IET Intelligent Transport Systems

Special issue Call for Papers

Be Seen. Be Cited.
Submit your work to a new IET special issue

Connect with researchers and experts in your field and share knowledge.

Be part of the latest research trends, faster.

Read more

The Institution of Engineering and Technology

IET Intelligent Transport Systems

Research Article

IET Journals
The Institution of Engineering and Technology

Deep reinforcement-learning-based driving policy for autonomous road vehicles

ISSN 1751-956X
Received on 26th April 2019
Revised 8th August 2019
Accepted on 11th October 2019
E-First on 7th November 2019
doi: 10.1049/iet-its.2019.0249
www.ietdl.org

Konstantinos Makantasis[1,2] ✉, *Maria Kontorinaki*[1,3], *Ioannis Nikolos*[1]

[1]*School of Production Engineering and Management, Technical University of Crete, Chania, Greece*
[2]*Institute of Digital Games, University of Malta, Msida, Malta*
[3]*Department of Statistics and Operations Research, Faculty of Science, University of Malta, Msida, Malta*
✉ *E-mail: konst.makantasis@gmail.com*

Abstract: In this work, the problem of path planning for an autonomous vehicle that moves on a freeway is considered. The most common approaches that are used to address this problem are based on optimal control methods, which make assumptions about the model of the environment and the system dynamics. On the contrary, this work proposes the development of a driving policy based on reinforcement learning. In this way, the proposed driving policy makes minimal or no assumptions about the environment, since a priori knowledge about the system dynamics is not required. Driving scenarios where the road is occupied both by autonomous and manual driving vehicles are considered. To the best of the authors' knowledge, this is one of the first approaches that propose a reinforcement learning driving policy for mixed driving environments. The derived reinforcement learning policy, firstly, is compared against an optimal policy derived via dynamic programming, and, secondly, its efficiency is evaluated under realistic scenarios generated by the established SUMO microscopic traffic flow simulator. Finally, some initial results regarding the effect of autonomous vehicles' behaviour on the overall traffic flow are presented.

1 Introduction

In recent years, there has been a growing interest in self-driving vehicles. Building such autonomous systems has been an active area of research [1, 2] for its high potential in leading to road networks that are much more safer and efficient. Although vehicle automation has already led to great achievements in supporting the driver in various monotonous and challenging tasks, see, e.g. [3], rising the level of automation to fully-automated driving is an extremely challenging problem. This is mainly due to the complexity of real-world environments, including avoiding obstacles, and human driving behaviour aspects.

According to Donges [4], autonomous driving tasks can be roughly classified into three categories; navigation, guidance, and stabilisation. Navigation tasks are responsible for generating road-level routes. Tactical-level guidance tasks are responsible for guiding autonomous vehicles along these routes in complex environments by generating tactical manoeuvre decisions. Finally, operational-level stabilisation tasks are responsible for translating tactical decisions into reference trajectories and then low-level controls that need to be tracked by the vehicle.

Several methodologies have been proposed for addressing the problem of generating efficient road-level routes. In [5, 6] Menelaou *et al.* propose a navigation algorithm based on a route reservation mechanism, in order to generate road-level routes, and at the same time, avoid traffic congestion. In [7] a computationally efficient algorithm that can scale to very large transportation networks is presented. Figliozzi [8] focuses on 'green' navigation by proposing a methodology to generate routes that minimise emissions. The work in [9] surveys exact algorithms for addressing the routing problem under capacity and travel time constraints. Finally, a comprehensive review regarding the generation of road-level routes in transportation networks can be found in [10]. Despite the research interest, navigation examined by the autonomous driving perspective can be considered as a mature technology, since already existing commercial and free applications for road-level route generation.

At the same time, vehicles are man-made products for which the automotive industry has decades-long experience in vehicle dynamics modelling, see, e.g. [11, 12]. Therefore, the operational-level stabilisation tasks, also known as the *acting* the part of autonomous driving, are well understood and modelled in control theory and robotics.

Tactical-level guidance, referred also as *driving policy*, is crucial for enabling fully autonomous driving. On the contrary, however, to navigation and stabilisation, tactical-level guidance methodologies cannot be considered mature enough, in order to be applied to autonomous vehicles that move in unrestricted environments. A driving policy should be able to make decisions in real-time and in complex environments, in order to plan and update a vehicle path, which should be safe, collision-free, and user acceptable [13]. The requirement for real-time operation in highly complex environments, as well as the safety constraints, makes current driving policies inadequate for fully autonomous driving.

Based on the discussion so far, this work aims to contribute towards the development of a robust driving policy for autonomous vehicles that is capable of making decisions in real time. The operation environments restricted by considering vehicles that move on a highway. Highways consist a specific and very important type of transportation network [14, 15]. Due to their high-capacity, they can serve millions of people every day, while at the same time, allow users to travel with higher speed and fewer accelerations/decelerations compared to urban transportation networks. The driving policy development problem is formulated from an autonomous vehicle perspective (ego vehicle), and, thus, there is no need to make any assumptions regarding the kind of other vehicles (manual driving or autonomous) that occupy the road.

Finally, the proposed methodology approaches the problem of driving policy development by exploiting recent advances in *Reinforcement Learning* (RL) combined with the responsibility sensitive safety model, proposed in [16]. The developed RL-based driving policy aims to avoid accidents (departures from the road and crashes with other vehicles), move the vehicle with the desired speed, minimise accelerations/decelerations, and minimise lane changes. The latter two criteria are also related to the comfort of vehicle passengers [17].

1.1 Related work

The problem of path planning for autonomous vehicles can be seen as a trajectory generation problem corresponding to the creation of a quasi-continuous sequence of states that must be tracked by the vehicle over a specific time horizon. Trajectory generation has been widely studied in robotics [18]. Considering, however, road vehicles, path planning is a much more critical task, since passengers' safety must be guaranteed.

Under certain assumptions, simplifications, and conservative estimates, heuristic, hand-designed rules can be used for tactical decision making [19]. Such methods, however, are often tailored for specific non-complex environments and do not generalise robustly [20]. Therefore, they are not able to cope with the complexity of real-world environments and the diversity of driving conditions, let alone human driving behaviour aspects. To overcome the limitations of rule-based methods, approaches based on the careful design and exploitation of *potential field* and *optimal control* methods have also been proposed.

Potential field methods generate a field, or, in other words, an objective function the minimisation of which corresponds to the objectives of an agent. These methods are based on the design of potential functions for obstacles, road structures, traffic regulation and the goals to be achieved. Then, the overall objective function is expressed as the weighted sum of the designed potential functions. The minimisation is achieved via the generation of a vehicle trajectory moving towards the descent direction of the overall objective function [21–23]. However, due to the fact that vehicle dynamics are not considered during decision making, the generated trajectory may turn out to be non-feasible to be tracked by the vehicle [24].

This drawback can be alleviated by formulating the trajectory generation problem as an optimal control problem, which inherently takes into consideration system dynamics. Specifically, optimal control approaches allow for the *concurrent* consideration of system dynamics and carefully designed potential fields [25]. In [17] an optimal control methodology for vehicles' trajectory planning in the context of cooperative merging on highways is presented. The authors of [26, 27] propose two optimal control-based methodologies for trajectory planning, which incorporate constraints for obstacles, so as to keep the automated vehicle robustly far from them. In the same spirit, the authors of [28, 29] design appropriate potential functions corresponding to the presence of obstacles, which, in turn, are incorporated in the objective function to generate a collision-free path. Optimal control approaches usually map the optimal control problem to a non-linear programming (NLP) problem that can be solved using numerical NLP solvers, see, e.g. [28, 30, 31]. Although potential field and optimal control methods are quite popular due to the intuitive problem formulation [32], there are still open issues regarding the decision making process.

First of all, mapping the optimal control problem to an NLP problem and solving it by employing numerical NLP solvers, produces a locally optimal solution for which the guarantees of the globally optimal solution may not hold, and, thus, the safety guarantees for the generated trajectory may be compromised [33]. For this reason, dynamic programming techniques have also been proposed for solving the optimal control problem. Although, dynamic programming techniques produce a globally optimal solution, due to the *curse of dimensionality* [34], they are restricted to small-scale problems. Moreover, another problem faced with potential field and optimal control approaches is the strong dependency on a relatively simple environment model, usually with hand-crafted observation spaces, transition dynamics, and measurement mechanisms. These assumptions limit the generality of these methods to complex scenarios since they are not able to cope with environment uncertainties and measurement errors. Finally, optimal control methods are not able to generalise, i.e. to associate a state of the environment with a decision without solving an optimal control problem. This means that every time a sequence of decisions needs to be made an optimal control problem needs to be solved, even if exactly the same problem has been solved in the past. This requirement significantly increases the computational cost of these methods.

Due to its recent success, supervised deep learning has also been considered as an alternative approach for developing driving policies. In [35] a convolutional neural network is trained in a supervised manner to output continuous steering actions. In [36] a recurrent neural network is trained to output a steering angle after a driving intention has been estimated. In [37, 38] also exploit end-to-end trainable neural networks that output feasible driving actions and affordance indicators (such as distance between cars). The aforementioned approaches are based on end-to-end trainable neural network architectures that are able to output low-level controls directly from input images. Therefore, this kind of driving policy corresponds to the outcome of a supervised learning algorithm, where deep neural networks were trained to imitate the behaviour of human drivers. However, such methods, first, result in black-box driving policies, which are susceptible to the influence of drifted inputs, and second, are restricted to the limitations of end-to-end learning [39].

Very recently, RL methods have also been proposed as challenging alternative approaches towards the development of driving policies. RL-based approaches alleviate the strong dependency on hand-crafted simple environment models and dynamics, and, at the same time, can fully exploit the recent advances in deep supervised machine learning [40]. Along this line of research, Isele *et al.* [41] utilise a deep Q-network to make decisions for intersection crossing, while Mukadam *et al.* [42] exploit a similar architecture to make decisions about lane changing in freeways. In [43], Paxton *et al.* propose a hierarchical RL-based approach for deriving a low-level driving policy capable of guiding a vehicle from an origin point to a destination point. In [44] a policy gradient RL approach is used to develop a driving policy for cooperative double merging scenarios. This approach combines an RL policy with a non-learnable mechanism to balance between efficiency and safety. Finally, Liu *et al.* [45] present some elements of efficient deep RL (empirically validated) for decreasing the learning time and increasing the efficiency of RL-based driving policies.

Despite the fact that only very recently RL was employed for developing driving policies, experimental results appear very promising. The main drawback, however, of these approaches regards safety guarantees. Due to the fact that, the probability of an accident is very small, learning-based approaches, as shown in [16], cannot assure collision-free trajectories.

1.2 Proposed work

This work proposes an RL-based approach towards the development of a driving policy for autonomous road vehicles. The proposed RL-based method has several advantages over potential field and optimal control methods. First, RL-based approaches are *model-free*. They make the assumption that there is a state-transition model that describes the system dynamics, which remains fixed. However, the exact form of this model is not required to be a priori known (typically such a model is considered unknown), but it is being inferred during training. Second, a driving policy based on the RL is able to *generalise*. After training, an RL-based policy has inferred a mapping for associating a given state of the environment with a decision. In contrast to potential field and optimal control methods, whenever a decision needs to be made no problem needs to be solved; decision making can be done by simply evaluating the policy function. Third, since an RL-based driving policy has been estimated, it can be shared across multiple autonomous vehicles, which in turn can make decisions through the policy function evaluations. On the contrary, driving policy sharing is not possible when potential field and optimal control methods are used, since each vehicle needs to solve a decision-making problem for its own sake. Finally, since no learning-based driving policy can guarantee absolute safety, our work is motivated by the formal responsibility sensitive safety model, proposed in [16], in order to derive and utilise *ad-hoc* rules that guarantee responsibility-wise safety. That is, the *ad-hoc* rules guarantee that the autonomous vehicles will not be responsible for any occurred accident. To the best of the authors' knowledge, this work is one of the first attempts that try to derive an RL driving policy, combined

with *ad-hoc* safety rules, targeting unrestricted highway environments, which are occupied by both autonomous and manual driving vehicles.

Furthermore, the proposed RL-based driving policy is compared against an optimal policy derived using dynamic programming, in terms of safety metrics, such as the number of collisions, and efficiency metrics, such as the average time the autonomous vehicle moves with the desired speed. Although, dynamic programming techniques, due to the curse of dimensionality [46], are restricted to small-scale problems, and are not suitable for real-time applications, they produce globally optimal solutions to an optimal control problem, i.e. optimal driving policies. Thus, the comparison of the proposed methodology against optimal driving policies, first, will result in an objective evaluation for the RL-based driving policy, and, second, can provide insights into the driving policy development problem.

The developed RL-based driving policy is also compared against manual driving using SUMO simulator. Through this comparison, the generalisation ability and stability of the proposed RL-based driving policy to ensure reliability is evaluated; any learning system must generalise well to out-of-sample data, and be stable, i.e. small perturbations in the input should slightly affect the output. Specifically, the RL-based driving policy is applied to randomly generate driving scenarios (previously unseen driving conditions), with and without drivers' imperfection and measurement errors. Drivers' imperfection and measurement errors can be seen as disturbances, and can be incorporated into driving scenarios using appropriate settings in SUMO simulator.

Finally, preliminary results regarding the effect of autonomous vehicles on the overall traffic flow are provided. The RL-based driving policy, seen by an autonomous vehicle perspective, is a selfish policy. That is, each autonomous vehicle that follows the RL policy tries to achieve its own goals disregarding the rest of the vehicles. Such behaviour might have a negative effect on the overall traffic flow.

The rest of the paper is organised as follows: Section 2 describes the problem and the underlying assumptions. Section 3 gives a brief description of the RL framework. Section 3 presents in detail the development of the RL-based driving policy and in Section 5, the derivation of *ad-hoc* rules towards the design of a collision-free trajectory. Section 6 presents the experimental setup and the experimental results, and Section 8 concludes this work.

2 Problem description and assumptions

The problem of path planning for an autonomous vehicle that moves on the freeway, which is also occupied by manual driving vehicles, is considered. Without loss of generality, it is assumed that the freeway consists of three lanes. The path planning algorithm, or in other words, the driving policy, should generate a collision-free trajectory for the autonomous vehicle to follow. Moreover, the generated trajectory should permit the autonomous vehicle to move forward with the desired speed, and, at the same time, minimise its longitudinal and lateral accelerations/decelerations. The aforementioned three criteria are the objectives of the driving policy, and therefore, the goal that the RL algorithm should achieve.

For the generation of an optimal trajectory using dynamic programming, the manual driving vehicles are required to move with a constant speed following the kinematics equations. The generation of the optimal trajectory, via dynamic programming, corresponds to the solution of a finite horizon optimal control problem. The aforementioned requirement assures that the dynamics of the system will be a priori and fully known, and no disturbances will be present in the system in order for the dynamic programming technique to produce the trajectory. However, for training the RL policy, the aforementioned system dynamics are not given to the algorithm, and, thus, are considered unknown.

Regarding the SUMO simulator, the manual driving vehicles move on the freeway using the Krauss car following model [47]. It is assumed that letting the manual driving vehicles to move using the Krauss car following model will produce realistic driving behaviours. Moreover, manual driving vehicles should move forward with the desired speed. In order to generate realistic and customary traffic conditions, we assume that at least two categories of manual driving vehicles should be present at the freeway; manual driving vehicles that want to move faster than the autonomous vehicle, and manual driving vehicles that want to move slower. At this point, it should be stressed that, although the manual vehicles are moving using the Krauss model, this model is not given to the RL training algorithm, and, thus, from an RL point of view it is considered unknown.

During the trajectory generation, this work does not assume any communication between the autonomous vehicle and other vehicles. Instead, the information available for the trajectory generation is obtained solely by sensors, such as cameras, LiDAR and proximity sensors, installed on the autonomous vehicle. This work also assumes the availability of a fusion module of the on-board sensors' information, with the appropriate redundancy and cross-checking, to assure the usefulness and accuracy of the provided information. Using such sensors, the autonomous vehicle can estimate the position and the velocity of its surrounding vehicles. Therefore, the state representation of the autonomous vehicle and its surrounding environment, includes information that is associated solely with the position and the velocity of the vehicles present in the sensing area of the autonomous vehicle.

Furthermore, it is assumed that the freeway does not contain any turns. However, the generated vehicle trajectory essentially reflects the vehicle longitudinal position, speed, and its travelling lane. The derived trajectory needs to be tracked by the underlying vehicle control loops based on high-definition maps. Therefore, for the trajectory specification, possible curvatures may be aligned to form an equivalent straight section [28].

Finally, the trajectory of the autonomous vehicle can be fully described by a sequence of goals that the vehicle should achieve. Each one of the goals should be achieved within a specific time interval, and represents vehicle's desires, such as change lane, brake with a given deceleration, etc. These goals define the trajectory to be followed by the autonomous vehicle at a higher level, and cannot be directly used by the vehicle control loops. Instead, it is assumed that the mechanism which translates these goals to low-level controls and implements them is given.

Based on the aforementioned problem description and underlying assumptions, the main objective of this work is to develop a driving policy. The driving policy will exploit the information coming from a set of sensors installed on the autonomous vehicle, in order to set a goal for the vehicle to achieve, via a high-level action, during a specific time interval. In other words, the objective is to derive a function that will map the information about the autonomous vehicle, as well as its surrounding environment to a specific goal and the corresponding high-level action for achieving it.

3 RL and prioritised experience replay (PER)

In this work, the development of a driving policy is being tackled as an RL problem, where the state-action value function Q is approximated by a double deep Q-network (DDQN) [48] using PER [49]. Therefore, for the sake of completeness, in this section the RL framework and the algorithm of PER are briefly presented.

3.1 Reinforcement learning

In the RL framework, an agent interacts with the environment in a sequence of actions (selected by following a specific policy), observations, and rewards. In particular, at each time step t, the agent (in our case the autonomous vehicle) observes the state of the environment $s_t \in \mathcal{S}$ and, based on a specific policy, it selects an action $a_t \in \mathcal{A}$, where $\mathcal{S}$ is the state space and $\mathcal{A} = \{1, \ldots, K\}$ is the set of available actions. Then, the agent observes the new state of the environment, s_{t+1}, which is the consequence of applying the action a_t at state s_t, and a scalar reward signal r_t, which is a quality measure of how good is to select action a_t at state s_t.

The goal of the agent is to interact with the environment by selecting actions in a way that maximises the cumulative future rewards, also known as a *future return*. Future rewards are

discounted by a factor $0 \leq \gamma < 1$ per time step, and the future return at time t is defined as

$$R_t = \sum_{t'=t}^{T} \gamma^{t'-t} r_{t'}, \tag{1}$$

where parameter T denotes how many time steps ahead t are taken into consideration for calculating R_t. The non-negative discount factor $0 \leq \gamma < 1$ determines the importance of future rewards. In other words, it weighs future rewards by giving higher weight to rewards received near rewards received further in the future.

The interaction of the agent with the environment can be explicitly defined by a policy function $\pi: \mathcal{S} \rightarrow \mathcal{A}$ that maps states to actions. The maximum expected future reward achievable by following any policy after observing a state s and selecting an action a is represented by the optimal *action-value* function $Q^*(s, a)$, which is defined as

$$Q^*(s, a) = \max_{\pi} \mathbb{E}[R_t \Big| s_t = s, a_t = a, \pi]\,. \tag{2}$$

The optimal action-value function obeys a very important identity known as the *Bellman equation*. That is, if the optimal action-value function $Q^*(s_{t+1}, a_{t+1})$ of the state s_{t+1} at the next time step was known for all possible actions a_{t+1}, then the policy maximising the future reward is to select the action maximising the expected value of $r + \gamma Q^*(s_{t+1}, a_{t+1})$, and, thus, the following

$$Q^*(s, a) = \mathbb{E}_{s_{t+1}}\left[r_t + \gamma \max_{a_{t+1}} Q^*(s_{t+1}, a_{t+1}) \Big| s_t = s, a_t = a \right] \tag{3}$$

holds for the optimal action-value function when the state s_t is observed and action a_t is selected. The expectation in relation (3) is with respect to all possible states at the next time step.

The relation in (3) implies that the problem of estimating the optimal policy is equivalent to the estimation of $Q^*(s, a)$ for every pair $(s, a) \in \mathcal{S} \times \mathcal{A}$. Although, $Q^*(s, a)$ can be efficiently estimated when small scale problems need to be addressed [50], for large state spaces estimating $Q^*(s, a)$ for every possible (s, a) pair is practically implausible. For such kinds of problems, the optimal action-value function is approximated, $\tilde{Q}^*(s, a; \theta) \simeq Q^*(s, a)$, using a learning machine, such as linear regression or neural networks [40], parameterised by θ. Parameters θ are estimated by the following an iterative procedure for minimising a sequence of loss functions:

$$L_i(\theta_i) = \mathbb{E}_{s,a}[(\tilde{Q}(s, a; \theta_{i-1}) - \tilde{Q}(s, a; \theta_i))^2], \tag{4}$$

where i stands for the iteration index. The (s, a, r, s') tuples used in relations (3) and (4) are generated by following an ϵ-greedy policy that selects at a given state a greedy action with probability $1 - \epsilon$ and a random action with probability ϵ.

The aforementioned procedure for estimating θ looks like a regression problem in the supervised learning paradigm. However, there are two significant differences. First, the learning machine sets itself and follows the targets $\tilde{Q}(s, a; \theta_{i-1})$, which can lead to instabilities and divergence, and, second, the generated (s, a, r, s') tuples are not independently generated; a property that is required by many learning machines.

To overcome the first problem, two identical learning machines are used; one for setting the targets and one for following them. The machine that a set the targets are frozen in time, i.e. its parameters are fixed for several iterations. After a predefined number of iterations has passed, the parameters of the machine that sets the targets are updated by copying the parameters from the machine that follows the targets. If we denote as $\hat{\theta}$ the parameters of the machine that sets the targets, then the loss function $L_i(\theta_i)$ in relation (4) is given by

$$L_i(\theta_i) = \mathbb{E}_{s,a}[(\tilde{Q}(s, a; \hat{\theta}) - \tilde{Q}(s, a; \theta_i))^2]\,. \tag{5}$$

3.2 PER algorithm

To overcome the latter problem, a PER algorithm is employed to break the correlations between the generated (s, a, r, s') tuples. The generated tuples are stored into a memory, and for minimising (4), a training set $\mathcal{D} = \{(a, s, r, s')_j\}_{j=1}^{n}$ is drawn from the memory according to a distribution that prefers tuples that do not fit well to the current estimate of the action-value function.

For estimating the sampling distribution, initially, the difference

$$d(s, a, r, s') = \left| \tilde{Q}(s, a; \hat{\theta}) - \tilde{Q}(s, a; \theta_i) \right| \tag{6}$$

is computed for each tuple in memory and is updated after each iteration i. Then, the difference is converted to priority

$$p = (d - \epsilon)^a, \tag{7}$$

with $\epsilon > 0$ to ensure that no tuple has zero probability of being drawn, and $0 \leq a < 1$ (when $a = 0$ the uniform distribution over tuples is used). Finally, the priorities are translated into probabilities. In particular, a tuple k has a probability

$$P_k = \frac{p_k}{\sum_{j=1}^{N} p_j} \tag{8}$$

of being drawn during the experience replay. Variable N in (8) stands for the cardinality of memory.

4 Driving policy

Having described the RL framework and the PER algorithm, in this section, the RL-based approach utilised in this work is presented, along with the state and action representation, and the design of the scalar reward signal. Finally, the architecture of the employed neural network, and details about the implementation, as well as the mechanism for generating (s, a, r, s') tuples for training the neural network are described.

4.1 State representation

Autonomous vehicles are equipped with multiple sensors that enable them to capture heterogeneous and multimodal information about their surrounding environment. This allows for a wide variety of state representations. The selection, however, of the representation significantly affects the ability of an agent to learn. In this work, a state representation that, on the one hand, can be constructed using current sensing technologies, and, on the other, it allows the agent to efficiently learn is utilised.

Specifically, this work considers autonomous vehicles that move on a freeway with three lanes. It is assumed that the vehicle can sense the surrounding environment that spans 60 m behind it and 100 m ahead of it, as well as, its two adjacent lanes. This means that the autonomous vehicle can estimate the relative positions and velocities of other vehicles that are present in the aforementioned area. Note that with current LiDAR and camera sensing technologies, such an assumption can be considered valid. A schematic representation of the sensed surrounding environment of a vehicle is presented in Fig. 1*a*.

In order to translate the information that can be sensed by the autonomous vehicle into a state vector, the sensed area is discretised into tiles of 1 m length, as shown in Fig. 1*b*. In order for this discretisation to be useful, the accuracy of the vehicle sensors must be in the order of centimetres, something that is feasible with current sensing technologies [51–53]. The value of the longitudinal velocity of the autonomous vehicle is assigned to the tiles beneath it. To tiles occupied by other vehicles the value of their longitudinal velocity is assigned. The velocity of the other vehicles is estimated by using their positions in two subsequent time instances. The value of zero is given to all non-occupied tiles that belong to the road and, finally, the value minus one to tiles outside

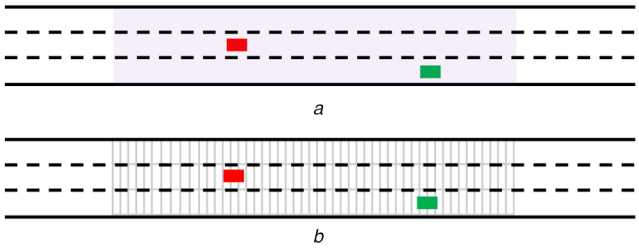

Fig. 1 *State representation. The autonomous vehicle is represented by the red rectangle, while the green rectangle represents another vehicle present on the road*

(a) Purple shaded area corresponds to the sensed surrounding environment of the autonomous vehicle, ***(b)*** Discretisation of the sensed environment

Table 1 Available actions of the autonomous vehicle

Action #1:	change lane to the left
Action #2:	change lane to the right
Action #3:	constant acceleration of $1\,\mathrm{m/s^2}$
Action #4:	constant acceleration of $2\,\mathrm{m/s^2}$
Action #5:	constant deceleration of $1\,\mathrm{m/s^2}$
Action #6:	constant deceleration of $2\,\mathrm{m/s^2}$
Action #7:	move on current lane with current speed

of the road (the autonomous vehicle can sense an area outside of the road in case it occupies the left-most or the right-most lane).

Using the representation above, the sensed environment is transformed into a matrix with 3 rows and 160 columns. Moreover, this matrix contains information about the absolute velocities of vehicles, as well as relative positions of other vehicles with respect to the autonomous vehicle. Finally, the vectorised version of this matrix, which is a vector with 480 elements, is used to represent the state of the environment at each specific time step.

The proposed state representation can be easily obtained by sensors installed on an autonomous vehicle. Despite the fact that this representation is relatively simple, as can be seen in Section 6, it contains adequate information for obtaining robust driving policies. More realistic and informative state representations can be constructed. For example current pattern and object recognition methods can be utilised to classify vehicles and, thus, incorporate into the state representation information regarding the type of surrounding vehicles and their size. In addition, if we assume that vehicles are equipped with communication enabling technologies, then vehicle-to-vehicle communication can be used to enhance the state representation with information regarding vehicles' longitudinal and lateral accelerations, while vehicle-to-infrastructure communication can provide information regarding the state of the network. Although, more accurate state representations can be constructed, using a simple state representation, like the proposed one, permits to gain insights with respect to the behaviour of the derived policy. Moreover, we deliberately do not assume any communication between the vehicles, to make the training of the RL policy much harder, and, at the same time, be able to evaluate its behaviour under minimal assumptions. Finally, this work is based on the argument that RL based techniques can be proved very valuable towards the developments of driving policies, even in mixed driving scenarios, and thus, it can be seen as a preliminary proof-of-concept.

4.2 Action representation

Seven available actions are defined; (i) change lane to the left, (ii) change lane to the right, (iii) accelerate with a constant acceleration of $1\,\mathrm{m/s^2}$ or $2\,\mathrm{m/s^2}$, (iv) decelerate with a constant deceleration of $-1\,\mathrm{m/s^2}$ or $-2\,\mathrm{m/s^2}$, and (v) move with the current speed at the current lane, as shown in Table 1. For the acceleration and deceleration actions, feasible acceleration and deceleration values are used to ensure that the autonomous vehicle will be able to implement them. Moreover, the autonomous vehicle is making decisions by selecting one action every 1 s, which implies that the first two actions are also feasible, i.e. a moving car is able to change lane in a time interval of 1 s.

Using the aforementioned action representation, each action can be seen as a goal or desire of the autonomous vehicle that should be achieved during 1 s. Practically, the first six actions represent goals that are associated with the avoidance of obstacles. The third to sixth actions represent also goals that are related to the fact that the autonomous vehicle should move forward with the desired speed. Finally, the seventh action implies that the vehicle is moving with the desired speed and there are no obstacles to avoid.

Note that the goal of this work is to develop a driving policy by approximating through RL the action-values $Q(s,a)$ for every possible $(s,a) \in \mathcal{S} \times \mathcal{A}$ pair. Therefore, adopting an action space with small cardinality can significantly simplify the problem leading to faster training. Moreover, the authors of [45] argue that low-level control tasks can be less effective and/or robust for high-level driving policies. For these reasons, an action space like the one presented above is used instead of lower-level commands such as longitudinal and lateral accelerations.

Finally, by using an action set of goals, the RL-based driving policy makes high-level decisions for leading the autonomous vehicle to the desired state. The implementation of these goals can efficiently take place by exploiting a separate non-learnable module, such as dynamic programming. This low-level module will produce state trajectories by translating each specific desire to lower-level commands, such as longitudinal and lateral accelerations. These state trajectories may then be used as a reference by the vehicle throttle and brake controllers, which are designed on the basis of vehicle dynamics, to produce the actual vehicle movement on the road. As mentioned in Section 2, the development of such a module is beyond the scope of this work, and, thus, it is assumed that is given.

4.3 Reward signal design

The reward signal is a measure of the quality of a selected action at a specific state, and is the only mean through which a policy can be evaluated. So, designing appropriate rewards signals is the most important tool for shaping the driving behaviour of an autonomous vehicle.

For driving scenarios, the autonomous vehicle should be able to avoid collisions, move with a specific desired speed, and avoid unnecessary lane changes and accelerations. Therefore, the reward signal should reflect all these objectives by employing one penalty function for collision avoidance, one that penalises deviations from the desired speed and two penalty functions for unnecessary lane changes and accelerations/decelerations.

The penalty function for collision avoidance should feature high values at the gross obstacle space, so that the autonomous vehicle is repulsed, and potentially unsafe decisions are suppressed; and low (or virtually vanishing) values outside that space. To this end, the exponential penalty function

$$f(\delta_i) = \begin{cases} e^{-(\delta_i - \delta_0)} & \text{if } l_e = l_i \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

is adopted. In (9), δ_i is the longitudinal distance between the automated vehicle and the ith obstacle (the ith vehicle in its surrounding environment), δ_0 stands for the minimum safe distance, and, l_e and l_i denote the lanes occupied by the autonomous vehicle and the ith obstacle, respectively. Note that this function is activated only when the automated vehicle and an obstacle are in the same lane. Finally, if the value of (9) becomes greater or equal to one, then the driving situation is considered very dangerous and it is treated as a collision.

The vehicle's mission is to advance with a longitudinal speed close to a desired one. Thus, the quadratic term

$$h(v) = (v - v_d)^2 \tag{10}$$

that penalises the deviation between the vehicle speed and its desired speed, is incorporated in the reward. In (10), the variable v

stands for the longitudinal speed of the autonomous vehicle, while the constant v_d represents its desired longitudinal speed.

Two terms are also introduced; one for penalising accelerations/decelerations, and one for penalising unnecessary lane changes. For penalising accelerations the term

$$a(v_t, v_{t-1}) = (v_t - v_{t-1})^2 \tag{11}$$

is used, while for penalising lane changes the term

$$g(l_t, l_{t-1}) = \mathbb{I}(l_t \neq l_{t-1}). \tag{12}$$

is used. Variables v_t and l_t correspond to the speed and lane of the autonomous vehicle at a time step t, while $\mathbb{I}(\cdot)$ is the indicator function.

The total reward at time step t is the negative weighted sum of the aforementioned penalty terms, i.e.

$$\begin{aligned} r_t = & -w_1 \sum_{i=1}^{O_t} f_t(\delta_i) - w_2 h_t(v_t) - w_3 \sum_{i=1}^{O_t} \mathbb{I}(f_t(\delta_i) \geq 1 \\ & - w_4 a(v_t, v_{t-1}) - w_5 g(l_t, l_{t-1}) \end{aligned} \tag{13}$$

In (13), the third term penalises collisions and variable O_t corresponds to the total number of obstacles that can be sensed by the autonomous vehicle at a time step t. The selection of weights defines the importance of each penalty function to the overall reward. In this work, the weights were set, using a trial and error procedure, as follows: $w_1 = 1$, $w_2 = 0.5$, $w_3 = 20$, $w_4 = 0.01$, $w_5 = 0.01$. The largest weighting factors are associated with the terms that penalise collisions and model obstacle avoidance, since the derived policy should generate collision-free trajectories. The weighting term associated with the desired speed of the vehicle defines how aggressive and/or how conservative will be the derived driving policy. Using a small value for this weight will result in a conservative policy that will advance the vehicle with very low speed or, even worse, keep the vehicle immobilised by setting its speed equal to zero. Finally, the values of the weighting factors associated with lane change accelerations/decelerations are small in order to enable the vehicle to make manoeuvres, such as overtaking other vehicles.

4.4 Neural network architecture

As mentioned before, the goal of this work is to develop a driving policy by approximating through RL the action-values $Q(s, a)$ for every possible $(s, a) \in \mathcal{S} \times \mathcal{A}$ pair. Towards this direction, a fully connected feed-forward neural network is utilised, due to its universal function approximation property [54].

Specifically, the action-values $Q(s, a)$ for each pair $(s, a) \in \mathcal{S} \times \mathcal{A}$ are approximated by using a neural network that maps a specific state $s \in \mathcal{S}$ to the action-values $Q(s, a_{s,i})$, where $\{a_{s,i}\}_i$ is a non-empty set that contains all actions that can be selected by the policy when the agent is at state s. In this work, the DDQN approach is followed, which utilises two identical neural networks with two hidden layers, consist of 256 and 128 neurons, respectively. The first neural network is responsible for setting the targets, while the second one is responsible for following them. The synchronisation between the two neural networks is realised every 1000 epochs. For more information regarding the DDQN model please refer to [48].

4.5 Training set generation and policy training

For generating (s_t, a_t, r_t, s_{t+1}) tuples that will be used for training the DDQN, two different microscopic traffic flow simulators are used. The first one is a custom made simulator that moves the manual driving vehicles with constant speed using the kinematics equations. The second simulator is the established SUMO [www.sumo.dlr.de/] microscopic traffic flow simulator. By exploiting traffic flow simulators driving scenarios can be simulated. For each one of the simulation steps during a simulated scenario, following the approach described in Section 4, one (s_t, a_t, r_t, s_{t+1}) tuple can be generated using information coming directly from the simulator.

After the collection of a set of (s_t, a_t, r_t, s_{t+1}) tuples, the training of the RL policy is starting following the procedures described in Section 3. It should be mentioned that during policy training (and testing) we implemented a rule-based action masking [45] for changing lanes. Our choice is justified by the fact that in some driving situations, undesirable lane changes can be straightforward identified, e.g. lane changes that result in immediate collisions. In such cases undesirable lane changes are filtered out instead of letting the agent learn to avoid those actions. The benefits of action masking are twofold. First, it restricts the action space, and, thus, it speeds up the learning process. Second, selection of inferior actions caused by the variance in observation will be avoided resulting in a policy that is less prone to false positives and easier to debug. Besides the aforementioned action masking, during training, no other safety mechanisms are applied to the behaviour of the autonomous vehicle. On the contrary, regarding manual driving cars, all safety mechanisms are enabled. Therefore, in case of a collision we are sure that the vehicle that caused the collision is the autonomous one.

These are the general rules applied during the driving scenarios generation (for training and testing the RL-based driving policy) using both of the aforementioned microscopic traffic-flow simulators. Depending on the specific characteristics of each experiment, extra rules may be applied. These are described in the corresponding subsections of Section 6.

4.6 Implementation details

For training the network, we set the discount factor $\gamma = 0.995$ [see relation (1)], we used the memory of 2000 samples capacity, a mini-batch of 64 samples and the ADAM optimiser with learning rate 0.003, $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The exploration factor ϵ_t at each step is annealed by

$$\epsilon_k = 0.01 + 0.99 e^{-\lambda k}, \tag{14}$$

where k stands for the index of the latest training step and λ was set equal to $7.5 \cdot 10^{-6}$. Finally, the training process started with $\epsilon_1 = 1.0$ and terminated when $\epsilon_k = 0.01$.

5 Safety rules

As mentioned before, no learning-based driving policy can guarantee a collision-free trajectory. There will always be corner cases (very rare events) that the learning algorithm will not encounter during its training phase. Therefore, it cannot be assured that the decisions corresponding to such event will be correct [for a formal proof of this result see [16] Lemma 2]. Moreover, a vehicle might be involved in an accident without being responsible for it. For these reasons the authors of [16] derive ad-hoc rules to guarantee responsibility-sensitive safety, i.e. to guarantee that an autonomous vehicle will never cause an accident, even if it will be involved in one.

The derivation of safety rules in this work is motivated by the responsibility-sensitive framework. There is, however, the main difference between the setting in [16] and our setting. The authors in [16] assume that the road is occupied only by autonomous vehicles whose behaviour can be programmed. In our case, there is no such assumption. On the contrary, mixed driving scenarios are considered, where the road is occupied both by autonomous and manual driving vehicles. This implies that the behaviour of manual driving vehicles cannot be affected either programmed. By restricting attention on vehicles that move on a highway the aforementioned assumption can be removed. This allows assuming that extreme events, such as vehicles that stop suddenly, will not occur.

Restricting attention on highways permits also the simplification of the responsibility-safety framework by considering two types of collisions. An autonomous vehicle can cause an accident, firstly, if it moves faster than its leader and

violates a minimum time gap, and, secondly, during lane changes. In the following we derive rules for avoiding these two types of collisions. Please note that the information that can be used to derive such rules is only the information available to the autonomous vehicle, i.e. the positions and the velocities of the vehicles surrounding it.

In order to avoid the first type of collisions, the minimum safety time gap ρ_s that must be maintained between the autonomous vehicles and its leader needs to be estimated. Obviously, the minimum safety time gap makes sense only when the autonomous vehicle is moving faster than its leader. Let us denote as $v_{e,t}$ and as $v_{l,t}$ the longitudinal speeds of the autonomous vehicle and its leader vehicle, respectively. Also, let us denote as $d_{\max}$ the maximum feasible deceleration of the autonomous vehicle. In order to avoid the first type of collisions after a time interval ρ, the following inequality should hold:

$$v_{l,t}\rho - v_{e,t}\rho + \frac{1}{2}d_{\max}\rho^2 > 0. \tag{15}$$

Solving for ρ the minimum safety time gap ρ_s can be obtained by

$$\rho_s = \inf\left\{\rho : \rho > \frac{2(v_{e,t} - v_{l,t})}{d_{\max}}\right\}. \tag{16}$$

Based on relation (16), the autonomous vehicle before performing an action, different than lane change actions, evaluates the minimum safety gap with respect to its leader. If the minimum time gap is violated, the autonomous vehicles decelerate with $d_{\max}$ until its speed becomes equal to the speed of its leader. Otherwise, it performs the RL selected action.

Regarding the second type of collisions that can be caused by lane changes, two different cases should be considered. The autonomous vehicle should avoid collisions with its leader vehicle and with its follower vehicle in the newly selected lane. In the first case, it estimates the minimum safety time gap ρ_s with respect to its leader in the newly selected lane. If the minimum time gap is not violated the RL lane change action is performed. Otherwise, the autonomous vehicle selects the last action of the action set $\mathscr{A}$ [see Section 4.2], that is, to retain current lane and move with current speed, and checks for the first type of collisions. In order to avoid the collisions with its follower vehicle in the newly selected lane, the autonomous vehicle is not permitted to change lane if the follower vehicle moves faster. In this case again, the autonomous vehicle selects the last action of the action set $\mathscr{A}$, and checks for the first type of collisions. The rule for avoiding collisions between the autonomous vehicle and its followers is very conservative. However, since the RL-based driving policy cannot affect the behaviour of the follower, and at the same time has no access to its maximum feasible deceleration (in order to relax this rule by estimating a safety time gap), such a rule is the only way to guarantee no collisions of the second type.

Although the derived safety rules lead to more conservative driving policy, as it can be seen in the experimental validation of the proposed approach, they permit the autonomous vehicle to advance with its desired speed and at the same time avoid collisions.

Table 2 Driving behaviour evaluation of the RL and DP driving policies, in terms of the total number of collisions and lane changes for 100 scenarios and percentage of time that the vehicle moves with its desired speed

	Collisions	Lane changes	Desired speed, %
1 veh./8 s			
DP policy	0	84	85
RL policy	0	81	73
1 veh./4 s			
DP policy	0	127	83
RL policy	0	115	64
1 veh./2 s			
DP policy	0	120	87
RL policy	0	108	62
1 veh./1 s			
DP policy	0	70	72
RL policy	2	62	56

6 Experiments

In this work, three different sets of experiments were conducted. In the first set of experiments, a simplified microscopic traffic flow simulator is utilised in order to compare the behaviour of the RL-based driving policy against an optimal policy derived via dynamic programming. In the second set of experiments, the established microscopic traffic simulator SUMO is used. Three different types of experiments are conducted. First, the behaviour of the autonomous vehicle is evaluated when it is controlled by the derived RL-based policy and when it is controlled by SUMO. Second, the robustness of the derived policy with respect to measurement errors is evaluated. Finally, in the third set of experiments, the effect of vehicles that move following the RL-based policy on traffic flow is investigated. In the following the details of the experimental setup and the obtained results are presented.

6.1 RL-based driving policy and dynamic programming

Dynamic programming techniques can produce optimal policies assuming that no disturbances occur in the system. Due to this fact, for this set of experiments, a simplified custom made microscopic traffic simulator was developed and utilised. This simulator moves the manual driving vehicles with constant longitudinal velocity using the kinematics equations. Moreover, manual driving vehicles are not allowed to change lanes. Despite its simplifying setting, this set of experiments allows the comparison of the RL driving policy against an optimal policy derived via dynamic programming. At this point it should be mentioned that for this set of experiments the *ad-hoc* safety rules derived in Section 5 are disabled in order to gain insights regarding the safety aspects of the RL-based driving policy.

For training the DDQN, driving scenarios of 60 s length were generated. In these scenarios, one vehicle enters the road every 2 s, while the tenth vehicle that enters the road is the autonomous one. All vehicles enter the road at a random lane, and their initial longitudinal velocity is randomly selected from a uniform distribution ranging from 12 to 17 m/s. Finally, the desired speed of the autonomous vehicle is set equal to 21 m/s.

The RL driving policy is compared against an optimal policy derived via dynamic programming under four different road density values. For each one of the different densities, 100 scenarios of 60 s length were simulated. In these scenarios, the simulator moves the manual driving vehicles, while the autonomous vehicle moves by following the RL policy and by solving a dynamic programming problem with 60 s horizon (which utilises the same objective functions and actions as the RL algorithm). Finally, statistics regarding the number of collisions and lane changes, and the percentage of time that the autonomous vehicle moves with its desired speed for both the RL and dynamic programming policies are extracted. At this point, it has to be mentioned that dynamic programming is not able to produce the solution in real time, and it is just used for benchmarking and comparison purposes. On the contrary the RL policy, at a given state can select an action very fast since this selection corresponds to one evaluation of the neural network function at the corresponding state.

Table 2 summarises the results of this comparison. The four different densities are determined by the rate at which the vehicles enter the road, i.e. 1 vehicle enters the road every 8, 4, 2, and 1 s. The RL policy is able to generate collision-free trajectories when the density is less than or equal to the density used to train the network. For larger densities, however, the RL policy produced 2 collisions every 100 scenarios. In terms of efficiency, the optimal

dynamic programming policy is able to perform more lane changes and advance the vehicle faster.

6.2 RL-based driving policy and SUMO policy

In this set of experiments, the behaviour of the autonomous vehicle when it follows the RL policy and when it is controlled by SUMO is evaluated. The training of the RL policy took place using scenarios generated by the SUMO simulator. During the generation of scenarios, all SUMO safety mechanisms are enabled for the manual driving vehicles and disabled for the autonomous vehicle. Furthermore, the manual driving cars are not permitted to implement cooperative and strategic lane changes. Such a configuration for the lane changing behaviour, impels the autonomous vehicle to implement manoeuvres in order to achieve its objectives. Moreover, in order to simulate realistic scenarios, two different types of manual driving vehicles are used; vehicles that want to advance faster than the autonomous vehicle and vehicles that want to advance slower. Finally, the density was equal to 600 veh./lane/h. For the evaluation of the trained RL policy, different driving scenarios, described in the following subsections, were simulated.

6.2.1 Evaluation of the derived RL driving policy and safety rules: In this set of experiments, different driving scenarios were simulated; (i) 100 driving scenarios during which the autonomous vehicle follows the RL driving policy without the *ad-hoc* safety rules derived in Section 5, (ii) 100 driving scenarios during which the autonomous vehicle follows the RL driving policy with the ad-hoc safety rules, (iii) 100 driving scenarios during which the default configuration of SUMO was used to move forward the autonomous vehicle (cooperative and strategic lane changes are enabled for the autonomous vehicle), and (iv) 100 scenarios during which the behaviour of the autonomous vehicle is the same as the manual driving vehicles, i.e. it does not perform strategic and cooperative lane changes. The duration of all simulated scenarios was 60 s. The aforementioned scenarios' generation framework was applied to two different driving conditions. In the first one, the desired speed for the slow manual driving vehicles was set to 18 m/s, while in the second one to 16 m/s. For both driving conditions, the desired speed for the fast manual driving vehicles was set to 25 m/s. Furthermore, in order to investigate how the presence of uncertainties affects the behaviour of the autonomous vehicle, simulated scenarios where drivers' imperfection was introduced by appropriately setting the σ parameter in SUMO ($0 \leq \sigma \leq 1$ with $\sigma = 0$ to imply a perfect driver) were also used. Finally, the behaviour of the autonomous vehicles was evaluated in terms of (i) collision rate, and (ii) average speed per scenario.

Table 3 Driving behaviour evaluation. *SUMO default* corresponds to the default SUMO configuration, while *SUMO manual* to the case where the behaviour of the autonomous vehicle is the same as the manual driving vehicles

	Collisions, %	Avg. speed
Desired speed for slow vehicles 18 m/s		
RL policy with rules ($\sigma = 0.0$)	0	20.62
RL policy w/o rules ($\sigma = 0.0$)	2	20.71
SUMO default ($\sigma = 0.0$)	0	20.22
SUMO manual ($\sigma = 0.0$)	0	19.48
RL policy with rules ($\sigma = 0.5$)	0	20.08
RL policy w/o rules ($\sigma = 0.5$)	3	20.09
SUMO default ($\sigma = 0.5$)	0	19.57
SUMO manual ($\sigma = 0.5$)	0	19.05
Desired speed for slow vehicles 16 m/s		
RL policy with rules ($\sigma = 0.0$)	0	19.87
RL policy w/o rules ($\sigma = 0.0$)	2	20.04
SUMO default ($\sigma = 0.0$)	0	18.41
SUMO manual ($\sigma = 0.0$)	0	17.47
RL policy with rules ($\sigma = 0.5$)	0	19.81
RL policy w/o rules ($\sigma = 0.5$)	4	19.87
SUMO default ($\sigma = 0.5$)	0	17.67
SUMO manual ($\sigma = 0.5$)	0	17.26

Table 4 Driving behaviour evaluation with *ad-hoc* safety rules when different magnitudes of measurements errors are introduced

	Collisions, %	Avg speed
5% noise		
RL policy with rules ($\sigma = 0.0$)	0	19.88
RL policy with rules ($\sigma = 0.5$)	0	19.84
10% noise		
RL policy with rules ($\sigma = 0.0$)	0	19.65
RL policy with rules ($\sigma = 0.5$)	0	19.59

Table 3 summarises the results of this comparison when the *ad-hoc* safety rules are disabled. In this way, the safety levels of the RL-based driving policy can be experimentally quantified. In Table 3, *SUMO default* corresponds to the default SUMO configuration for moving forward the autonomous vehicle, while *SUMO manual* to the case where the behaviour of the autonomous vehicle is the same as the manual driving vehicles. Irrespective of whether a perfect ($\sigma = 0$) or an imperfect ($\sigma = 0.5$) driver is considered for the manual driving vehicles, the RL policy is able to move forward the autonomous vehicle faster than the SUMO simulator, especially when slow vehicles are much slower than the autonomous one. However, it results in a collision rate of 2–4%, which is its main drawback. No guarantees for collision-free trajectory is the price paid for deriving a learning-based approach capable of generalising to unknown driving situations and inferring driving actions with minimal computational cost.

However, when the *ad-hoc* safety rules are enabled, the derived RL driving policy achieves to provide collision-free trajectories. The average speed of the autonomous vehicle slightly decreases after the application of *ad-hoc* rules, but again the derived policy advances the autonomous vehicle faster than the SUMO policies. Specifically, when the speed of the slow vehicles is 18 m/s the RL-based policy with the *ad-hoc* safety rules advances the autonomous vehicle 2 and 2.6% faster than the SUMO default policy for $\sigma = 0.0$ and $\sigma = 0.5$, respectively. For the case where the speed of the slow vehicles is 16 m/s, the improvement, in terms of speed, of the RL-based policy over the SUMO default policy is more significant. In particular, the RL-based policy advances the autonomous vehicle 8 and 12% faster than the SUMO default policy for $\sigma = 0.0$ and $\sigma = 0.5$, respectively.

The aforementioned results suggest that the RL-based driving policy is not significantly more efficient than the SUMO default policy when the average speed of the manual driving vehicles is close to the desired speed of the autonomous vehicle. However, when the deviation between the desired speed of the autonomous vehicle and the average speed of the manual driving vehicles increases, the RL-based driving policy is able to advance the autonomous vehicle much faster.

6.2.2 Evaluation of the derived RL driving policy under measurement errors: In this set of experiments, the robustness of the RL-based driving policy, with the application of *ad-hoc* safety rules, is evaluated with respect to measurement errors regarding the position of the manual driving vehicles. At each time step, measurement errors proportional to the distance between the autonomous vehicle and the manual driving vehicles are introduced. Two different error magnitudes were used; ± 5 and $\pm 10\%$. The RL policy was evaluated in terms of collisions and average speed in 100 driving scenarios of 60 s length for each error magnitude. In these scenarios, the desired speed of the slow vehicles is 16 m/s. Finally, for these experiments, perfect and imperfect drivers were also considered.

The results of this evaluation are presented in Table 4. Despite the introduction of noise, the RL-based driving policy is able to produce collision-free trajectories, and at the same time, retain a high speed for the autonomous vehicle. In particular, the

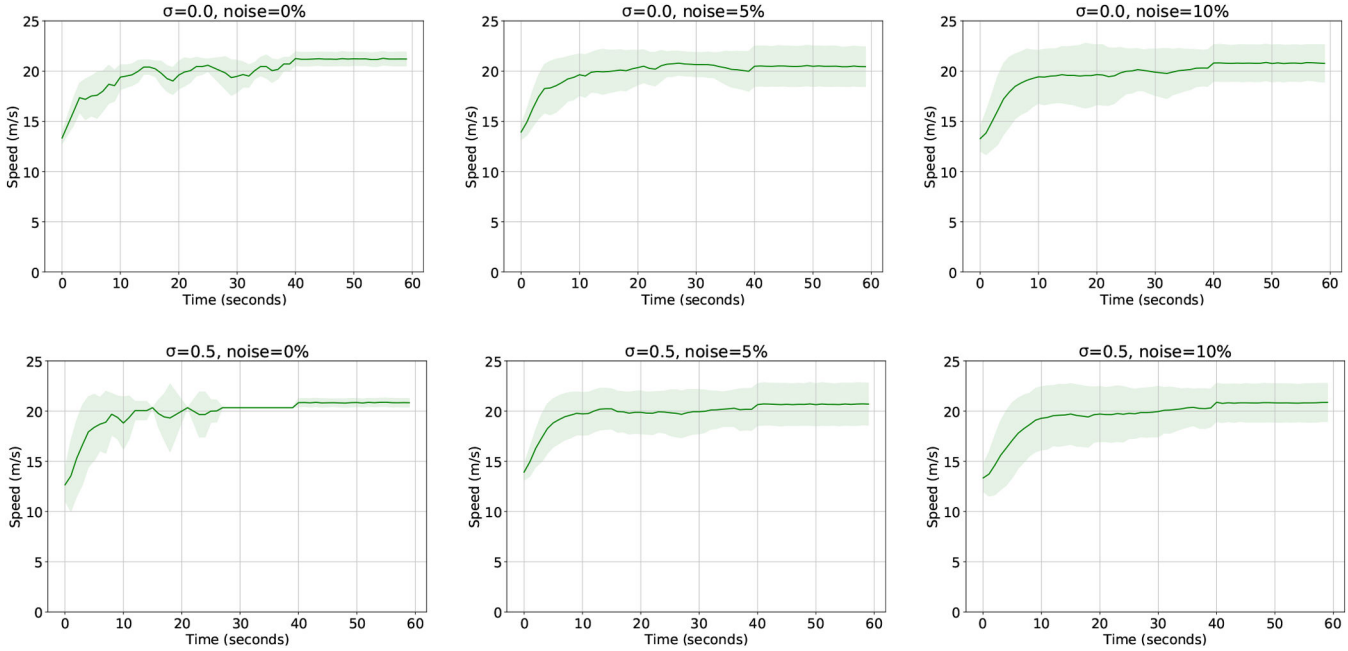

Fig. 2 *Speed trajectories for different measurement errors and driver imperfection. The solid green line represents the mean speed of the vehicle overall 100 scenarios, while the shaded area represents 1 standard deviation of the speeds below and above their mean value*

Table 5 Vehicle types present on the road

	Percentage, %	Maximum speed, m/s
slow passenger vehicles	40	16
fast passenger vehicles	40	25
trucks	5	14
buses	5	16
motorcycles	10	21

Table 6 Driving behaviour evaluation with ad-hoc safety rules when different types of vehicles occupy the road

	Collisions, %	Avg. speed
RL policy with rules ($\sigma = 0.0$)	0	19.74
RL policy with rules ($\sigma = 0.5$)	0	19.67

introduction of a 5% noise does not seem to affect the average speed of the vehicle. By increasing the noise to 10% the average speed of the vehicle slightly decreases compared to the case with noiseless measurements. Fig. 2 presents the speed trajectories of the autonomous vehicle when different drivers' imperfection and different magnitude of noises are introduced. The solid green line represents the mean speed of the vehicle overall 100 scenarios, while the shaded area represents 1 standard deviation of the speeds below and above their mean value. Irrespective of the introduced uncertainties, during the first steps of the simulation the autonomous vehicle increases its speed to reach a speed close to its desired one, and then, it retains this speed. Moreover, by increasing the noise, the deviation of the speeds over the 100 scenarios increases. This, however, is rational behaviour, since increasing the uncertainty, in terms of noisy measurements, will increase the variance during the decision making process.

6.2.3 Evaluation of the derived RL driving policy with unknown vehicle types: In this set of experiments, the robustness of the derived RL-based driving policy is evaluated when the road is occupied by types of vehicles that were not present during the training phase. The RL-based driving policy was trained using driving scenarios where the road was occupied by *passenger* manually driving vehicles that were moving faster and slower than the autonomous vehicle. In this set of scenarios, the road is occupied by the previously mentioned passenger vehicles, but also by truck, buses, and motorcycles. The percentage of these types of vehicles, as well as their desired speed, is presented in Table 5. Under this experimental setting, the robustness of the derived driving policy can be evaluated when vehicles of different sizes and different desired speeds occupy the road. Towards this direction 100 driving scenarios considering perfect drivers and 100 scenarios considering drivers' imperfections were simulated. All driving scenarios were 60 s long. Finally, the RL-based driving policy was evaluated in terms of collisions and average speed with which the autonomous vehicle moves forward.

Table 6 presents the RL driving policy evaluation results for the aforementioned set of experiments. By comparing these results with the results in Table 3, it can be seen that the average speed of the autonomous vehicle is slightly decreased by 0.13 and 0.14 m/s, for $\sigma = 0.0$ and $\sigma = 0.5$, respectively, when types of vehicles not seen during the training phase are present in the road. This decrease is mainly due to the randomness during driving scenarios generation and not due to the presence of trucks, buses and motorcycles on the road. More importantly, the RL driving policy is able to produce collision-free trajectories despite the fact that the road is occupied by types of vehicles not seen during the training phase. This is justified by two facts. First, the proposed state representation utilises encodes about the position and the velocity of manual driving vehicles present on the road. This kind of information can be obtained and encoded for any vehicle irrespective of its type. Second, the development and application of the proposed safety rules compensate for the presence of manually driving vehicles of different sizes. It should be mentioned, however, that more realistic and accurate state representations (see Section 4.1) can also be utilised to explicitly encode vehicles size information in state representation.

Table 7 Driving behaviour evaluation with ad-hoc safety rules rainy weather conditions

	Collisions, %	Avg. speed
RL policy with rules ($\sigma = 0.5$)	0	19.48

Table 8 Effect of autonomous vehicles on the overall traffic flow

	Avg. speed, m/s	Improvement over 0%, %
autonomous vehicles 0%	15.32	0.0
autonomous vehicles 5%	15.41	0.6
autonomous vehicles 10%	16.11	5.1
autonomous vehicles 20%	15.91	1.3

6.2.4 Evaluation of the derived RL driving policy under rainy weather conditions: In this set of experiments, the RL driving policy is evaluated under rainy weather driving conditions. Rainy weather shifts the fundamental diagram to the left, which implies, on the one hand, that the vehicles move slower, and on the other, that their maximum acceleration/deceleration becomes lower. In order to simulate rainy weather driving scenarios, the desired speed of all vehicles, except the autonomous one, was decreased by 10%, and their maximum acceleration/deceleration by 30%. Moreover, drivers' imperfections were also included by setting $\sigma = 0.5$. Finally, different types of vehicles, i.e. slow and fast passenger vehicles, trucks, buses, and motorcycles, were also present on the road. 100 driving scenarios of 60 s long were simulated, and the derived driving policy was evaluated in terms of number of collisions and average speed with which the autonomous vehicle moves forward. Table 7 presents the result of this evaluation.

The RL-based driving policy is able to produce collision-free trajectories and, at the same time, move forward the autonomous vehicle with speed larger than 19 m/s. The longitudinal velocity of the autonomous vehicle is slightly lower (0.19 m/s) than the previous experiments. This, however, is mainly caused by the decrease in overall traffic flow due to weather conditions.

6.3 Effect of the RL-based driving policy on traffic flow

In this set of experiments, preliminary results on the effect of autonomous vehicles on the overall traffic flow are presented. Four different experiments are conducted by varying the percentage of autonomous vehicles that occupy the road. In the first experiment, all vehicles are manual driving, i.e. the percentage of autonomous vehicles is zero. For the rest three experiments, percentages of 5, 10, and 15%, respectively, are used. For each experiment, 100 scenarios of 120 s length were simulated, and for each scenario the average speed of all vehicles on the road is computed, which is an indicator of the flow; the higher the average speed, the higher is the flow. For all experiments and all scenarios, the desired speed of manual driving vehicles is 16 m/s, and the option for cooperative and strategic manoeuvres is disabled, while the desired speed for all the autonomous vehicles is 21 m/s. In this way, the behaviour of manual driving vehicles can be seen as a moving bottleneck.

The results of these experiments are presented in Table 8. For each one of the experiments, the average speed is reported. As a baseline the case where the percentage of autonomous vehicles is zero is considered, and the relative improvement of the rest of cases (5, 10, and 15% autonomous vehicles) against this one is also reported. The average speed for the baseline is 15.32 m/s, a little bit lower than the desired speed of the manual driving vehicles. This happens because the manual driving vehicles should satisfy the safety constraints imposed by SUMO. When the percentage of autonomous vehicles increases to 5% the average speed of the vehicles is 15.41 m/s resulting in a very small improvement of 0.6% over the baseline. In this case, the autonomous vehicles move faster than the manual driving ones. Their percentage, however, is very small, and, thus, they only slightly improve the average speed compared to the baseline. Increasing more the percentage of autonomous vehicles to 10% results in an average speed of 16.11 m/s and 5.1% improvement compared to the baseline. Increasing, however, more the percentage of autonomous vehicles to 20% results in an improvement of 1.3% over the baseline, which is smaller than the improvement of the previous case. This mainly happens due to the selfish behaviour of autonomous vehicles.

Autonomous vehicles want to move faster than manual driving cars, and in order to achieve that they have to perform manoeuvres. Keeping the density of autonomous vehicles low permits the performance of manoeuvres, and thus, the faster advancement of the autonomous vehicles. Increasing, however, the density of autonomous vehicle above a threshold, makes the performance of manoeuvres more difficult, since each one autonomous vehicle, present in a limited space, wants to perform its own manoeuvres in a selfish way. This results in competitive behaviours among autonomous vehicles, which has a negative effect on the overall traffic flow.

These preliminary results show that selfish and competitive behaviours deteriorate the overall traffic flow. Deriving an RL-based driving policy trained on scenarios where the manually driven vehicles occupy a selfish behaviour will not improve the overall traffic flow. Due to limited space and a large number of manoeuvres performed by the manually driven vehicles the RL training algorithm will result in a very conservative policy. In our view, the only way to improve the overall traffic flow, under mixed driving scenarios, is to derive cooperative driving policies for clusters of autonomous vehicles, in order to achieve not vehicle-centric, but overall traffic flow goals. This could be done by introducing appropriate penalty terms regarding the overall traffic flow, such as minimum travel time or average traffic flow, in the reward function. Deriving, however, cooperative RL-based driving policies for clusters of autonomous vehicles are outside the scope of this work.

7 Discussion

The simulation results presented in Section 6 indicate that the derived RL-based driving policy is more efficient for moving forward the autonomous vehicle, than the car following model used by SUMO simulator. At the same time, the derived policy can produce collision-free trajectories, and it seems to be robust under measurement errors, different types of vehicles and weather conditions. Although the current work makes the first steps towards the exploitation of deep RL techniques for autonomous vehicles' path planning, the proposed methodology is not yet ready for real-world adoption. More complicated scenarios should be generated and utilised during the training and testing phases, such as scenarios where the autonomous vehicle is approaching a crash site ahead, heavy traffics, highway merging, emergency lane switching, and night driving.

Being able to identify the limitations of the current work motivates our ongoing and future work, which comprises (i) training and testing an RL-based driving policy under more complicated and realistic scenarios, (ii) derive more accurate state representations by exploiting vehicle-to-vehicle and vehicle-to-infrastructure communication technologies, and (iii) move from a selfish driving policy to the derivation of a cooperative driving policy in order to achieve not vehicle-centric, but overall traffic flow goals.

8 Conclusions

In this work, the problem of path planning for an autonomous vehicle that moves on a freeway is considered. For addressing this problem, RL techniques are employed to derive a driving policy. The driving policy is implemented using a DDQN. Two different simulators to train and validate the derived driving policy are used; a custom made microscopic traffic flow simulator and the established SUMO microscopic traffic flow simulator.

The custom made microscopic traffic flow simulator is utilised for comparing the RL-based driving policy against an optimal policy derived via dynamic programming. The results of this comparison indicated that, although dynamic programming can advance the autonomous vehicle faster than the RL-based driving policy, it cannot produce the trajectory in real time. Moreover,

dynamic programming requires a priori and exact knowledge of the system dynamics in a disturbance-free environment to produce an optimal solution. Due to these facts, an RL-based driving policy that incorporates the ad-hoc safety rules [see Section 5] can be proved a valuable approach for emerging driving behaviours with very low-computational cost, minimal or no assumptions about the environment, and the capability to generalise to driving situations that are not known a priori.

The SUMO simulator is utilised in order to train and validate the RL-based driving policy under customary and realistic traffic scenarios. Since no learning-based approach can guarantee collision-free trajectories, *ad-hoc* safety rules are derived motivated by the responsibility-safety framework presented in [16]. The derived RL-based driving policy is compared against SUMO policies with and without the introduction of uncertainties. The results of this comparison indicated that the autonomous vehicle following the RL-based policy is able to achieve higher scores.

Finally, preliminary results regarding the effect of selfish autonomous vehicles behaviour on the overall traffic flow are presented. These results suggest that, although an individual autonomous vehicle that follows a selfish policy can achieve its goals, when multiple autonomous vehicles follow a selfish policy, their impact on the overall traffic flow is negative. Selfish policies lead to competitive behaviours that deteriorate the overall traffic flow. This effect is known as the *user optimum* versus *system optimum* trade-off.

9 Acknowledgment

This research was implemented through and has been financed by the Operational Program 'Human Resources Development, Education and Lifelong Learning' and is co-financed by the European Union (European Social Fund) and Greek national funds.

10 References

[1] Ziegler, J., Bender, P., Dang, T., *et al.*: 'Trajectory planning for bertha – a local, continuous method'. Intelligent Vehicles Symp, Dearborn, MI, USA, June 2014, pp. 450–457

[2] Cosgun, A., Ma, L., Chiu, J., *et al.*: 'Towards full automated drive in urban environments: a demonstration in Gomentum station, California'. Intelligent Vehicles Symp. (IV), Los Angeles, CA, USA, June 2017, pp. 1811–1818

[3] Reimer, B., Mehler, B., Coughlin, J.F.: 'An evaluation of driver reactions to new vehicle parking assist technologies developed to reduce driver stress' (Massachusetts Institute of Technology, Cambridge, MA, USA, 2010), pp. 1–26

[4] Donges, E.: 'A conceptual framework for active safety in road traffic', *Veh. Syst. Dyn.*, 1999, **32**, (2–3), pp. 113–128

[5] Menelaou, C., Timotheou, S., Kolios, P., *et al.*: 'Improved road usage through congestion-free route reservations', *Transp. Res. Rec., J. Transp. Res. Board*, 2017, **2621**, pp. 71–80

[6] Menelaou, C., Kolios, P., Timotheou, S., *et al.*: 'Controlling road congestion via a low-complexity route reservation approach', *Transp. Res. C, Emerg. Technol.*, 2017, **81**, pp. 118–136

[7] Bast, H., Carlsson, E., Eigenwillig, A., *et al.*: 'Fast routing in very large public transportation networks using transfer patterns'. European Symp. on Algorithms, Liverpool, UK, September 2010, pp. 290–301

[8] Figliozzi, M.: 'Vehicle routing problem for emissions minimization', *Transp. Res. Rec., J. Transp. Res. Board*, 2010, **2197**, pp. 1–7

[9] Baldacci, R., Mingozzi, A., Roberti, R.: 'Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints', *Eur. J. Oper. Res.*, 2012, **218**, (1), pp. 1–6

[10] Bast, H., Delling, D., Goldberg, A., *et al.*: 'Route planning in transportation networks', in: '*Algorithm engineering*' (Springer, Cham, Switzerland, 2016), pp. 19–80

[11] Gillespie, T.D.: '*Vehicle dynamics*' (SAE International, Warrendale, PA, USA, 1997)

[12] Rajamani, R.: '*Vehicle dynamics and control*' (Springer Science & Business Media, New York, NY, USA, 2011)

[13] Zhang, S., Deng, W., Zhao, Q., *et al.*: 'Dynamic trajectory planning for vehicle autonomous driving'. 2013 IEEE Int. Conf. on Systems, Man, and Cybernetics (SMC), Manchester, UK, October 2013, pp. 4161–4166

[14] Brilon, W., Geistefeldt, J., Regler, M.: 'Reliability of freeway traffic flow: a stochastic concept of capacity'. Proc. of the 16th Int. Symp. on Transportation and Traffic Theory, College Park, MA, USA, July 2005, pp. 125–144

[15] Yazici, M., Kamga, C., Ozbay, K.: 'Highway versus urban roads: analysis of travel time and variability patterns based on facility type', *Transp. Res. Rec., J. Transp. Res. Board*, 2014, **2442**, pp. 53–61

[16] Shalev-Shwartz, S., Shammah, S., Shashua, A.: 'On a formal model of safe and scalable self-driving cars', arXiv preprint arXiv:170806374, 2017, pp. 1–37

[17] Ntousakis, I.A., Nikolos, I.K., Papageorgiou, M.: 'Optimal vehicle trajectory planning in the context of cooperative merging on highways', *Transp. Res. C, Emerg. Technol.*, 2016, **71**, pp. 464–488

[18] Goerzen, C., Kong, Z., Mettler, B.: 'A survey of motion planning algorithms from the perspective of autonomous UAV guidance', *J. Intell. Robot. Syst.*, 2010, **57**, (1–4), pp. 65–100

[19] Werling, M., Gindele, T., Jagszent, D., *et al.*: 'A robust algorithm for handling moving traffic in urban scenarios'. Intelligent Vehicles Symp. (IV), Eindhoven, Netherlands, June 2008, pp. 1108–1112

[20] Fletcher, L., Teller, S., Olson, E., *et al.*: 'The MIT–Cornell collision and why it happened', *J. Field Robot.*, 2008, **25**, (10), pp. 775–807

[21] Wolf, M.T., Burdick, J.W.: 'Artificial potential functions for highway driving with collision avoidance'. Int. Conf. on Robotics and Automation (ICRA), Pasadena, CA, USA, May 2008, pp. 3731–3736

[22] Wang, J., Wu, J., Li, Y.: 'The driving safety field based on driver–vehicle–road interactions', *IEEE Trans. Intell. Transp. Syst.*, 2015, **16**, (4), pp. 2203–2214

[23] Schildbach, G., Borrelli, F.: 'Scenario model predictive control for lane change assistance on highways'. Intelligent Vehicles Symp. (IV), Seoul, South Korea, 28 June – 1 July 2015, pp. 611–616

[24] Erlien, S.M.: 'Shared vehicle control using safe driving envelopes for obstacle avoidance and stability', Stanford University, PhD Thesis, 2015

[25] Zhang, Y.J., Malikopoulos, A.A., Cassandras, C.G.: 'Optimal control and coordination of connected and automated vehicles at urban traffic intersections'. 2016 American Control Conf. (ACC), Boston, MA, USA, July 2016, pp. 6227–6232

[26] Carvalho, A., Gao, Y., Lefevre, S., *et al.*: 'Stochastic predictive control of autonomous vehicles in uncertain environments'. 12th Int. Symp. on Advanced Vehicle Control (AVEC), Tokyo, Japan, September 2014, pp. 1–8

[27] Gao, Y., Gray, A., Tseng, H.E., *et al.*: 'A tube-based robust nonlinear predictive control approach to semiautonomous ground vehicles', *Veh. Syst. Dyn.*, 2014, **52**, (6), pp. 802–823

[28] Makantasis, K., Papageorgiou, M.: 'Motorway path planning for automated road vehicles based on optimal control methods'. *Transp. Res. Rec. J. Transp. Res. Board*, 2018, **2672**, pp. 112–123

[29] Gao, Y., Lin, T., Borrelli, F., *et al.*: 'Predictive control of autonomous ground vehicles with obstacle avoidance on slippery roads'. ASME 2010 Dynamic Systems and Control Conf., Cambridge, MA, USA, September 2010, pp. 265–272

[30] Gray, A., Ali, M., Gao, Y., *et al.*: 'Semi-autonomous vehicle control for road departure and obstacle avoidance'. IFAC Control of Transportation Systems, Sofia, Bulgaria, September 2012, pp. 1–6

[31] Werling, M., Liccardo, D.: 'Automatic collision avoidance using model-predictive online optimization'. 51st Annual Conf. on Decision and Control (CDC), Maui, HI, USA, December 2012, pp. 6309–6314

[32] Rasekhipour, Y., Khajepour, A., Chen, S.K., *et al.*: 'A potential field-based model predictive path-planning controller for autonomous road vehicles', *IEEE Trans. Intell. Transp. Syst.*, 2017, **18**, (5), pp. 1255–1267

[33] Papageorgiou, M., Marinaki, M., Makantasis, K., *et al.*: 'A feasible direction algorithm for the numerical solution of optimal control problems', Dynamic Systems and Simulation Laboratory, Technical University of Crete, Chania, Greece, 2016

[34] Bellman, R.: 'On the theory of dynamic programming', *Proc. Natl. Acad. Sci.*, 1952, **38**, (8), pp. 716–719

[35] Bojarski, M., Yeres, P., Choromanska, A., *et al.*: 'Explaining how a deep neural network trained with end-to-end learning steers a car', arXiv preprint arXiv:170407911, 2017, pp. 1–8

[36] Chen, S., Zhang, S., Shang, J., *et al.*: 'Brain-inspired cognitive model with attention for self-driving cars', *IEEE Trans. Cogn. Dev. Syst.*, 2019, **11**, (1), pp. 13–25

[37] Chen, C., Seff, A., Kornhauser, A., *et al.*: 'Deepdriving: learning affordance for direct perception in autonomous driving'. 2015 IEEE Int. Conf. on Computer Vision (ICCV), Santiago, Chile, December 2015, pp. 2722–2730

[38] Xu, H., Gao, Y., Yu, F., *et al.*: 'End-to-end learning of driving models from large-scale video datasets'. Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, Honolulu, HI, USA, July 2017, pp. 2174–2182

[39] Glasmachers, T.: 'Limits of end-to-end learning', arXiv preprint arXiv:170408305, 2017, pp. 1–15

[40] Mnih, V., Kavukcuoglu, K., Silver, D., *et al.*: 'Human-level control through deep reinforcement learning', *Nature*, 2015, **518**, (7540), pp. 529–533

[41] Isele, D., Cosgun, A., Subramanian, K., *et al.*: 'Navigating intersections with autonomous vehicles using deep reinforcement learning', arXiv preprint arXiv:170501196, 2017, pp. 1–6

[42] Mukadam, M., Cosgun, A., Nakhaei, A., *et al.*: 'Tactical decision making for lane changing with deep reinforcement learning'. submitted to Int. Conf. on Learning Representations (ICLR), Long Beach, CA, USA, December 2017

[43] Paxton, C., Raman, V., Hager, G.D., *et al.*: 'Combining neural networks and tree search for task and motion planning in challenging environments', arXiv preprint arXiv:170307887, 2017, pp. 1–8

[44] Shalev-Shwartz, S., Shammah, S., Shashua, A.: 'Safe, multi-agent, reinforcement learning for autonomous driving', arXiv preprint arXiv:161003295, 2016, pp. 1–13

[45] Liu, J., Hou, P., Mu, L., *et al.*: 'Elements of effective deep reinforcement learning towards tactical driving decision making', arXiv preprint arXiv:180200332, 2018, pp. 1–7

[46] Bellman, R.: 'The theory of dynamic programming', *Bull. Am. Math. Soc.*, 1954, **60**, (6), pp. 503–515

[47] Kanagaraj, V., Asaithambi, G., Kumar, C.N., *et al.*: 'Evaluation of different vehicle following models under mixed traffic conditions', *Procedia-Soc. Behav. Sci.*, 2013, **104**, pp. 390–401

[48] Van Hasselt, H., Guez, A., Silver, D.: 'Deep reinforcement learning with double Q-learning'. Proc. of the Thirtieth AAAI Conf. on Artificial Intelligence (AAAI), Phoenix, AZ, USA, February 2016, pp. 2094–2100
[49] Schaul, T., Quan, J., Antonoglou, I., *et al.*: 'Prioritized experience replay', arXiv preprint arXiv:151105952, 2015, pp. 1–21
[50] Watkins, C.J., Dayan, P.: 'Q-learning', *Mach. Learn.*, 1992, **8**, (3-4), pp. 279–292
[51] Akai, N., Morales, L.Y., Yamaguchi, T., *et al.*: 'Autonomous driving based on accurate localization using multilayer LiDAR and dead reckoning'. 2017 IEEE 20th Int. Conf. on Intelligent Transportation Systems (ITSC), Yokohama, Japan, October 2017, pp. 1–6
[52] Wolcott, R.W., Eustice, R.M.: 'Robust LiDAR localization using multiresolution Gaussian mixture maps for autonomous driving', *Int. J. Rob. Res.*, 2017, **36**, (3), pp. 292–319
[53] De Silva, V., Roche, J., Kondoz, A.: 'Fusion of LiDAR and camera sensor data for environment sensing in driverless vehicles', 2018, https://hdl.handle.net/2134/33170
[54] Csáji, B.C.: 'Approximation with artificial neural networks', MSc Thesis, Eotvos Loránd University, Hungary, 2001