Technische Universität München
Lehrstuhl für Datenverarbeitung

Applied Reinforcement Learning Project

-Title-

Alperen Gündogan, Rachid Ellouze, Uzair Akbar

May 13, 2019

1 Motivation

Some robotic applications in the real world i.e. autonomous driving, shipping in warehouse are difficult to engineer, therefore a agent that exploits reinforcement learning algorithms would become necessary to tame complex models. The RL agent learns from the interactions i.e. experiences with the real world. The agent does not require a predefined model of the problem and a good design of the algorithm leads to a successful result to solve the complex tasks. The key difference of RL from the other learning algorithms is that the RL considers the whole problem during the learning process while other methods e.g. supervised learning only focuses on sub-problems. In this project, we aim to apply deep reinforcement learning algorithm for obstacle avoidance. * Add something about the algorithm that we will choose, advantages and maybe put some references of other papers.

2 Goals

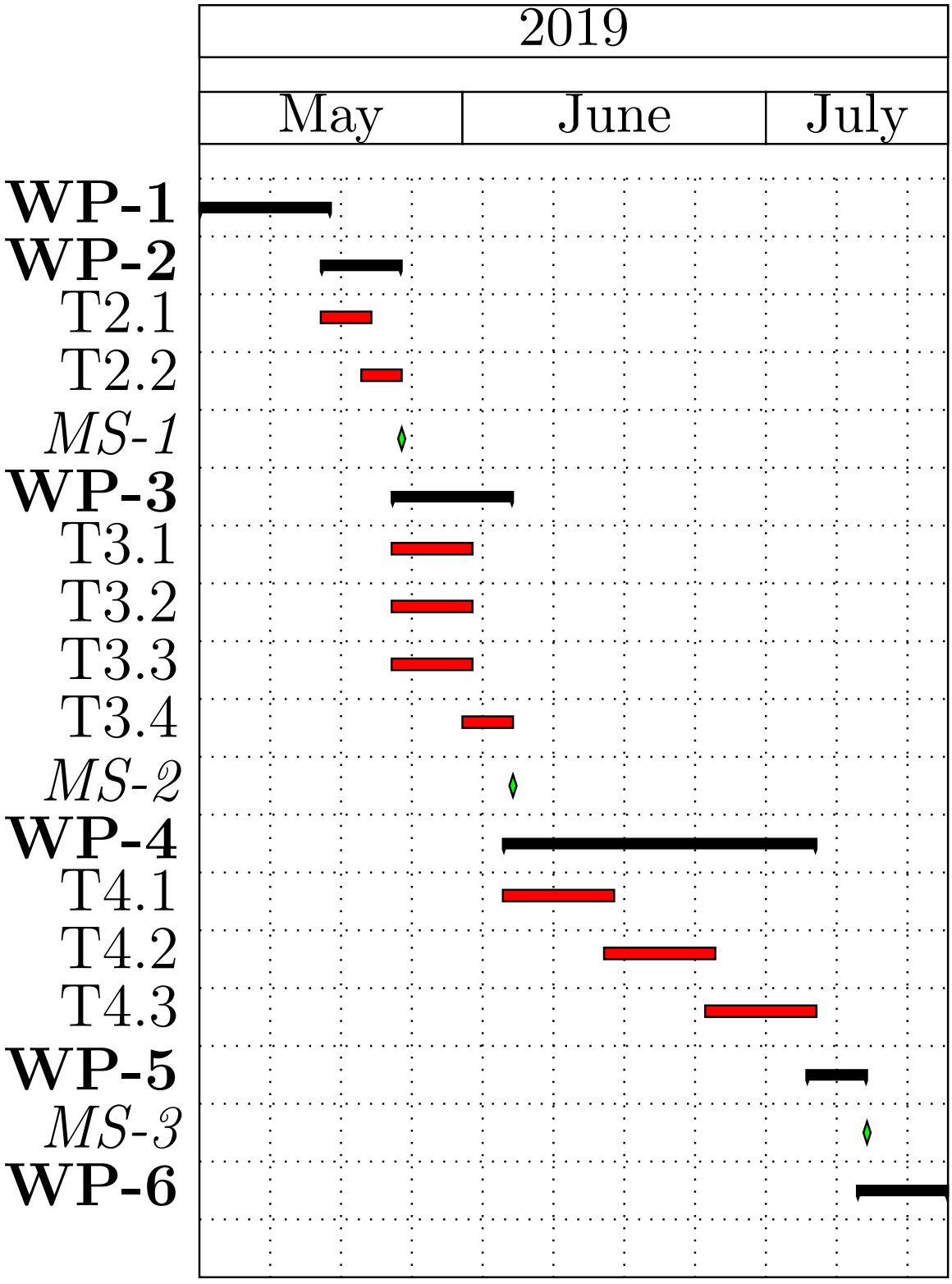
- The robot can reach the goal position starting from any position.
- The robot avoids the collision with obstacles during its path.
- The robot moves smoothly and reaches to the goal position as fast as possible.

3 Project Steps

- Get familiar with the turtlebot2, ROS, sensor, libraries.
- Setup the simulation framework.
- Implement basic functionalities i.e. movement, rotation, reading sensors etc.
- Apply different RL approaches containing algorithms, reward signals and states.
- Validation of the final approach.
- Final report and presentation.

4 Time Plan

- Work Package 1: Tutorials(Alperen, Rachid, Uzair)
  1. Practice with ROS, actors, sensors and libraries.
- Work Package 2: (Alperen, Rachid, Uzair)
  1. Setup of the environment and simulation framework.
  2. Implement basic functionalities.(*Milestone 1*).


2019
May
June
July
WP-1
WP-2
T2.1
T2.2
MS-1
WP-3
T3.1
T3.2
T3.3
T3.4
MS-2
WP-4
T4.1
T4.2
T4.3
WP-5
MS-3
WP-6

- Work Package 3: Apply a draft algorithm in the simulation environment.
  1. Feature extraction design (...)
  2. Reward function and action space design (...)
  3. Implement reinforcement learning algorithm.
  4. Tests with the simulation and decide for the parameters i.e. discount factor, learning rate.
- Work Package 4: Test with the real agent and apply different approaches containing algorithms, reward signals and states.
  1. Approach A (i.e. TD Learning with LFVA)
  2. Approach B (SARSA(λ))
  3. Approach C (Q-learning)
- Work Package 5: Decide and validate the final approach.
- Work Package 6: Presentation and final report.