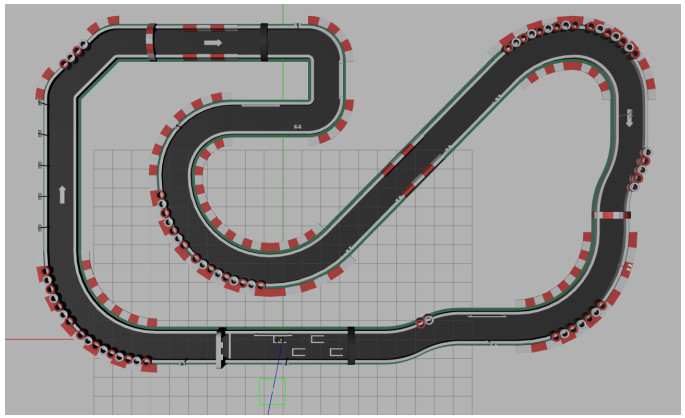


Figure: Mirrored Racetrack.

- Training on both the original and mirrored versions of the racetrack promotes model generalization and mitigates the risk of overfitting, ensuring robust performance in varied track conditions.

Resources


Deep reinforcement learning based mobile robot navigation: A review
https://ieeexplore.ieee.org/document/9409758


Proximal Policy Optimization Algorithms
https://doi.org/10.48550/arXiv.1707.06347

Questions ?