AGH University of Krakow

FIELD OF SCIENCE: ENGINEERING AND TECHNOLOGY

SCIENTIFIC DISCIPLINE: AUTOMATION, ELECTRONIC, ELECTRICAL ENGINEERING AND SPACE TECHNOLOGIES

SUMMARY OF ACCOMPLISHMENTS

Applications of reinforcement learning methodologies to autonomous driving

Author: Mateusz Orłowski

Supervisor: dr hab. inż. Paweł Skruch, prof. AGH
dr inż. Krzysztof Kogut

Completed in: Faculty of Electrical Engineering, Automatics, Computer Science and Biomedical Engineering

Kraków, 2024

1 Abstract

The autonomous driving (AD) field is currently one of the most advanced and active frontiers in technology development, which needs to address both perception and control problems. Today, AD cars are required to deal with more and more complex environments and scenarios, which often require a data-driven approach to solve. At the same time, reinforcement learning (RL) is a subfield of artificial intelligence which aims at developing intelligent agents capable of acting in predefined environments. This work summarises the research conducted using reinforcement learning methodologies to control the motion of an autonomous car. By performing a series of experiments, we were able to test how different control approaches can be used in combination with the RL policy and what kind of road scenarios can be solved with such a methodology.

In the first experiment, we trained the agent to control the behaviour of the simulated car in a highway environment using a high-level control interface, defining the manoeuvre and the velocity set point. Execution of this control has been in charge of deterministic, model-based methods. The agent's goal was to reach the lane-based goal, defined in a predefined distance in the shortest time while adhering to traffic rules and optimising comfort. We examine how different strategies for executing agent action impact both functional performance and training efficiency. In the second experiment, an RL agent was trained to derive the path of a vehicle aiming to park itself at a predefined spot. With straightforward reward design and problem definition, the agent was able to park in complex parking scenarios, including parallel and perpendicular parking spots. In these experiments, we also tested the use of different neural network architectures and checked their impact on functional and computational performance. In the last series of experiments, we applied RL to a multi-agent coordination problem, where multiple cars need to navigate complex road scenarios, such as bottleneck or cross-road. All of the vehicles in the scene were controlled with the same RL-trained policy and was able to derive successful strategies to navigate those challenging scenarios. We were able to show that using the reward-sharing mechanism, in which each agent was rewarded for its individual and group performance, improves the overall performance of the group and speeds up training.

In summary, we were able to demonstrate that reinforcement learning methodology can be successfully applied to the autonomous driving domain, although its application to the production environment requires a careful design of the whole system. However, we think that the presented research proves that RL methodologies apply to the AD domain, and might be necessary to solve the most challenging road scenarios.

2 Introduction and Motivation

Autonomous Driving has been recently one of the most advanced endeavours to bring automation to the broad public. Advanced driver-assisted systems have become standard equipment for most of the car brands. The initial years of development of such systems focused mostly on the perception systems development, keeping the feature function limited in scope and capability. However, currently, with the advanced perception of the outer world and access to maps, prospects handling more elaborate scenarios arose, and decision-making became a challenge in itself.

Driving a car is a multi-agent, closed-loop control problem without a one-and-only good solution. The behaviour of a car must meet a diverse set of requirements, related to safety, legality, comfort and efficiency in the vastness of road scenarios. Different methods, including hand-written rules, optimization control and data-driven approaches seem to be good ways of tackling those different challenges.

Artificial Intelligence (AI) and Machine Learning (ML) methods are nowadays the foundation of perception systems, including automotive. Most of the applications of those methods are in the form of supervised learning methodology, there machine learning is trained to perform predictions based on labelled datasets. As another branch, Reinforcement Learning (RL) concerns the creation of intelligent agents, capable of making intelligent decisions while interacting with the environment to maximize defined reward signals. As RL from definition aims to solve closed-loop problems, its application to autonomous driving motion planning is a natural research direction.

After analysis of the current state of the art in both autonomous driving and reinforcement learning, the research has been focused on the application of RL methodologies to the problem of motion planning for automated cars. To ensure that the proposed methods would have industrialization potential and might be applied to systems in a real car, attention has been paid to hybrid solutions, which combined RL-based parts with more classical control methods.

3 Research Hypothesis

As the selected research area is relatively new, a general thesis has been formulated as follows:

Research Hypothesis. *The reinforcement learning methodology is applicable to solve the decision-making and trajectory planning problems of autonomous driving vehicles. This statement will be tested and supported by the following claims:*

(i) *Controlling a car with high-level control interface by a reinforcement learning agent is possible.*

(ii) *Introduction of deterministic rules at the time of training improves the training time and the resulting policy.*

(iii) *Controlling the vehicle on a low level with the use of a direct path-planning interface by reinforcement learning agent is possible.*

(iv) *Multi-agent coordination of vehicle scenarios can be solved by reinforcement learning techniques.*

(v) *Making the individual agent's reward dependent on the objectives of other agents improves the overall average performance of all agents.*

A series of experiments has been conducted to test the theorem and its claims. Claims (i) and (ii) have been examined as a part of behaviour planning agent experiments. Claim (iii) has been validated as an outcome of work on problem of parking a car. Claims (iv) and (v) have been evaluated by applying reinforcement learning to the problem of movement coordination of multiple vehicles in urban scenarios.

4 Methods and Results

As mentioned above, to test our general thesis three experiments have been conducted. All of them involved designing not only the policy in the form of the neural network but more importantly creating the whole system architecture, and control concepts and representing specific parts of it both in the form of policy itself as well environment. In all experiments, the Proximal Policy Optimization (PPO) algorithm has been used to train the agent policy.

4.1 Goal-based Behaviour Planning With Manoeuvre and Desired Speed Control

In the first experiment, the focus has been paid to planning the behaviour of an automated car in a highway-like environment. By behaviour, we defined high-level definition of what to do on the road, defined as maneuver to execute and velocity setpoint. The problem has been defined as navigating on lanes of the highway, making sure we arrive at the correct one at specified distance, assuring that along the way that controlled car will optimize its speed and comfort of the drive.

Motivation to use reinforcement learning-based policy arises from multiple factors. First, behaviour planning is not directly responsible for safety, where application of data-driven methods are not well-suited. In the same time, high-level behavior planning is a complex problem with a lot of unobserved states, with necessity for negotiation impacting the selection of the right behavior. Saying that, rule-based methods along with control mechanism brings a lot of useful ways of dealing with speed control, lane keeping and ruling out forbidden maneuvers. Understanding that, research aimed at validating how to design a hybrid system architecture consisting of an Reinforcement Learning based policy, in a way allowing for efficient training and at the same time benefiting from state-of-the-art control mechanism for other parts.

To train reinforcement learning agent, commercially available simulator, TrafficAI, has been adopted to fulfill requirements needed from compute perspective. Based on that, an TrafficAI Environment have been defined, which consisted of simulator itself, trajectory planning block which interpreted RL Agent action in form of the behavior as well as included deterministic rules, and observation creation pipeline, allowing to present the scene to the agent NN in predefined manner.

To fulfill the requirements for efficient RL system training and in the same time assure safety and comfort, series of modifications have been introduced. First, an deterministic available action definition system has been introduced, which based on deterministic rules predefined both maneuvers and velocities available at given moment. This availability information have been injected into the neural network policy, where specific mechanism have been used to guarantee selection of only available actions. Secondly, to stabilize maneuver selection Finite State Machine (FSM) mechanism has been used to assure consistency in action selection and improve transparency. To stabilize the speed control, mechanism allowing for defining delta speed along with deceleration in additive manner to current speed have been introduced. Lastly, predefined maneuver and velocity setpoint have been executed with trajectory planning module, based on classical control method. System operated in Frenet Coordinate System and was relaying on the concept of the Pairwise Const-Jerk Velocity Cruise Algorithm. Additionally, trajectory planning algorithm was working in a ACC

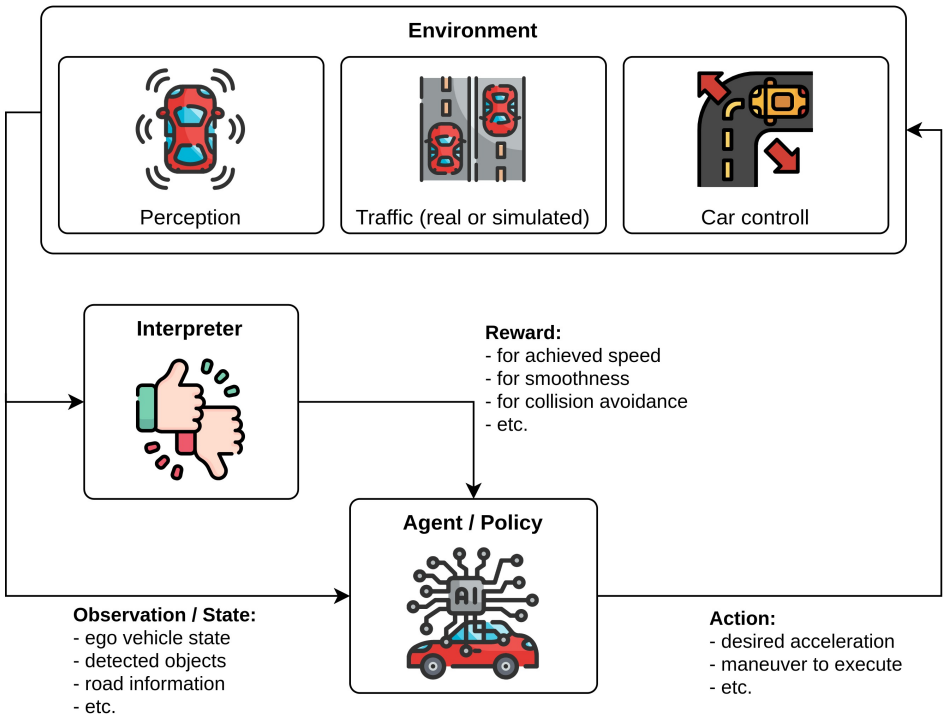


Figure 1: AIPilot TrafficAI Environment architecture with interacting policy.

manner, therefore it was overwriting the request from the RL Agent policy and adapting the velocity to potential car in the front.

The reward signal depended on goal achievement, average speed with respect to speed limit, acceleration and maneuver execution.

To test how proposed system design was working, policy has been trained in two settings. In first, both Maneuver FSM and ACC control has on, which means that the maneuvers availability was controlled by FSM and trajectory planning was adapting its speed to leading cars. In the second setting, both FSM and ACC was off, which gave more direct control over car to RL Policy. Additional setting also has been evaluated, which check how adding mentioned systems after performing training which did not include results in car behavior.

After careful analysis following conclusions might be drawn. From a goal-achievement perspective, the FSM ACC On agent showed the best performance. It has also caused the least number of collisions. FSM ACC Off agent, which had more direct control over the agent speed, minimised the mean absolute acceleration and at the same time presented a slightly better mean velocity. One also argue that the FSM mechanism improves the efficiency of lane change manoeuvres and stabilised the action selection process. A smaller amount of manoeuvre changes and less time spent in lane change, along with a higher probability of reaching the goal, support this claim. Additionally it was proven that adding deterministic rules after the training has a negative effect on the policy performance.

Table 1 KPIs calculated for three experiments with a behavior planner agent.

Experiment Name	**FSM ACC On**	**FSM ACC Off**	**FSM ACC Off in FSM ACC On**
goal reached [%]	**99.2**	98.6	97.8
goal missed [%]	**0.4**	0.6	1.0
collision [%]	**0.4**	0.8	1.2
outside of road [%]	0.0	0.0	0.0
safety violation [%]	1.12	1.59	**0.85**
velocity mean [m/s]	27.21	**27.72**	26.03
velocity std [m/s]	4.55	3.435	4.58
acceleration mean [m/s^2]	1.33	**0.957**	1.3
acceleration std [m/s^2]	1.57	1.23	1.56
manoeuvre change count	2.16	4.02	1.82
follow lane [%]	58.7	24.5	59.6
prepare for lane change left [%]	12.1	1.8	4.33
prepare for lane change right [%]	25.0	63.4	33.2
lane change left [%]	2.34	4.74	1.60
lane change right [%]	1.81	5.6	1.04
abort lane change [%]	0.05	**0.0**	0.2

4.2 Parking

Second experiment involved another essential skill of automated driving portfolio, which is parking a car. In this setting, definition of the objective was much clearer, environment dynamic was much simpler and common application of tree-search methods in parking domain suggested bigger potential for industrialization.

The problem has been formulated as aiming to park a car in designated position, assuring that during motion car will not cause any collision with predefined, stationary obstacles. Depending on the different designs of neural network, obstacles and resulting freespace around the car have been encoded in a different format. Action space have been the same in all experiments, and was a discrete set of combination of wheel angle and movement, both forward and backward. Obstacles have been defined in the form of polygons. Reward function was sparse, and depend on the goal achievement (+1), eventual collision (0) and number of direction changes (-0.1 for each).

Experiments have been conducted with two different neural network architectures. One have accepted encoded freespace around controlled vehicle in the form of evenly spreader rays of different lengths (Figure 2. Second used the concept of graphs neural networks, and encoded the obstacles in their natural form, as a set of encoded vertexes of each obstacle - Figure 3. An curriculum learning mechanism have been introduced, which gradually increased the difficulty of the scenarios along the training course. This process included gradual shrinking of spot and goal tolerance. This allowed effective start of the training as well as further improvements of in most difficult scenarios.

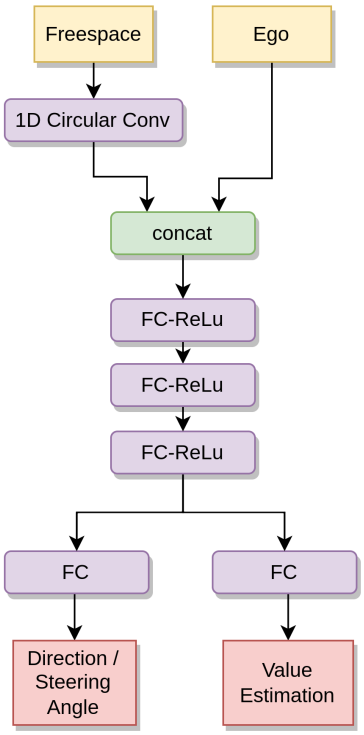


Figure 2: Freespace neural network

Conducted experiments and further evaluation resulted in following conclusions. It has been showed that inference and training of graph neural network is significantly slower than of freespace version, but in the same time training of graph NN is more sample efficient. The functional evaluation in most of the cases ended in slightly better performance of graph neural network. On the other hand, the freespace NN showed greater robustness to out of distribution scenarios.

An open loop evaluation have been as well executed in a test car. With changes introduced to training process and parsing of occupancy grid detected with use of radar sensors, real-time potential of freespace NN solution has been proven.

Table 2 KPIs measures for parking agents trained with two kinds of neural network - Graph neural network and Freespace neural network, calculated for three kind of scenarios.

	All scenarios		Parallel		Perpendicular		At Angle	
	Graph	Freespace	Graph	Freespace	Graph	Freespace	Graph	Freespace
samples number	5000	5000	1648	1680	1717	1683	1635	1637
goal reached [%]	**98.46**	98.26	**99.27**	98.3	98.19	**98.57**	**97.92**	97.86
in collision [%]	**1.44**	2.1	**0.73**	1.67	1.63	**1.42**	**1.96**	3.23
average path length [m]	**14.24**	14.53	**14.88**	15.45	14.43	**14.26**	**13.4**	13.71
average episode length	**16.6**	17.33	**17.2**	18.93	**16.85**	16.89	**15.7**	16.13
average direction changes	1.5	**1.42**	**1.11**	1.17	1.66	**1.55**	1.73	**1.54**

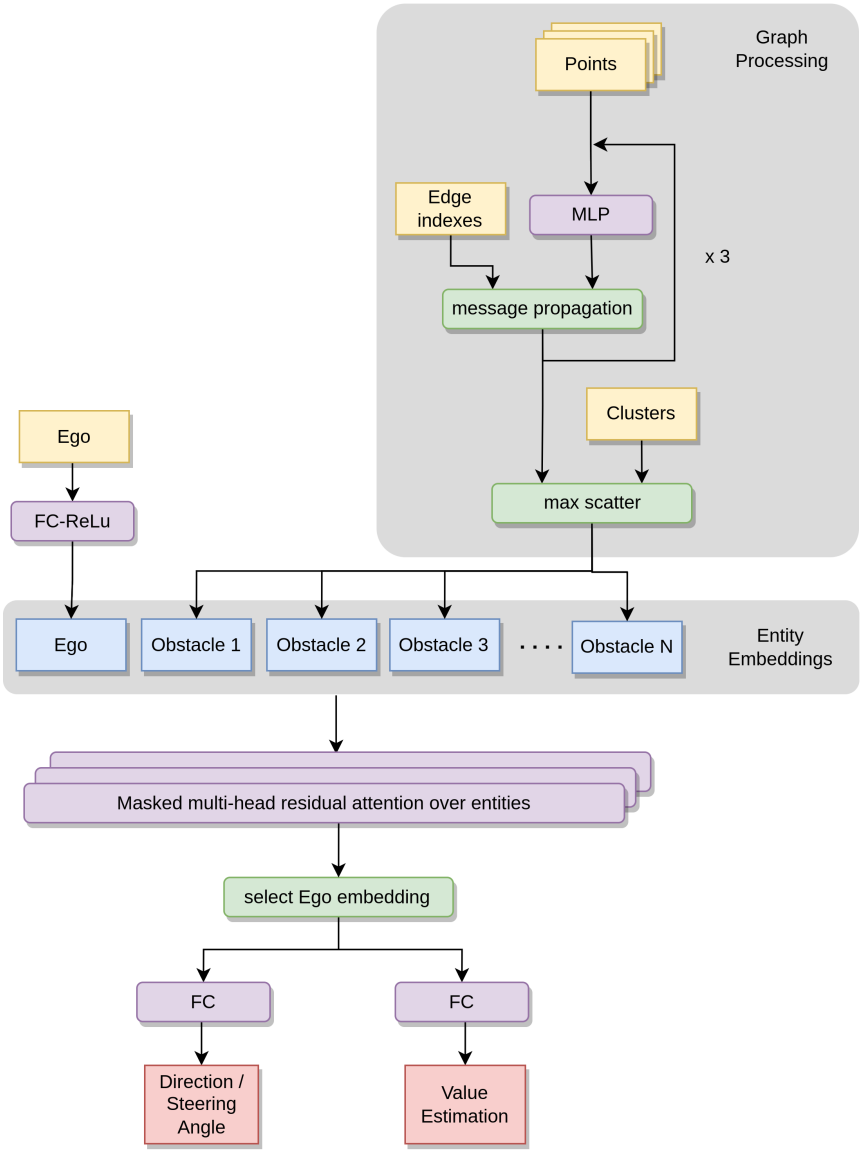
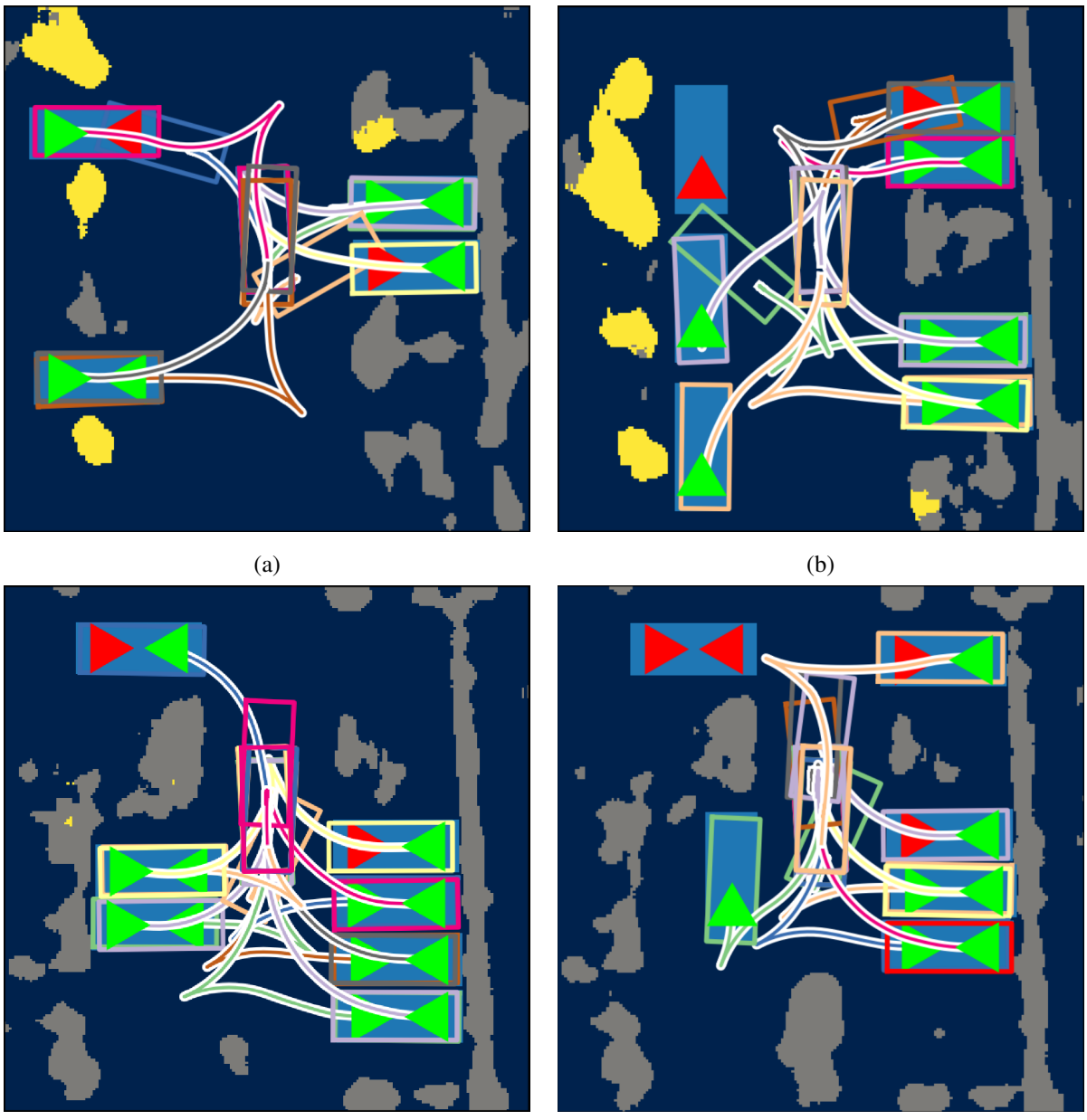


Figure 3: Graph neural network



(a) (b)

(c) (d)

Figure 4: Examples of paths found by RL-based policy in real-world data. Both yellow and grey areas indicate obstacles, while dark blue colour represents the free space. Rectangles of different colours, along with corresponding paths, represent multiple agents with their corresponding parking spots in light blue. Arrows represent the target position, with green indicating successful parking of agent in a given spot, and red indicating failure in doing so.

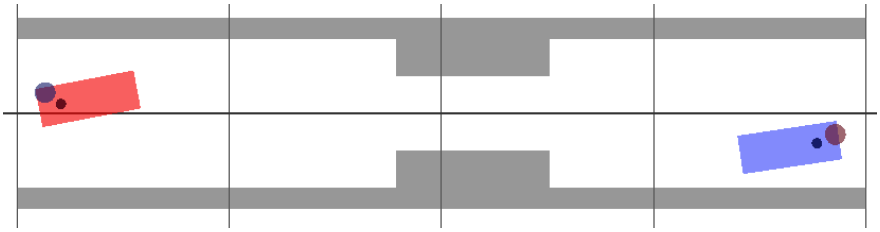


Figure 5: The bottleneck scenario with a centrally placed bottleneck.

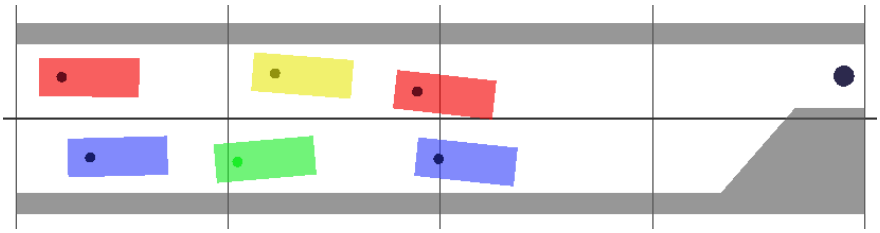


Figure 6: The zipper scenario with the narrowing located on the left side of the road.

4.3 Multi-agent Maneuvering

In the last experiment, the aspect of cooperation with other road users has been tackled, which is inextricably linked with driving. In a lot of the cases, those interaction are clearly codified by traffic rules or can be seen as a response of a one vehicle to actions of the other. Still, there are situations in which vehicles needs to cooperate between each other in uncodified manner.

In this part, multi-agent reinforcement learning methods has been applied to challenging on-road scenarios that require extensive cooperation between the road users. To do so, parking environment introduced in previous experiment has been extended to handle multiple agents. In this environment, multiple agents has been simulated at once, each of them with their own goal.

With the environment described above, three families of scenarios has been simulated, including bottleneck, zipper and crossroad scenario.

The observation space of the environment included freespace measurement, but additionally encoded each of the vehicles state, including ego. As the important part of the on-road cooperation is understanding intentions of other road users by observing their motion profile, motion model of the vehicle has been adopted to one which tracked the velocity. With that, RL policy action space has been designed as discrete set of combinations of different accelerations and turn angles.

Main experiments focused on evaluating different reward applied. In the most simple setting, reward was non-zero only at the end of episode when given agent arrived at goal position. With this reward formulation, all above settings has been solved by the agent setups. In the same time it turned out, that those environments are highly cooperative ones.

To introduce a small incentive for competitiveness, reward based on the average velocity of each agent has been used. With that, analysis of the reward sharing concept has been investigated, where each of the agent's reward is based on its own performance, but as well on the performance of other agents.

The results of the evaluation has been presented in Table 3.

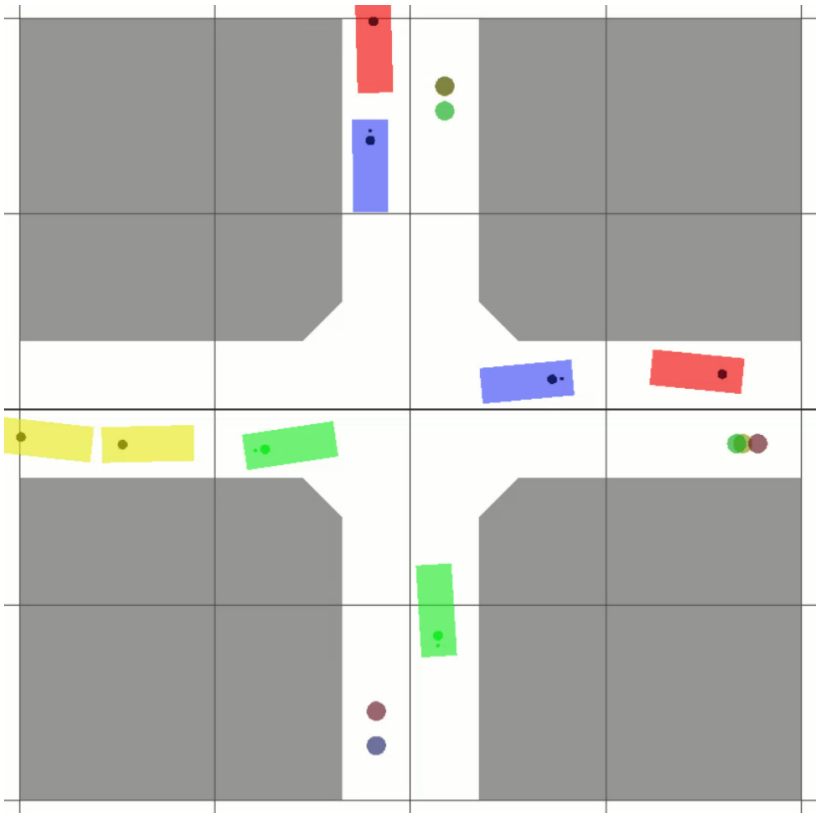


Figure 7: The crossroad scenario, with multiple agents each aiming at a different end goal, which is color-coded.

Table 3 Table presents performance evaluation for three crossroads setups: Baseline, with reward not taking into consideration time, Timed with such incentive and Timed with shared reward, where additionally performance of all agents has been shared. The values which relate to episode duration, speed, and acceleration have been only calculated for agents successfully arriving at the destination.

	Baseline	**Timed**	**Timed with Reward Sharing**
Goal reached [%]	99.5	96.9	97.65
Obstacle collision [%]	0.12	0.43	0.24
Agent collision [%]	0.32	2.73	2.16
Avg episode length	31.08	23.33	22.86
Avg speed [$\mathrm{m/s}$]	1.9	2.566	2.584
Max speed [$\mathrm{m/s}$]	3.41	5.21	5.761
Min speed [$\mathrm{m/s}$]	0.52	1.01	1.032
Static in episode [%]	13.23	6.14	6.34
Avg sum acc [$\mathrm{m/s^2}$]	20.64	18.58	18.27
Std sum acc [$\mathrm{m/s^2}$]	0.76	0.856	0.855

The above experiments prove that with straightforward problem formulation it is possible to acquire policies performing well in road scenarios that require a lot of cooperation between road users. Cooperation seems to be a natural strategy for all the policies, as the collision has equally detrimental effect on all its participants. In all experiments, all behaviors seemed very human like even thought no direct incentive have been applied to achieve this. It also have been proven that with time incentive rewards the reward sharing mechanism improves both functional performance as well as training efficiency.

5 Summary and Contribution

Based on the experiments performed in different domains, the conclusion can be drawn that the reinforcement learning methodology is an attractive and reasonable alternative to the standard control methods used so far in the autonomous driving domain.

Following key contributions of this work might be listed:

- It has been proven that a simulated autonomous car can be controlled by a high-level interface, such as a behaviour planning one, with the use of reinforcement learning methodology, which supports claim (i).
- Research showed that the introduction of a proposed deterministic mechanism during training, including a finite-state machine manoeuvre, available action predefinition mechanism, and trajectory generation, results in better end performance and faster convergence compared to doing so after training. This supports claim (ii).
- It has been confirmed that introduction of deterministic rules decreases the transparency of the system from reinforcement learning agent perspective, therefore, such integration needs to be done with care.
- As part of a collaborative effort, a traffic simulator applicable for the autonomous driving reinforcement learning application has been created.
- It has been proven that controlling a vehicle in parking scenarios with a low-level control interface by reinforcement learning policies is possible, supporting claim (iii).
- A comparison of two observation models and associated neural network architectures with them has been made, showing the benefits and drawbacks of both solutions.
- The integration of the RL-based parking application within the car test system has been carried out, proving the real-time potential of the proposed solution and the applicability to real-world data.
- The reinforcement learning methodology has been successfully used to coordinate the movement of multiple agents in city-like scenarios, which required careful coordination of all agents' behaviour, supporting claim (iv).
- The proposed reward sharing mechanism with a straightforward definition of the reward function improved the effectiveness of training and the resulting functional performance for all agents, which proved claim (v).

- The research carried out resulted in the implementation of a set of reinforcement learning environments which can be further customised and reused to perform new experiments.

To summarise, author argues that reinforcement learning methodologies are applicable to autonomous driving problem, however, due to high complexity and safety requirements, introduction of them to real-world system will require further development, careful analysis and validation accordingly to all automotive standards.