The SimuLane Highway Traffic Simulator for Multi-Agent Reinforcement Learning

Manon Legrand*, Roxana Rădulescu, Diederik M. Roijers, and Ann Nowé

Vrije Universiteit Brussel, Artificial Intelligence Lab
manon.legrand@live.be,{rradules,droijers,ann.nowe}@ai.vub.ac.be

Abstract. SimuLane[1] is a highway traffic simulator modeling a multi-agent learning environment. Human drivers are simulated through a behavior-based model translating basic characteristics and desires of real-life drivers. SimuLane is designed in a fully parameterizable way and serves as a learning environment for self-driving cars; as such, a ratio of autonomous drivers – all using the same learning model – can be defined for either training or a simple simulation.

1 Simulator Overview

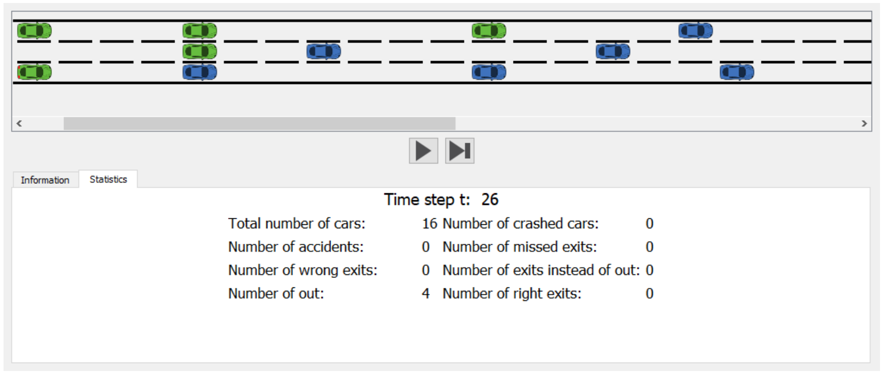
SimuLane features a fully discrete representation of straight highway lanes, i.e., each lane is represented as a list of cells by which cars pass. Each lane has a preferred speed, reflecting the idea that cars' speed usually increases from right to left on highway lanes. Exits are subsets of consecutive cells in the exit lanes preceded by an indicator. The number of exits, the size of the exits and the distance between them are all parameters of the simulator.

Another important component of the simulator are the human drivers. They act according to a behavioral model which can be summarized by three "rules": "*the driver must not crash*", "*the driver must reach their desired speed*" and "*the driver must respect the lanes' speeds*". To simulate the *human* nature of these drivers, the fact that they make mistakes or non-logical choices, the *irrationality* is defined by the probability that the driver chooses an action randomly.

The *traffic density* is defined as the probability for each lane that a new driver arrives in that lane. The first cell of each lane is thus updated according to the traffic density at every step. The simulation step for the other highway constituents is divided into two sub-steps: the *observation* and the *update* phases. During the former, drivers in the simulator "observe" their environment and chooses an action while the actual state of the highway is updated during the latter. The observation phase introduces an arbitrary, pseudo-temporal order over the drivers' actions; a driver choosing their action is aware of the action – the direction and the acceleration – of the drivers who precede them. Finally, if a crash occurs, it "stays" in the cell for a certain number of simulation steps. While it is present, the cell must be avoided by the drivers, otherwise it would cause another crash.

* This work was carried out for the first author's master thesis studies at VUB [1].

[1] Demonstration video: `https://youtu.be/tHBzCNTKA6w`



```
Time step t:  26
Total number of cars:        16  Number of crashed cars:          0
Number of accidents:          0  Number of missed exits:          0
Number of wrong exits:        0  Number of exits instead of out:  0
Number of out:                4  Number of right exits:           0
```

Fig. 1. Graphical interface of SimuLane where simulated human drivers are the blue cars, and the autonomous cars, the learning agents, are the green cars.

2 Learning Environment

The simulator aims to be used as the learning environment for a reinforcement learning problem with *autonomous cars* as agents [2]. SimuLane allows agents to retrieve their local state in the environment (limited by their field of view) and perform their selected actions. An action is a tuple <*direction, acceleration*>; the direction can be left, forward or right, and the acceleration is bounded by the *maximum acceleration* of the car. The reward function is fully customizable for each possible outcome (e.g., crash, missed exit, correct exit). SimuLane can therefore also easily be extended for multi-objective RL [3, 4].

SimuLane offers two learning settings: single-agent and multi-agent. In the first case, there is only one autonomous car in the highway at a time, together with "human" drivers. In the second case, the *ratio* of autonomous cars is the probability that a new driver entering the highway is an autonomous car; in this multi-agent setting, all agents use the same learning model during learning (i.e., centralized training but decentralized execution).

SimuLane is implemented in Python, version 2.7 and can be used in combination with any Python-based (deep) reinforcement learning library.

Acknowledgements This work is supported by Flanders Innovation & Entrepreneurship (VLAIO), SBO project 140047: Stable MultI-agent LEarnIng for neTworks (SMILE-IT). Diederik Roijers is an FWO postdoctoral fellow (grant number 12J0617N).

References

1. M. Legrand. Deep Reinforcement Learning for Autonomous Vehicle Control among Human Drivers. Master dissertation, Vrije Universiteit Brussel, 2017.
2. M. Legrand, R. Rădulescu, D.M. Roijers, and A. Nowé. Neural network reuse in deep RL for autonomous vehicles among human drivers. In *BNAIC*, 2017.
3. H. Mossalam, Y.M. Assael, D.M. Roijers, and S. Whiteson. Multi-objective deep reinforcement learning. In *NIPS workshop on Deep RL*, 2016.
4. D.M. Roijers, P. Vamplew, S. Whiteson, and R. Dazeley. A survey of multi-objective sequential decision-making. *JAIR*, 48:67–113, 2013.