MULTI-AGENT REINFORCEMENT LEARNING FOR SAFE LANE CHANGES BY CONNECTED AND AUTONOMOUS VEHICLES: A SURVEY

Bharathkumar Hegde*
School of Computer Science and Statistic
Trinity College Dublin
Ireland
hegdeb@tcd.ie

Mélanie Bouroche
School of Computer Science and Statistic
Trinity College Dublin
Ireland
melanie.bouroche@tcd.ie

August 11, 2023

ABSTRACT

Connected Autonomous vehicles (CAVs) are expected to improve the safety and efficiency of traffic by automating driving tasks. Amongst those, lane changing is particularly challenging, as it requires the vehicle to be aware of its highly-dynamic surrounding environment, make decisions, and enact them within very short time windows. As CAVs need to optimise their actions based on a large set of data collected from the environment, Reinforcement Learning (RL) has been widely used to develop CAV motion controllers. These controllers learn to make efficient and safe lane changing decisions using on-board sensors and inter-vehicle communication.

This paper, first presents four overlapping fields that are key to the future of safe self-driving cars: CAVs, motion control, RL, and safe control. It then defines the requirements for a safe CAV controller. These are used firstly to compare applications of Multi-Agent Reinforcement Learning (MARL) to CAV lane change controllers. The requirements are then used to evaluate state-of-the-art safety methods used for RL-based motion controllers. The final section summarises research gaps and possible opportunities for the future development of safe MARL-based CAV motion controllers. In particular, it highlights the requirement to design MARL controllers with continuous control for lane changing. Moreover, as RL algorithms by themselves do not guarantee the level of safety required for such safety-critical applications, it offers insights and challenges to integrate safe RL methods with MARL-based CAV motion controllers.

Keywords Connected and autonomous vehicle (CAV) · Artificial intelligence (AI) · Multi-agent reinforcement learning (MARL) · Safe control · Safe reinforcement learning · Intelligent transportation system (ITS) · Lane change · Lateral control

1 Introduction

Autonomous Vehicles (AVs) are one of the major components of a rapidly developing Intelligent Transportation System (ITS). Recent developments are enabling AVs with various capabilities from driving assistance systems to automating some of the driving tasks. Based on these capabilities, the Society of Automotive Engineers (SAE) classifies the AVs into six levels, in J3016 standards [1]. These levels vary from no driving automation (Level 0) to full driving automation (Level 5) as illustrated in Figure 1.

Currently, in the commercial market, the Tesla Model S has achieved an autonomy level of 2.5 and the Audi A-8 has achieved level 3 in the SAE classification by automating major driving tasks [2]. Companies like Waymo, Cruise, FiveAI, and Oxobatica are focused on building AVs with level 3+ autonomy. Achieving full autonomy (level 5) is

*Corresponding author. E-mail: hegdeb@tcd.ie.

a challenging task as AVs need to be capable of performing all driving tasks safely and efficiently in all kinds of environment.

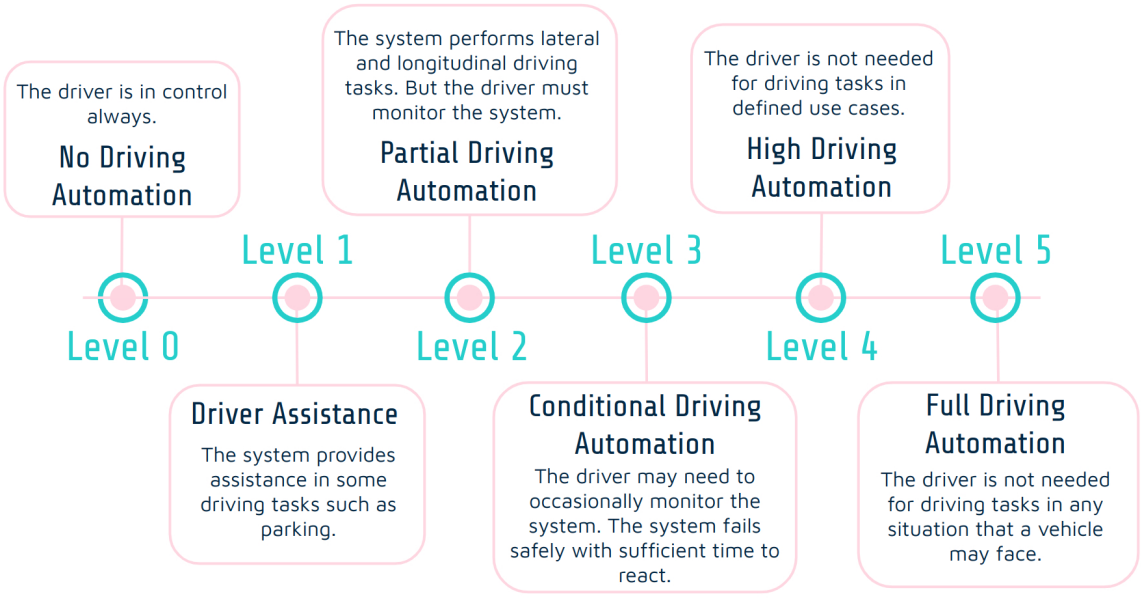


Figure 1: Levels of Automation in vehicles

Developments in communication technology are expected to complement the development of AV technology, and lead to the deployment of Connected and Autonomous Vehicles (CAVs). CAVs are expected to improve the performance of driving tasks required to achieve autonomy of level 3 and above [3]. Recent studies indicate that a motion controller for such CAVs can be designed based on overlapping concepts in the field of motion control, CAVs, Reinforcement Learning (RL), and safe control [4, 5, 6]. These overlapping fields are visualised in Figure 2 and they also define the structure of this survey.

Motion control, which is a branch of automation, uses principles of control theory to formulate the movements of machines [7]. Generally, motion control theories provide definitions of the systems and components involved in autonomous machines. These definitions can be applied to solve control problems in the context of various fields such as robotics, manufacturing, autonomous driving, and many more. In addition, control theories can also be applied to formulate Multi-Agent Systems (MAS) [8], which are being used to design CAV motion controllers.

With the help of vehicle-to-everything (V2X) communication, *CAVs* extend the capabilities of AVs [9]. One such ability is the extended view of the environment. For example, CAVs can build a model of downstream traffic based on the information collected from other vehicles in the environment [10]. Such model can provide useful information about the environment beyond the horizon of AV sensors. Another extended ability of CAVs is the ability to interact with other traffic participants. For example, CAVs can share their intent to enable coordinated motion planning [11]. With the help of coordination, CAVs can find safe trajectories with minimal negative impact on traffic flow [12]. This makes CAVs particularly suited to work collaboratively as a MAS to improve safety and efficiency of traffic [13].

Recent developments in *RL* have proven useful for making fast decisions in dynamic environments with a large set of parameters [14]. This makes RL an ideal candidate for a CAV to learn efficient manoeuvres in dynamic traffic based on a large set of data collected from the surrounding environment. Furthermore, a CAV controller must make intelligent trade-offs between several objectives: safety, but also improving mobility, comfort of travel, fuel efficiency, and reducing emissions. Such trade-offs can be achieved by appropriately formulating a reward function in the RL algorithm. Therefore, RL is a promising option, especially Multi-Agent RL (MARL), for designing a CAV motion controllers as a MAS.

Although MARL can be applied to improve safety of CAVs, it does not provide any guarantees, as agents can enter unsafe states during training and execution. CAV motion controllers, however, are safety-critical applications, as failing to comply with safety requirements can have a fatal impact. Therefore, *Safe control* methods are necessary to impose safety constraints on autonomous controllers. With the increased application of AI to autonomous control systems, safe control methods have gained attention in recent times [15]. These methods, however, have been tested only on simplified AI-based robotic controllers to date.

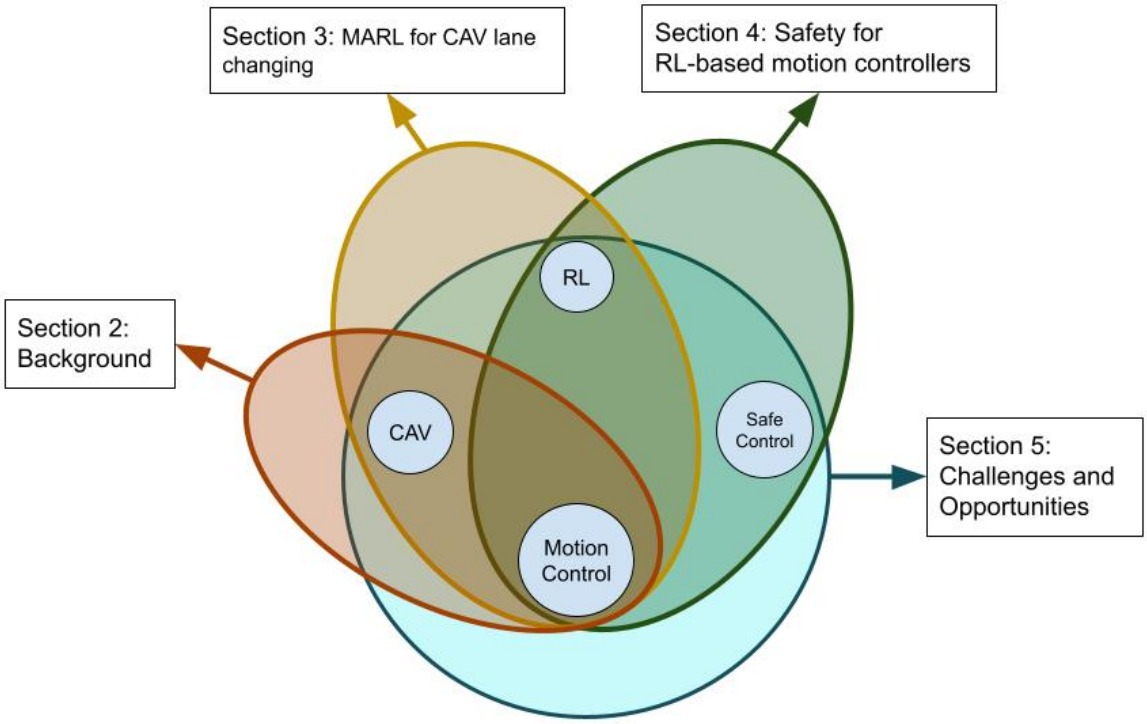


Figure 2: Overlapping disciplines presented in the survey

Overall, all four fields discussed above contribute to the development of a safe and efficient motion controller for CAVs. Generally, motion controllers plan and execute the overall movement of the vehicle, whereas this survey focuses mainly on one of the tasks of motion controllers, lane changing. Our previous paper surveyed AI-based CAV lane change controllers [16], but did not discuss the control methods to achieve safety. Conversely, safe control for RL has been surveyed in Gu et al. [17] and Brunke et al. [15], but the application of these safety methods to CAV controllers was not analysed. To evaluate the applicability of safety methods to CAV lane changing, Lenka et. al defines a set of requirements for safety methods [18], but, these requirements are not intended to evaluate state-of-the-art MARL CAV controllers.

This paper surveys MARL-based CAV motion controllers and safety methods for RL-based controllers, which are overlapping areas of previously discussed key fields, and analyse the applicability of safety methods to AI-based CAV controllers. In addition, it identifies future opportunities to develop a safe CAV motion controller. Thus, the main contributions of this work can be summarised as follows:

- Define the requirements for a CAV motion controller, which can be used to evaluate state-of-the-art CAV controllers and safety methods.
- Provide insights into the recent developments in the design of MARL-based CAV motion controllers and compare them to the requirements defined before.
- Review the methods to guarantee the safety of RL-based motion controllers and analyse their applicability to CAV motion controllers.
- Summarise the challenges and opportunities for designing MARL-based safe CAV motion controllers.

The overlap of the fields discussed above is visualised in Figure 2, which also indicates the organisation of this article. Section 2 provides background details of CAV motion controllers, which combine the concepts of CAVs and motion control. In addition, this section also defines the requirements for safe CAV motion controllers. Next, Section 3 surveys recent applications of MARL motion controllers from the fields of CAVs, motion control, and RL, and evaluates them by comparing them to the requirements specified in the previous section. Then, Section 4 reviews safety methods for RL-based motion controllers and analyses the applicability of the safety methods to CAV motion controllers.

Finally, Section 5 considers all four fields of studies to identify limitations in current CAV controllers and highlight the opportunities created by those limitations.

2 Background

This section establishes the context of this survey and defines a set of requirements for CAV motion controllers. First, it provides an overview of the architecture of a CAV, identifying the various modules that contribute to lane changing. Next, a brief overview of lane changing is provided with a history of the development of lane change models and a discussion of lane change scenarios. Finally, a set of requirements for CAV motion controllers are presented. These requirements provide a reference to compare existing motion controllers and safety methods.

2.1 CAV architecture

The motion controller has the responsibility of controlling the lateral and longitudinal movements of the vehicle. The lateral controller usually makes lane changing decisions and executes the manoeuvres as shown in Figure 3. Lane Change (LC) decisions can be considered as a high-level decision, whereas the execution of manoeuvres can be considered as a low-level control process [19]. At the low-level, longitudinal control is used to stay in the current lane, whereas to change the lane lateral control is used. The longitudinal controller adjusts the speed of the vehicle by regulating its acceleration based on the surrounding environment. The vehicle's lateral movement is executed by adjusting the acceleration and steering angle of the vehicle. Overall, a control flow integrates high-level LC decision and low-level lateral and longitudinal controls to perform autonomous driving tasks.

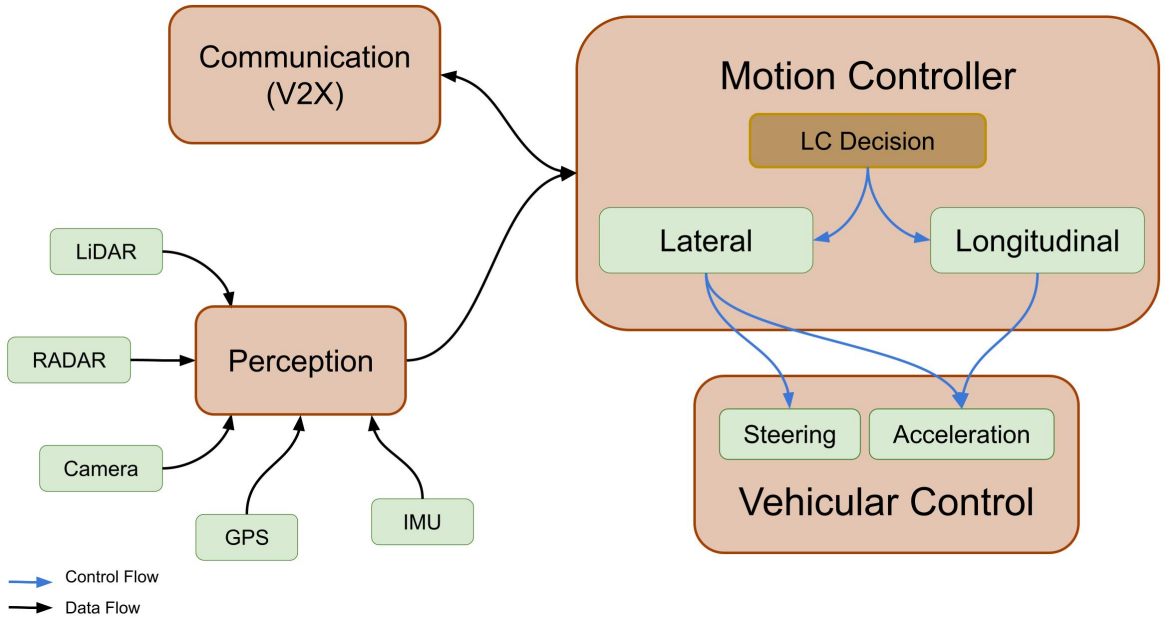


Figure 3: CAV motion controller architecture

The motion controller typically interacts with three major CAV components, namely perception, communication, and vehicular control as shown in Figure 3. The *perception* module creates a perception of the environment around the vehicle by combining inputs from various sensors such as LiDAR, RADAR, camera, GPS, IMU, etc. The perception of the surroundings is used as input to the CAV motion controller to make appropriate decisions. The *communication* module provides V2X interfaces to communicate with other components of the ITS, such as other vehicles, Road Side Units (RSUs), Mobile Edge Computing (MEC) server, cloud server, etc [10]. Using V2X communication, a motion controller can obtain extended information or services from other traffic participants, which could be useful to perform efficient and safe movements. The *Vehicular control* module consists of the physical controls of the vehicle, such as steering and acceleration. These controls act based on the instructions provided by the CAV motion controllers and provide feedback of the execution to the controller [2].

2.2 Lane changing

Among AV driving tasks, changing lanes is one of the most complex and a challenging problem for researchers [20]. It can have a significant impact on traffic, and CAV lane changing can improve the traffic flow at both microscopic and macroscopic level. The macroscopic traffic benefits may include increased safety, traffic efficiency, and road capacity [21], while the microscopic traffic benefits may include increased comfort for travellers with minimal speed variation and reduced travel delays [22]. This section reviews the history of lane changing and the different scenarios in which they are undertaken.

2.2.1 History

A Lane Changing (LC) model encodes a rational decision to change lanes based on various parameters that describe the environment around a vehicle. The first known LC model is the Gipps lane changing model [23]. The Gipps model is based on maintaining a desired speed and being in the correct lane for an upcoming desired manoeuvre. Later, the Gipps model was extended to develop a probabilistic rule-based model to improve realism [24, 25]. Rule-based models consist of a decision process defined in four steps: the decision to consider a lane change, the choice of the target lane, the search for an acceptable gap, and executing the lane change [25]. Next, Kesting et al. proposed a novel incentive-based lane changing model, MOBIL (Minimising Overall Braking Induced by Lane change)[26], which additionally considers the acceleration of the surrounding vehicles to make a lane change decision [25]. One of the recent LC models is LC2013, which considers lane change intentions and uses a decision-tree algorithm to make LC decisions [27]. These models have been used as standard lane changing models in popular traffic simulators and as a baseline to validate recent AI-based motion controllers. A more detailed discussion of recent AI-based motion controllers is presented in Section 3.

2.2.2 Scenarios

The lane change scenario can be defined based on the motive of a vehicle to perform a lane change. The motive to change lane can be broadly categorised as *discretionary lane change*, *mandatory lane change*, and *lane change in bottleneck sections*. The dynamics of vehicle movement and the parameters considered for the lane change decision making differ for each of these categories of lane change. Therefore, the lane change scenario can be one of the factors to consider while designing a motion controller.

An optional lane change by a vehicle, for the benefit of its own or other vehicles in traffic, is considered a *Discretionary Lane Change (DLC)*. DLCs often result in increased speed for the ego vehicles, and they may have various positive impacts on the traffic at the macroscopic level, such as increasing road capacity, increasing traffic throughput, minimising traffic jam propagation, etc. DLCs focus primarily on safety and achieving macroscopic objectives like increased driving comfort, mobility, or throughput [22, 28, 29].

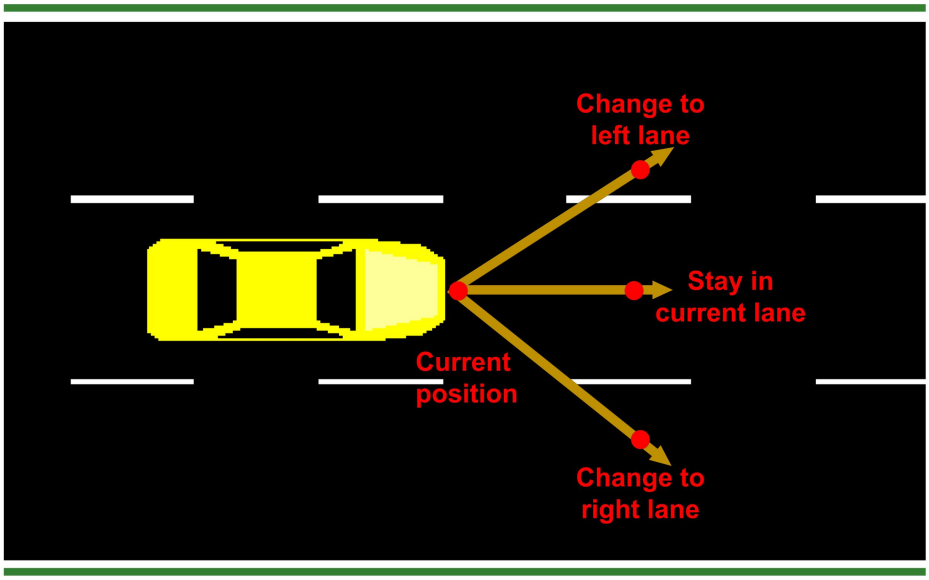
On occasion, a vehicle may be required to change lane to reach a desired destination, such lane changes are classified as *Mandatory Lane Changes (MLCs)*. Some examples of MLCs include changing lane to enter a highway, exit a highway, or before reaching an intersection. Since lane changing is mandatory in these cases, the vehicle may need to execute a riskier lane change, especially in high traffic. An MLC by a vehicle may affect the other vehicles in traffic, therefore, the ideal MLC controller should be capable of ensuring safety even in risky situations and it should have a minimal negative impact on the mainstream traffic flow [30, 31, 32].

Similarly, a vehicle needs to change lane when the current lane reaches a dead end or merges into an adjacent lane. Such lane changes can be categorised as *lane changes in bottleneck sections*. In bottleneck sections, coordination among vehicles plays a key role as the vehicles changing lane will interrupt the main traffic flow. A bottleneck may be created because of construction works, reduced road space, a vehicle broken-down, or accidents. Hence, bottleneck sections are often not known in advance. Therefore, lane changes in bottleneck sections may need to be handled differently compared to an MLC. Typically, the motion controller for bottleneck sections aims to achieve a smooth traffic flow with less congestion, and increase traffic throughput by avoiding stop-go traffic [33, 34].

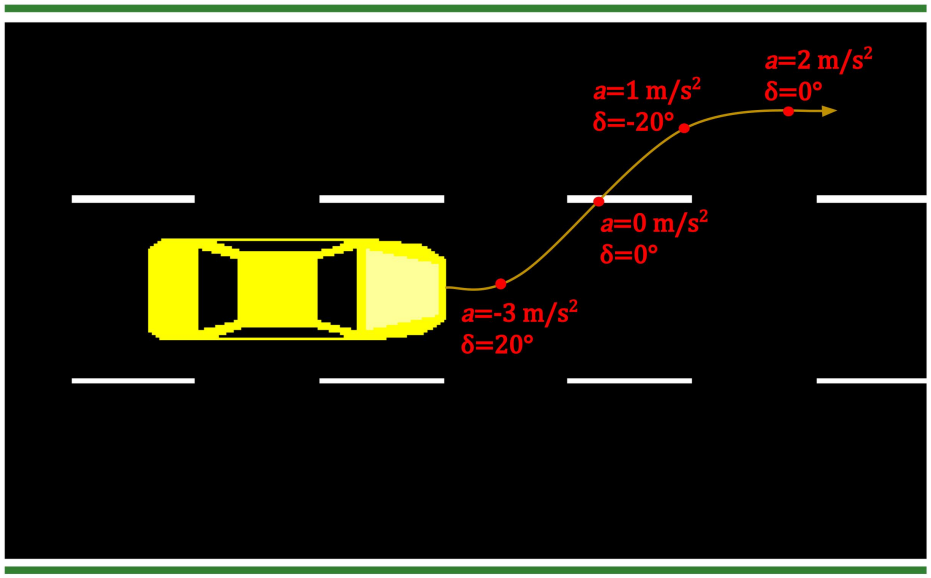
2.3 Requirements for a CAV motion controller

Achieving a completely autonomous AV (SAE's level 5) necessitates a number of requirements to be satisfied and communication can play a key role in achieving them. Additionally, the requirements can be a useful base for comparing MARL controllers and safety methods. Recently, a set of requirements for a safe CAV controller was proposed by Lenka et al. for high-level lane change decision making. These requirements include support for multi-agent interactions, mixed traffic scenario, discrete action space, prior knowledge of unsafe states, and backup policies [18]. Similarly,

the low-level controller of CAVs has some significant requirements in terms of environment, control parameters, deployment, and safety.



(a) Discrete lane change decision



(b) Continuous lane change manoeuvre

Figure 4: Types of lateral control

In terms of the *Environment* in which a CAV has to drive, three main requirements can be observed. Firstly, CAVs should be capable of executing safe manoeuvres in a mixed traffic environment, as CAVs will be introduced alongside human drivers and other vehicles with various levels of autonomy and connectivity. This increases the variability of the situations to be tackled and the complexity of the driving tasks. For example, a CAV may get extended information of the environment when it is surrounded by CAVs with similar communication capability, whereas when it is surrounded by non-communicating vehicles it may have a view of the environment limited to the range of its sensors only. Thus, a CAV controller needs to be designed to handle such complications in mixed traffic environment. Next, a controller should be able to coordinate with other vehicles in a multi-agent environment. A MAS would provide an abstraction of the environment around CAVs to make well-informed control decisions [35]. Moreover, CAV traffic dynamics can be modelled explicitly in multi-agent environments to support control decisions [36]. Similarly, a carefully designed

multi-agent environment of CAVs can enable cooperative driving to achieve macroscopic goals such as improved mobility and stability [32]. For example, CAVs can share their intended trajectories with other vehicles and negotiate the future course of actions to avoid possible conflicts in their path. Therefore, a CAV motion controller should be considered as an agent in a MAS. Finally, a CAV should perform safe lane change manoeuvres in all driving scenarios that a vehicle may face while driving, such as discretionary lane change, mandatory lane change, and lane changes in bottlenecks. Traffic dynamics can be different in each of these scenarios, as explained in subsection 2.2.2. To achieve full autonomy CAVs must learn the dynamics of such scenarios and execute lane change manoeuvres using appropriate control parameters.

The high-level *Control parameters* of a CAV, such as LC decision, should be a discrete variable, as this decision is to either stay in the lane or change to left or right lane as shown in Figure 4a. At the low-level, controllers adjust steering angle (δ) and acceleration (a) to perform manoeuvres. For longitudinal manoeuvres, only acceleration must be controlled, whereas for lateral manoeuvres both steering and acceleration must be controlled. An example trajectory of a continuous lateral manoeuvre with multiple way points is shown in Figure 4b. To follow this trajectory, various values from a continuous domain are considered for acceleration and steering angle.

A CAV controller *Deployment* should be scalable as the number of CAVs on the road increases. The controller can be implemented using either a centralised or decentralised architecture. In a centralised architecture, the motion controller can be placed in an road side unit, an edge server, or a cloud server which can be a centralised controller. A centralised controller can integrate information from traffic participants and use it for trajectory planning and lane changing decisions [37, 31]. Conversely, a decentralised architecture can be implemented by placing the motion controller in individual CAVs. A decentralised controller can communicate through a direct V2V communication interface or through the network infrastructure to collect information from other AVs for trajectory planning and lane change decisions [13]. Although the centralised deployment can support cooperative driving behaviour [31], decentralised deployment is preferred to achieve scalability [20]. Therefore, minimum dependency on the centralised services is suitable for developing scalable motion controller.

A CAV controller must ensure *Safety* while training and in deployment. A MARL-based controller would have to be trained initially in a controlled environment before being deployed. In the training phase CAVs need to drive safely without any collisions, as an accident may damage costly equipments in the vehicle. Furthermore, when a controller is deployed in traffic, accidents can be catastrophic and even cause death. Therefore, a controller needs to avoid any kind of collisions to ensure safety. To quantify safety, Brunke et al. defines three levels of safety [15]. A no safety level (Level 0) can be used to indicate minimal safety. The first level (Level I) only encourages safety as an objective, the second level (Level II) defines safety based on a threshold probability, and the third level (Level III) ensures safety based on hard constraints. So, to ensure safety, a CAV motion controller must comply with Level III safety.

Table 1 summarises the necessary requirements for a level 5 AV to drive safely in the near future.

Table 1: Requirements for a level 5 AV controller

Environment	**Mixed traffic**	Capable of operating in mixed traffic
	Multi-agent	Capable of coordinating manoeuvres with other vehicles
	Driving scenarios	Capable of driving in all the lane change scenarios
Control parameters	**LC Decision**	Discrete decisions
	Longitudinal	Continuous acceleration control
	Lateral	Continuous steering angle and acceleration control
Deployment		Support decentralised deployment with minimal dependency on centralised services
Safety		Zero collisions while training and in deployment

Overall, the above-mentioned requirements make a number of assumptions to avoid the limitations imposed by sensors, the V2X network, and hardware. Firstly, the controller is assumed to receive reliable readings from the sensors, and the surrounding view of the environment constructed based on these readings is accurate. Also, possible faults in sensor readings due to weather conditions are not taken into account. The second assumption is that each CAV is capable of communicating with its surroundings using a dependable V2X connection. The physical limitations of V2X networks,

such as unexpected disconnections, network interference, and packet drops are not considered. Finally, the underlying hardware is expected to accurately execute control commands in real-time. In addition, the limitations of hardware resources, such as CPU and memory, are not considered in this survey.

3 MARL for CAV lane changing

Recent developments in ITS are adopting AI for various tasks such as, traffic predictions, traffic signal control, navigation, and AV control to improve transportation systems [38]. For AV control development, AI has been a popular option to solve some of the most complex problems such as localisation, mapping, perception, route planning, motion control [39]. Specifically for CAV motion controllers that are capable of performing advanced driving tasks such as lane changing, MARL is a popular choice [16]. This section first introduces a generic MARL controller. Next, it provides insights into the recent developments in the design of MARL motion controllers used for CAV lane changing. Finally, the limitations of theses controllers are discussed.

3.1 MARL controllers

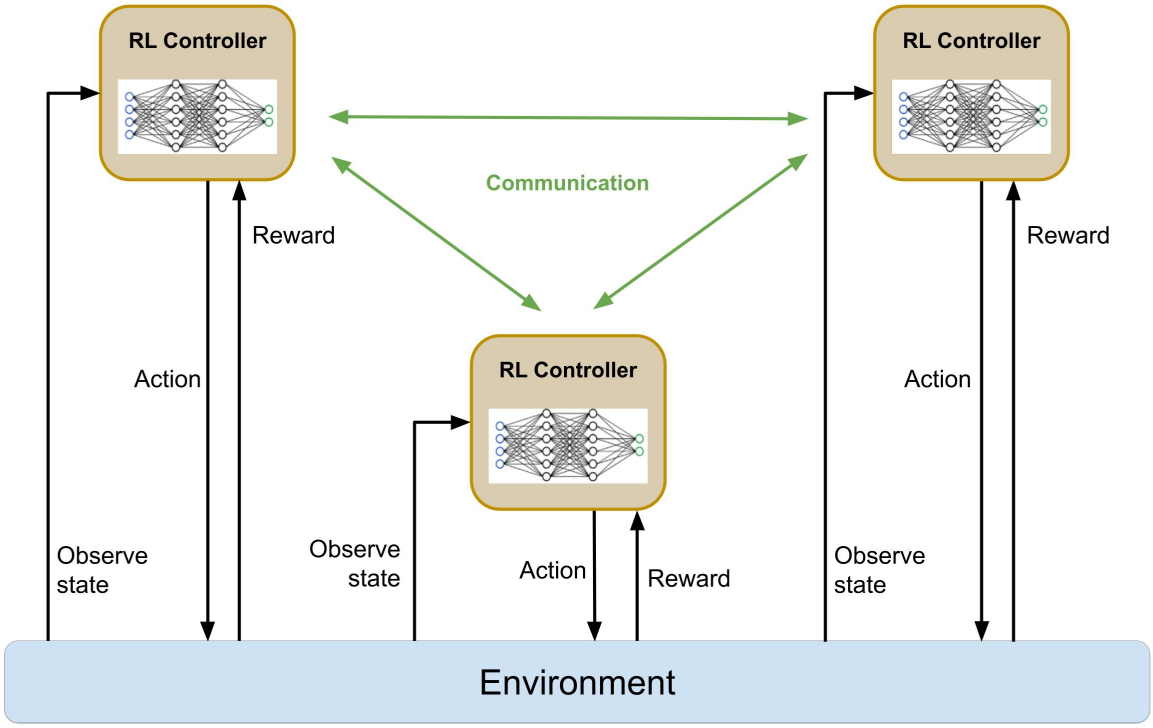


Figure 5: Typical MARL system

A typical decentralised MARL system consists of independent RL agents that interact within the same environment as shown in Figure 5. Such a MAS can be designed by extending Markov Decision Process (MDP), also known as stochastic game [40]. A stochastic game can be formulated by the tuple $\{\mathcal{N}, \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma\}$, where

- $\mathcal{N}$ is number of agents,
- $\mathcal{S}$ is the state space. Each state consists of local states of each agent. Therefore, if the state space of agent $i \in [1, \mathcal{N}]$ is S_i, then the overall state space $\mathcal{S} = S_1 \times S_2 \times ... \times S_{\mathcal{N}}$,
- $\mathcal{A}$ is the joint action space consisting of local action space A_i of an agent $i \in [1, \mathcal{N}]$. Therefore, $\mathcal{A} = A_1 \times A_2 \times ... \times A_{\mathcal{N}}$,
- $\mathcal{P}$: $\mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ is the state transition function that provides the likelihood of changing the overall state of the MAS based on a specific joint action from $\mathcal{A}$,

- $\mathcal{R}: \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$ is a common reward function for all agents. The common reward function can be useful in achieving cooperative goals. The reward depends on the previous state, joint actions, and the current state of the system. The reward function formulation plays an important role in defining the outcome of a MARL system,
- $\gamma \in [0, 1)$ is the discount factor.

The agent i in a MARL system observes the state S^t of the system at the time step t and makes a decision to perform action A_i^t that maximises the reward. Similarly, all the agents define their actions to perform a joint action $\mathcal{A}^t$ to move the system to the next state S^{t+1}, and each agent immediately receives a common reward R^t, evaluated by the reward function $\mathcal{R}$. The goal of a MARL system is to learn a joint optimal policy $\pi^* : \mathcal{S} \rightarrow \mathcal{A}$ to maximise the reward from the system.

One of the simplest policies can be choosing the action that maximises the state value function (Q-function) of each agent i at step t. The value of Q-function for an agent i provides an estimate of the future reward that can be achieved by choosing an action A_i^t, after observing a state S_i^t at step t. This value can be updated at every step to maximise the overall reward from the MARL system. The updates are evaluated based on learning rate α, the common reward R^t, and the expected reward Φ^π after choosing next actions based on a joint policy π as formulated in the below equation [40],

$$Q(S_i^t, A_i^t) \leftarrow (1 - \alpha)\, Q(S_i^t, A_i^t) + \alpha \left(R_t + \gamma\, \Phi^\pi(S_{i+1}^t)\right)$$

The MARL is a useful framework to develop CAV controllers. In the simplified MARL setting explained above, the agents are assumed to observe the state and the actions of other agents in the system. In case of CAVs, this can be achieved by using communication channels (V2X) to collect the required state information from the environment [41]. For example, the state information, such as position and velocity can be collected from neighbouring vehicles via V2V communication channels [42]. Although previously presented MARL framework assumes common reward function for all agents, CAVs may have different goals. Hence, each vehicle may need to define its own reward function to achieve cooperative or competitive behaviour in the system. To accommodate such requirements, the MARL can be designed with different reward functions for each agents or a group of agents [41]. However, this may lead each agent to develop an individual policy which may not achieve the overall traffic goal. To make autonomous driving decisions, CAV controllers can be designed using various formulations of MARL to suit their specific requirements.

3.2 Existing MARL controllers for CAV lane changing

While there have been many applications of MARL for CAV motion controllers, very few of them have addressed lateral movement. These lateral controllers demonstrated the advantages of MARL such as reduced congestion and enhanced safety, driving comfort and fuel efficiency. For example, Ha et al. leveraged topological information from CAVs to mitigate congestion [33]. By aggregating the state information from CAVs, a MARL algorithm developed by Chen et al. prescribed efficient lane changes to improve safety and mobility CAVs [31]. Another algorithm introduced by Zhou et al. aggregates the MARL parameters to achieve fuel efficiency along with improved safety and driving comfort [20]. To review such LC controllers in detail, they can be broadly categorised based on the type of RL algorithm used: Deep-Q Networks (DQNs) or Actor-Critic Networks (ACNs).

3.2.1 Deep Q-Networks

The Q-function, formulated using deep neural networks is known as Deep Q-Networks (DQN). DQNs can be applied to map a high-dimensional input space to a discrete action space, based on a policy π [43]. This makes them suitable for making high-level lane change decisions, which can be discrete, like change lane to the left, to the right, or stay in the same lane. These decisions may depend on a variety of inputs recorded from local sensors and surrounding vehicles [44]. Furthermore, DQNs can be applied to achieve multiple lane changing objectives by using a reward function that accounts for safety, mobility, and comfort [22].

While DQNs are a promising approach for lane change decision making, some challenges still need to be addressed. One of the challenges is that DQNs require inputs of fixed size, but the dynamic state space of the CAVs, leads to inputs of variable size [22]. This challenge can be addressed by encoding the dynamic state space with variable length to a set of parameters with a fixed length. For example, Dong et al. used three neural networks to encode each component of a dynamic state space, which contains the state of a CAV, the states of the surrounding vehicles, and the states of the downstream vehicles [22]. This LC controller, however, does not take advantage of the possibility of collaboration among CAVs. To enable collaboration between CAVs, Graph Convolution Networks (GCNs) can be used

to include topological information about traffic to make collaborative lane change decisions. A GCN is implemented in a centralised unit to encode dynamic input data and topological information to a set of fixed length parameters, which are used as input to a DQN [31]. Conversely, Yu et al. used a decentralised approach to encode the dynamic traffic topology as a Dynamic Coordination Graph (DCG) to achieve collaborative lane change decisions [29].

In summary, DQN-based CAV controllers is a promising option for lane change decision making in CAVs. The implementations of DQNs for CAV lane changing address the limitation of fixed length input and achieve coordination among CAVs using innovative methods. However, these DQN implementations consider only single-step lane changing and do not model continuous controls such as acceleration or speed.

3.2.2 Actor-Critic Network

The Actor-Critic Network (ACN) is an extension of DQNs which implements the Actor-Critic (AC) algorithm [14]. The AC is a type of RL algorithm that consists of policy (Actor) and value (Critic) functions [45]. Policy functions use optimisation methods such as the Deterministic Policy Gradient (DPG) or the Deep DPG (DDPG) to estimate a policy in the continuous action space. Optimisation methods, however, suffer from high variance to estimate the gradient, as a result learning can be slow [45]. On the other hand, value functions use Temporal Difference (TD) learning to reduce variance in the expected return. Hence, the AC algorithm, which combines optimisation method and TD learning, can quickly converge to learn a policy for a continuous action space. Overall, ACNs can provide the combined advantages of AC algorithms and DQNs to design a CAV controller that can handle a large state space and a continuous action space.

Existing ACN implementations aim to achieve a balance between the scalability of the controller and cooperation among CAVs based on the requirements of the lane change scenario. Since ACNs allow learning a policy in a continuous action space, they can be used to adjust the continuous variables of CAV control, such as acceleration or speed, to enable cooperation between CAVs by creating the necessary gaps to allow safe lane changes. Cooperation among CAVs can be enabled by using a centralised LC controller, but this compromises scalability. Conversely, a decentralised LC controller could improve scalability, but a cooperation mechanism has to be implemented explicitly.

For example, an LC controller can implement cooperation among CAVs by using a centralised ACN-based controller to adjust the speed of CAVs in a congested highway bottleneck [33]. Cooperation among CAVs would be necessary in a congested bottleneck scenario as vehicles need to create gaps that allow safe merging of vehicles into the main stream. Cooperation can also be induced among CAVs using a decentralised motion controller. For example, a decentralised LC controller can be used for lane merging in a work zone section. Such a controller can be developed using ACN to adjust the acceleration of the CAV to allow cooperative lane changes in a work zone section [34]. Overall, for CAV lane changes in a work zone section or a bottleneck section, both centralised and decentralised architecture can be used to implement cooperation among CAVs with an ACN-based controller.

For lane changes on a highway or in a weaving section of the highway, a decentralised approach would enable an independent strategy for each vehicle [32]. An example of a decentralised controller for lane changes in a weaving section of a highway is the multi-agent DRL controller proposed by Hou et al., which used ACN to make lane change decisions and speed adjustments to allow cooperation among vehicles [32]. This decentralised controller relies on global state information to make its decisions. As global state information may need to be obtained from an external centralised system, it could compromise the scalability of the controller. On the other hand, a shared ACN can also be used to implement cooperative lane change among CAVs, without compromising scalability. Zhou et al. proposed a cooperative and decentralised LC controller [20]. This controller uses a shared ACN to make lane change decisions and control vehicle speed. Furthermore, the controller achieves cooperation and improved performance compared to the individual ACN implementation. Overall, ACN-based LC controllers that are designed mainly for MLC and DLC in highway traffic can provide scalable cooperation.

While ACN-based LC controllers provide advantages compared to DQN-based LC controllers, they suffer from some limitations. They assume that lane change is executed in a single step and consider the LC controllers as a single concrete module. To overcome these limitations, a modular lane change approach can be used [46]. In the modular lane change approach, the LC controller can be a combination of different methods to achieve the best overall results. Such sub-modules can have their own way of handling a specific task such as lane change decision making, trajectory planning or predicting the probable trajectory of other vehicles, which might add additional benefits to improve the performance of the motion controller. For example, Liao et al. proposed an online model to predict the possibility of lane changes by surrounding vehicles. This model is a combination of two sub-modules [47]. The first sub-module uses a Long-Short Term Memory (LSTM) network, and the second sub-module uses Inverse Reinforcement Learning (IRL) to predict the trajectory of the vehicle. The predictions generated from this module can be used to improve the performance of the LC controller . In general, the modular approach seems to be a promising trend for RL-based LC controllers as it opens up new dimensions to improve their efficiency.

3.3 Limitations

Although MARL controllers have specific advantages, they suffer from some limitations to match the requirements of CAV controller presented in Table 1. Some of these limitations can be observed from Table 2, which lists the properties of the MARL applications discussed. These limitations reveal several challenges that need to be addressed.

Table 2: MARL applications of CAV motion controller for lane changing

RL Method	Reference	Environment		Control parameters		Deployment	Safety
		Mixed traffic	Driving scenarios	Longitudinal	Lateral		
DQN	[29]	No	Discretionary	NA	Discrete	Decentralised	Level I
	[22]	Yes	Discretionary	NA	Discrete	Decentralised	Level I
	[31]	Yes	Mandatory	NA	Discrete	Centralised	Level I
ACN	[33]	Yes	Bottleneck	Continuous	Discrete	Centralised	Level I
	[34]	No	Bottleneck	Continuous	Discrete	Decentralised	Level I
	[20]	Yes	Discretionary	Discrete	Discrete	Decentralised	Level I
	[32]	No	Mandatory	Continuous	Discrete	Decentralised	Level 0
	[47]	Yes	Mandatory	NA	Discrete	Decentralised	Level I

The requirements of the *Environment*, in which a level 5 AV drives, are partially satisfied with some open ended challenges. The discussed multi-agent controllers fulfil the requirement of operating in MAS. In contrast, other requirements of CAV motion controller on operating in a mixed traffic and various driving scenarios are not completely satisfied. Firstly, most MARL motion controllers have considered operation in *Mixed traffic*, though some have left it for future work [32]. The mixed traffic scenario was simulated using the baseline car-following and lane changing models (MOBIL, LC2013) for HDVs in most cases. However, using the same standard driving model for HDVs may not reflect realistic mixed traffic. It is important to design a realistic mixed traffic scenario for simulation that can accurately predict the effect of CAV driving on traffic [48]. Therefore, uncertainties must be considered in HDV models to create a realistic simulation environment with mixed traffic. Secondly, most of these controllers are designed for a specific *Driving scenario*. While some controllers consider a generic approach, their evaluation considers only a single or simplified traffic scenario. In a real-world situation, a CAV may need to perform lane changes in different scenarios in a single journey. Therefore, a practical LC module needs to consider all possible scenarios of lane change in its design.

Next, the *Control parameter* requirements are also satisfied partially. Both longitudinal and lateral controls must be considered as continuous controls. While some of the CAV controllers considered continuous longitudinal controls, lateral controls are considered discrete. Designing a controller with continuous lateral control is more complex compared to discrete lateral control, as such a design requires vehicle movements models, such as kinematic bicycle model [49]. Moreover, to plan vehicle movements on curved roads, mathematical tools such as Frenet frames are necessary [50]. Continuous lateral control has recently been implemented to improve the performance of overtaking in single-agent controllers [51], but its advantages are yet to be realised in multi-agent controllers.

Among the controllers *Deployments*, the decentralised architecture is a popular choice. This could be due to the high-cost and time required to deploy the ITS infrastructure necessary to support centralised controllers [31]. Such ITS infrastructure may include edge servers, roadside units, centralised servers, and other vehicular communication infrastructure. Conversely, a decentralised architecture does not require external infrastructure, though establishing reliable coordination is necessary. Furthermore, achieving coordination among CAVs while driving is a challenging task [13].

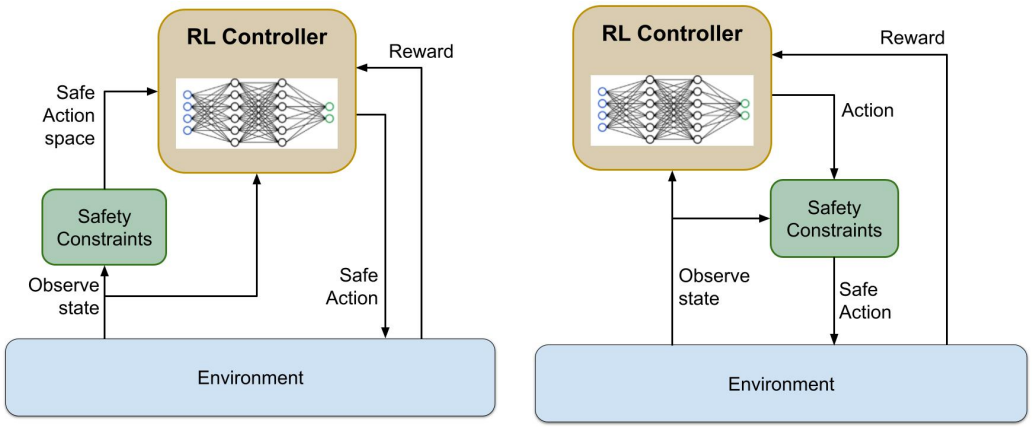
Finally, significant improvements may be required to meet *Safety* requirements. A fully autonomous vehicle needs to be robust and avoid collisions in any situation, i.e., it should comply to level III safety. Existing MARL controllers incorporate safe behaviour only as part of the reward function, which encourages a controller to make safe decisions but does not provide any safety guarantee, thereby only achieving level I safety. Furthermore, the safety of some controllers is only demonstrated on simple driving scenarios. Overall, safe control is still an active research area [17]. Therefore, it is necessary to review recent developments of safety methods for RL controllers and analyse their applicability to CAV motion controllers.

4 Safety for RL-based motion controllers

Although RL-based motion controllers exhibit good performance, they may opt for unsafe actions. This black-box property of RL algorithms limits their applicability to safety critical applications from traffic management to robotic control to autonomous driving [52]. In particular, an AV changing lanes is vulnerable to collisions and therefore it must act safely to avoid serious damage to itself and human lives. Safe actions can be defined as actions that avoid bad or unsafe states, i.e., states violating safety constraints [53]. For example, a vehicle should be restricted from going off the road to ensure safety. The safety constraints can be formulated to identify that the area off the road is unsafe. By integrating the safety constraints in the MARL formulation, an agent can explore the states while cautiously accounting for incomplete knowledge, thus ensuring safety [15]. Recent advances in safe RL have focused on integrating safety constraints into MARL in various ways, which can be broadly classified as shielding, RL customisation, and stability supervisors [17].

4.1 Shielding

To ensure the safety of MARL controllers, an extra layer of software can be added to enforce safety constraint satisfaction. Such a layer can be considered as a safety shield, and methods implementing such shield can be categorised as shielding safety methods. Shielding methods can ensure safety by pre-shielding or post-shielding as shown in Figure 6.



(a) Pre-shielding (b) Post-shielding

Figure 6: Safety by shielding

4.1.1 Pre-shielding

The pre-shielding methods restrict an agent from choosing actions that lead to such unsafe states [54]. Based on the observed state, safety constraints filter the action space to contain only safe actions (see Figure 6a). Hence, an RL agent has to optimise the reward by exploring only the states that are considered as safe. Some of the recent AV controllers that uses pre-shielding are presented below.

Shielding can mask unsafe actions to ensure that a controller remains in safe states. Unsafe actions can be identified as actions that may lead to collisions. Possible collision of the controllers in the environment can be predicted based on the possible overlapping occupancy on a fixed time horizon [54]. Krasowski et al. applied such a set-based prediction to mask unsafe actions for an AV controller and have shown that collisions can be avoided while changing lanes. Their safety evaluation of an action is, however, still based on predictions, therefore only providing safety level II. Moreover, this approach may be difficult to scale for an increased number of controllers in an environment, as it involves estimating the possible trajectories of all other controllers. Finally, this method has been demonstrated for high-level lane change decisions with discrete actions, whereas implementing a similar safety shield for continuous control can be challenging.

Another way to identify unsafe states can be by evaluating the risk of transitioning to a state [55]. Li et al. propose one such method by identifying three risk levels as $\{Dangerous(2), Attentive(1), Safe(0)\}$, for an AV lane change controller. The risk level is evaluated using a probabilistic model, and therefore the safety constraints are satisfied only with some threshold probability, hence ensuring level II safety only. Moreover, this method only considers a simplified traffic environment and simple safety constraints with continuous motion control.

4.1.2 Post-shielding

The post-shielding methods usually re-evaluate or override the actions identified by RL algorithms based on observed state to ensure that a controller remains in a safe state. Safety constraints therefore act as a filter to avoid unsafe actions (see Figure 6b). For this reason, post-shielding methods can provide level III safety.

One of the simplest ways to implement a post-shielding safety layer for RL controllers is demonstrated by Dalal et al. This safety layer maps the output of the controller (actions) to the safe states space [56]. This method allows implementing the safety constraints as simple addition, along with policy gradient methods such as Deep Deterministic Policy Gradient (DDPG) to map a state to a safe state. Therefore, an RL controller is restricted from leaving safe states throughout the optimisation process. While this safety layer ensures level III safety for continuous control with an easy implementation and low computation costs, this method has been demonstrated only in single-agent systems.

The extension of Dalal et al. to MAS was demonstrated by Sheebaelhamd et al. [57]. This extended safety method relaxes some of the conservative assumptions in Dalal et al. to improve the agents' performance. Although this method allows decentralised agents, the safety layer was implemented as a central unit, which maps the states of all the agents to the safe states space at each iteration of the policy. The method provides promising results by achieving level III safety in physics-based MoJuCo simulations.

A safety layer for an RL controller can also be implemented using Control Barrier Functions (CBFs). Wang et al. provide an implementation of CBFs in learning-based motion planners [58]. This safety layer corrects the action of the controller to ensure safety constraint satisfaction. The CBF is used in this method to formulate safety constraints based on longitudinal and lateral movements, speed limits, and control limits. Moreover, the safety layer provides feedback to the controller in the form of a safety reward, which in turn encourages the controller to take safer actions. Results show zero collisions for simulated driving in prerecorded traffic on German highways (the HighD test dataset). The approach was, however, only evaluated on small segments of straight road, and for a single agent.

CBFs have also been applied to a MARL controller to ensure safety in a multi-agent environment. Cai et al. implement CBFs for free-moving decentralised controllers in a bounded space [59]. This implementation consists of both cooperative and non-cooperative CBFs. The cooperative CBFs account for avoiding collisions among agents, whereas the non-cooperative CBFs are used to avoid collisions with obstacles. The notion of non-cooperative CBFs may also be useful to implement safe CAV controllers in mixed-traffic. The evaluation of this method shows promising results with zero collisions, but it only includes a simplified environment. Moreover, this method considers interaction with only two agents, and would therefore need to be significantly extended to be applied to CAV controllers in mixed traffic [18].

4.2 RL Customisation

Generally, an RL algorithm consists of various formulations of an MDP, a value function, a policy gradient, and a reward function. It is possible to customise these formulations by integrating safety constraints. These customisations may improve the safety of the controller, but they do not provide any safety guarantees [15]. Safety constraints can be integrated into RL in many ways including constrained RL, distributional RL, and reachability methods.

4.2.1 Constrained RL

One way of constraining an RL algorithm is using the constrained MDP (CMDP), which is an extension of MDP with predefined constraints [60]. Using CMDP, Chow et al. introduced a Lyapunov-based safe policy optimisation, which can guarantee near-constraint satisfaction, while allowing scope for performance improvements [61]. This method defines a linearised Lyapunov constraint for states that are included for policy update. The method effectively balances performance and constraint satisfaction in MoJuCo simulation tasks with continuous controls.

Another way to customise an RL algorithm can be to divide a policy into a learnable component and hard constraints that are included in the policy but outside of the learning framework [62]. The method was implemented in multi-agent setting with discrete control, though the deployment framework of this safe controller was not clearly explained. Furthermore, this method claims improved performance in terms of smooth trajectory in double merging scenario with safety guarantees based on hard constraints in the policy. However, empirical data was not presented to prove the claims as the experiment involves proprietary software and data [62].

4.2.2 Distributional RL

A typical RL algorithm learns a value function to maximise the expected reward from a given state or transition (state-to-action mapping), whereas distributional RL learns the distribution of the value function [63]. Distributional RL can be used to achieve safety if safety constraints can be integrated as a distributional constraint. For example,

Conditional Value at Risk (CVaR) is a risk metric that can be used to define safety constraints, especially in financial systems [53]. Ma et al. integrated distributional RL and soft ACN to develop a safe RL [64]. An RL controller designed using the proposed distributional soft actor-critic (DSAC) method was evaluated in gym environments such as MoJuCo tasks. Since the method uses ACN, it supports continuous control. Furthermore, the RL agent was evaluated in a single-agent environment, however, it can also be extended to MAS. Furthermore, this method was demonstrated using tasks from MoJuCo and Box2d with physical constraints, however, authors claim that DSAC can accommodate complex safety constraints as well.

Another Safe Distributional Policy Optimisation (SDPO) framework, developed by Zhang et al., allows formulation of safety constraints based on variance, probability of reaching unsafe states, and CVaR [53]. SDPO outperforms state-of-the-art optimisation algorithms in the safety gym, which is a tool to evaluate the safe exploration of RL. Although SDPO was demonstrated in a single agent, it is applicable for multi-agent systems. Furthermore, the optimisation method also supports both continuous and discrete action spaces. Overall, SDPO can provide up to level II safety; as the safety bounds are formulated based on the distribution of the value function, it cannot guarantee safety.

4.2.3 Hamilton-Jacobi Reachability

Hamilton-Jacobi (HJ) reachability is one more way to define the safe states for the controllers. This method uses a pre-computed guaranteed safe set and a safety override controller to keep the system within the safe set [65]. Fisac et al. provides one possible way of integrating the HJ reachability analysis with temporal difference algorithms to build a safe RL controller [66]. This method ensures that a controller stays in safe states while exploring the environment to optimise its policy. As the method focuses on state-wise safety analysis, it can be applied to controllers with continuous or discrete action spaces. HJ reachability, however, involves complex computations, which limits its applicability to high dimensional systems such as CAV driving.

To simplify the computations, Herbert et al. apply HJ reachability analysis for RL with simplified formulations to speed up calculations [65]. These formulations were evaluated using a single-agent simulation of a 10D near-hover quadcopter. Moreover, the HJ reachability was integrated with soft AC, hence it is applicable to controllers with a continuous action set. Overall, HJ reachability can ensure level III safety in an environment with uncertainties.

4.3 Stability Supervisors

One more way to ensure safety of CAV controllers is by having an external supervisor. A supervisor can assess the safety of a given state and certify that it is safe. Such supervisors are called stability certificates, as these safety certificates can be used to stay in the stable (i.e. safe) state set. Another kind of stability supervisors defines a boundary for a set of safe states, which may lead the system to equilibrium state. Such a bounded set of states can be defined as Region of Attraction (ROA). This section reviews these types of stability supervisors in turns.

4.3.1 Stability Certificates

To ensure safe learning, a state can be explored based on its stability certificate. Stability certificates evaluate if the state is safe. Such stability certificates can be developed by encoding safety constrains using some of the tools from control theory.

In the field of control theory, one of the well-known ways to evaluate the stability of a system is the Lyapunov constraint. To identify the boundary of safe states, a Lyapunov function can be formulated as a Neural Network (NN), as demonstrated by Richards et. al. [67]. This method claims to identify the largest possible subset of states that are safe and can be used for exploration of the RL. Furthermore, the Lyapunov NN is designed to allow the possibility of learning efficiently, while ensuring level III safety. As the method certifies the safety of state, it can be applied to both centralised and decentralised deployments. Lyapunov stability is therefore a potentially promising method to ensure safety of RL, however, it has only been demonstrated in simple single-agent simulation of inverted pendulum with continuous action. It may therefore be challenging to extend this method to ensure its applicability for large-scale application such as CAV controllers.

According to Jin et al., another way to implement a stability certification can be regulating the partial derivatives of the policy with respect to the input parameters [68]. Based on a set of numerical bounds on partial gradients of the policy, a set of policies is pre-defined as "safe sets". These bounds are defined in such a way that the stability of the systems is guaranteed, as long as the policy stays within the safe set. Contrary to other stability certification methods, this method is applicable to large-scale systems as well. This method was demonstrated to achieve stability with level III safety in systems such as coordinated flight formation and frequency regulation of the power system. Additionally, this stability certification is applicable to decentralised multi-agent control systems with continuous actions. Evaluating

Table 3: Summary of safety methods for RL controllers

Safety Methods		Reference	Multi-agent Env.	Control parameters	Deployment	Safety
Shielding	Pre-shielding	[55]	No	Continuous	Decentralised	Level II
		[54]	No	Discrete	Decentralised	Level II
	Post-shielding	[56]	No	Continuous	Decentralised	Level III
		[57]	Yes	Continuous	Centalised	Level III
		[58]	No	Continuous	Decentralised	Level III
		[59]	**Yes**	**Continuous**	**Decentralised**	**Level III**
RL Customisation	Constrained RL	[61]	No	Continuous	Decentralised	Level II
		[62]	Yes	Discrete	Not Specified	Level II
	Distributional RL	[64]	No	Continuous	Decentralised	Level II
	HJ Reachability	[53]	No	Continuous	Decentralised	Level II
		[66]	No	Continuous	Decentralised	Level III
		[65]	No	Continuous	Decentralised	Level III
Stability Supervisor	Stability Certificates	[67]	No	Continuous	Centralised	Level III
		[68]	**Yes**	**Continuous**	**Decentralised**	**Level III**
	Region of Attraction	[69]	No	Continuous	Decentralised	Level III

safety based on partial derivatives of a policy might be one of the simplest ways to enable safe lane changing in CAVs. It may, however, be computationally complex as CAV traffic can be much more complex than the systems evaluated in the above method.

4.3.2 Region of Attraction

According to control theory, ROA is a set of states from which a system can converge asymptotically to a fixed point x^*, which defines the equilibrium of a closed loop system [15]. Accurately defining x^* is a challenging task, especially in complex dynamic systems. Zhou et al. define a computationally effective learning framework based on ROA [69]. Using ROA, a predefined corrective controller is used to move the system towards the ROA if the system reaches towards the edges of ROA, otherwise a learning-based controller is used to improve the performance of the controller. Since this method uses an external stability supervisor, in multi-agent setup it can be implemented either in a centralised or decentralised framework. To demonstrate this method, single-agent inverted pendulum simulation and a quadcopter flight with continuous control were used. By definition, the use of ROA can guarantee level III safety. This method, however, does not ensure safety in the learning phase as the proposed controller requires visiting unsafe states to learn them. Thus it requires careful training in simulations for its adaptation to CAVs.

4.4 Limitations

In general, the safety methods reviewed in the previous subsections can ensure safety level II and above, however, some research gaps can be observed between the safety methods summarised in Table 3 and the CAV controller requirements presented in Table 1. To compare these tables, only the relevant parameters from the requirements table are used: support for *multi-agent environment*, continuous *control parameters*, scalable *deployment*, and high level of *safety* guarantees.

From Table 3, it can be observed that only few of the safety methods have demonstrated safety in *multi-agent environment*. Although other methods may be extendable to multi-agent systems, it is likely to require a significant effort to reformulate them and evaluate their applicability in the CAV context.

Conversely, most of the safety methods satisfy the requirement of supporting continuous *control parameters*. Moreover, some of the methods that use discrete parameters can be extended to continuous controls by modifying the definition

of safety constraints [54]. Therefore, the requirements for control parameters may not be a significant variable when choosing a safety method for the lane change controller.

As most of the safety methods are applied in a single-agent system, they can be *deployed* as in either a centralised or decentralised framework in multi-agent systems. The implementation of these methods in a decentralised framework, however, is complicated. For example, the HJ reachability analysis is computationally complex for a single-agent, and is likely to be more complex and difficult to solve in the multi-agent case. Hence, the scalability of these safety methods is yet to be analysed when they are applied in multi-agent scenarios.

Although the safety methods presented can ensure level II safety and above, they require accurate formalisation of the *safety* constraints. In complex applications such as CAV driving, it could be challenging to formulate safety constraints [18]. Therefore, a critical analysis of the compatibility of these safety methods with CAV motion controllers is necessary to integrate them.

While some safety methods have been implemented in AVs, one of them does not consider MAS [58], whereas another multi-agent safety method only considers discrete control parameters [54]. Therefore, they do not fulfil the criteria of CAV motion controller requirements. In contrast, two safety methods are a close match to the requirements. The first is a CBF-based safety method implemented for a decentralised MAS [59]. In this safety method, constraints are formulated for a simple simulation of two agents. Therefore, extending them to the CAV domain can be a challenging task [18] and would require in-depth analysis to formulate a CBF function. The second safety method is a stability supervisor which regulates partial derivatives of a policy to ensure safety [68]. This method is computationally simple and adaptable to large scale systems, however, it requires pre-defining the constraints to define a set of safe policies. The applicability of such predefined safe policies in the CAV context must be further investigated.

5 Challenges and Opportunities

Designing safe CAV controllers requires combining knowledge from the fields of motion control, CAVs, RL, and safe control (Figure 2). In particular, a set of requirements are identified in Table 1. To our knowledge, existing MARL CAV motion controllers and safety methods do not meet these requirements because of many challenges. These challenges present further opportunities.

One of the challenges of applying MARL to CAV motion control is the practicality of assumptions. In particular, the CAV motion controllers reviewed in this paper use MARL for high-level lane change control, i.e. making discrete lane change decisions and controlling acceleration. To practically employ such high-level controllers in a CAV, a low-level lane change controller with continuous steering control is required. Moreover, assumptions on the model of surrounding vehicles in mixed traffic must incorporate variability in making driving decisions similar to human drivers. These dynamics can be designed as part of the multi-agent environment using a formal framework, such as the framework developed by Helleboogh et al., to allow training a safe and robust MARL agent [36]. Another impractical assumption is the consideration of specific driving scenarios, which limits the controller from further practical application. To overcome this limitation, MARL-based controller needs to be developed to perform lane changes in all scenarios. Although these considerations may increase the complexity controller, they could improve the applicability of the MARL controllers in a real vehicle.

The lack of safety guarantees is another major challenge with the application of MARL to CAV controllers. The state-of-the-art MARL controllers, reviewed in Section 3, improve the efficiency of lane changes, however, they only encourage safety (level I) through the reward function. To ensure safety, control theory-based safety methods can be integrated with MARL. This integration may uncover new challenges and opportunities for the development of safe and efficient CAV motion controllers.

Another challenge in integrating existing safety methods with MARL is computational complexity. Existing safety methods are integrated with RL in simplified environments, in both single-agent and multi-agent environments. However, in more complex systems, formulating safety constraints can be challenging, which might limit the applicability of some safety methods. Two safety methods, multi-agent CBF [59] and stability supervisor [68], stand out, as they comply with the requirements of the CAV motion controllers presented in Table 1. These methods provide feasible options for the integration of MARL-based CAV controllers. Furthermore, they are likely to expose more challenges. For example, achieving coordination among CAVs may be difficult, as safety methods also influence driving decisions. Such external influence may also affect the convergence of MARL-based CAV controllers. Hence, the integration of safety methods with MARL requires further analysis to evaluate their applicability to CAV controllers.

In a nutshell, MARL-based CAV motion controllers improve the efficiency of CAVs. However, it does not guarantee vehicle safety, which is one of the primary requirements of vehicle controllers. Safety methods, on the other hand, ensure the safety of RL agents in simplified simulations or control tasks only. Therefore, an open research challenge is to

design a safe MARL for CAV motion controllers that is practical and ensures safety level III to achieve the requirements of the MARL motion controllers specified in Table 1.

Acknowledgements

This work was supported by the SFI Centre for Research Training in Advanced Networks for Sustainable Societies (ADVANCE CRT), Ireland under the Grant number 18/CRT/6222.

References

[1] SAE, "J3016_202104: Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles - SAE International," SAE International, Tech. Rep., Apr. 2021. [Online]. Available: https://www.sae.org/standards/content/j3016_202104/

[2] J. Wang, J. Liu, and N. Kato, "Networking and Communications in Autonomous Driving: A Survey," *IEEE Communications Surveys Tutorials*, vol. 21, no. 2, pp. 1243–1274, 2019, conference Name: IEEE Communications Surveys Tutorials.

[3] SMMT, "Connected and Autonomous Vehicles Position Paper," SMMT, Tech. Rep., Feb. 2017. [Online]. Available: https://www.smmt.co.uk/wp-content/uploads/sites/2/SMMT-CAV-position-paper-final.pdf

[4] F. Ye, S. Zhang, P. Wang, and C.-Y. Chan, "A Survey of Deep Reinforcement Learning Algorithms for Motion Planning and Control of Autonomous Vehicles," in *2021 IEEE Intelligent Vehicles Symposium (IV)*, Jul. 2021, pp. 1073–1080.

[5] H. Shi, Y. Zhou, K. Wu, X. Wang, Y. Lin, and B. Ran, "Connected automated vehicle cooperative control with a deep reinforcement learning approach in a mixed traffic environment," *Transportation Research Part C: Emerging Technologies*, vol. 133, p. 103421, Dec. 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X21004150

[6] F. Berkenkamp, "Safe Exploration in Reinforcement Learning: Theory and Applications in Robotics," Doctoral Thesis, ETH Zurich, 2019, accepted: 2019-10-16T10:53:26Z. [Online]. Available: https://www.research-collection.ethz.ch/handle/20.500.11850/370833

[7] J. Ma, X. Li, and K. K. Tan, *Advanced Optimization for Motion Control Systems*. CRC Press, 2020.

[8] F. Chen and W. Ren, "On the Control of Multi-Agent Systems: A Survey," *Foundations and Trends® in Systems and Control*, vol. 6, no. 4, pp. 339–499, Jul. 2019, publisher: Now Publishers, Inc. [Online]. Available: https://www.nowpublishers.com/article/Details/SYS-019

[9] M. N. Ahangar, Q. Z. Ahmed, F. A. Khan, and M. Hafeez, "A Survey of Autonomous Vehicles: Enabling Communication Technologies and Challenges," *Sensors*, vol. 21, no. 3, p. 706, Jan. 2021, number: 3 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/1424-8220/21/3/706

[10] J. He, K. Yang, and H.-H. Chen, "6G Cellular Networks and Connected Autonomous Vehicles," *arXiv:2010.00972 [cs, eess]*, Oct. 2020, arXiv: 2010.00972. [Online]. Available: http://arxiv.org/abs/2010.00972

[11] M. Tajalli, R. Niroumand, and A. Hajbabaie, "Distributed cooperative trajectory and lane changing optimization of connected automated vehicles: Freeway segments with lane drop," *Transportation Research Part C: Emerging Technologies*, vol. 143, p. 103761, Oct. 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X22001942

[12] A. Sharma and Z. Zheng, "Connected and Automated Vehicles: Opportunities and Challenges for Transportation Systems, Smart Cities, and Societies," in *Automating Cities: Design, Construction, Operation and Future Impact*, ser. Advances in 21st Century Human Settlements, B. T. Wang and C. M. Wang, Eds. Singapore: Springer, 2021, pp. 273–296. [Online]. Available: https://doi.org/10.1007/978-981-15-8670-5_11

[13] P. Shi and B. Yan, "A Survey on Intelligent Control for Multiagent Systems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 1, pp. 161–175, Jan. 2021, conference Name: IEEE Transactions on Systems, Man, and Cybernetics: Systems.

[14] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv:1509.02971 [cs, stat]*, Jul. 2019, arXiv: 1509.02971. [Online]. Available: http://arxiv.org/abs/1509.02971

[15] L. Brunke, M. Greeff, A. W. Hall, Z. Yuan, S. Zhou, J. Panerati, and A. P. Schoellig, "Safe learning in robotics: From learning-based control to safe reinforcement learning," *Annual Review of Control, Robotics,*

and Autonomous Systems, vol. 5, pp. 411–444, 2022, publisher: Annual Reviews. [Online]. Available: https://doi.org/10.1146/annurev-control-042920-020211

[16] B. Hegde and M. Bouroche, "Design of AI-based lane changing modules in connected and autonomous vehicles: a survey," in *Twelfth International Workshop on Agents in Traffic and Transportation*, Vienna, 2022, p. 16. [Online]. Available: http://ceur-ws.org/Vol-3173/7.pdf

[17] S. Gu, L. Yang, Y. Du, G. Chen, F. Walter, J. Wang, Y. Yang, and A. Knoll, "A Review of Safe Reinforcement Learning: Methods, Theory and Applications," Jun. 2022, arXiv:2205.10330 [cs]. [Online]. Available: http://arxiv.org/abs/2205.10330

[18] L. P. Lenka and M. Bouroche, "Safe Lane-Changing in CAVs using External Safety Supervisors : A Review," Dec. 2022, p. 12, conference Name: AICS 2022.

[19] J. Duan, S. Eben Li, Y. Guan, Q. Sun, and B. Cheng, "Hierarchical reinforcement learning for self-driving decision-making without reliance on labelled driving data," *IET Intelligent Transport Systems*, vol. 14, no. 5, pp. 297–305, 2020, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1049/iet-its.2019.0317. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1049/iet-its.2019.0317

[20] W. Zhou, D. Chen, J. Yan, Z. Li, H. Yin, and W. Ge, "Multi-agent reinforcement learning for cooperative lane changing of connected and autonomous vehicles in mixed traffic," *Autonomous Intelligent Systems*, vol. 2, no. 1, p. 5, Mar. 2022. [Online]. Available: https://doi.org/10.1007/s43684-022-00023-5

[21] H. Yu, R. Jiang, Z. He, Z. Zheng, L. Li, R. Liu, and X. Chen, "Automated vehicle-involved traffic flow studies: A survey of assumptions, models, speculations, and perspectives," *Transportation Research Part C: Emerging Technologies*, vol. 127, p. 103101, Jun. 2021. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0968090X21001224

[22] J. Dong, S. Chen, Y. Li, R. Du, A. Steinfeld, and S. Labi, "Space-weighted information fusion using deep reinforcement learning: The context of tactical control of lane-changing autonomous vehicles and connectivity range assessment," *Transportation Research Part C: Emerging Technologies*, vol. 128, p. 103192, Jul. 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X21002084

[23] P. G. Gipps, "A model for the structure of lane-changing decisions," *Transportation Research Part B: Methodological*, vol. 20, no. 5, pp. 403–414, Oct. 1986. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0191261586900123

[24] Q. Yang and H. N. Koutsopoulos, "A Microscopic Traffic Simulator for evaluation of dynamic traffic management systems," *Transportation Research Part C: Emerging Technologies*, vol. 4, no. 3, pp. 113–129, Jun. 1996. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X9600006X

[25] Z. Zheng, "Recent developments and research needs in modeling lane changing," *Transportation Research Part B: Methodological*, vol. 60, pp. 16–32, Feb. 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S019126151300218X

[26] A. Kesting, M. Treiber, and D. Helbing, "General Lane-Changing Model MOBIL for Car-Following Models," *Transportation Research Record*, vol. 1999, no. 1, pp. 86–94, Jan. 2007, publisher: SAGE Publications Inc. [Online]. Available: https://doi.org/10.3141/1999-10

[27] J. Erdmann, "SUMO's Lane-changing model," in *LECTURE NOTES IN CONTROL AND INFORMATION SCIENCES*, M. Behrisch and M. Weber, Eds., vol. 13. Berlin: Springer Verlag, 2015, pp. 105–123. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-319-15024-6_7

[28] Y. Zheng, W. Ding, B. Ran, X. Qu, and Y. Zhang, "Coordinated decisions of discretionary lane change between connected and automated vehicles on freeways: a game theory-based lane change strategy," *IET Intelligent Transport Systems*, vol. 14, no. 13, pp. 1864–1870, 2020, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1049/iet-its.2020.0146. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1049/iet-its.2020.0146

[29] C. Yu, X. Wang, X. Xu, M. Zhang, H. Ge, J. Ren, L. Sun, B. Chen, and G. Tan, "Distributed Multiagent Coordinated Learning for Autonomous Driving in Highways Based on Dynamic Coordination Graphs," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 2, pp. 735–748, Feb. 2020, conference Name: IEEE Transactions on Intelligent Transportation Systems.

[30] Z. Wang, X. Shi, X. Zhao, and X. Li, "Modeling decentralized mandatory lane change for connected and autonomous vehicles: An analytical method," *Transportation Research Part C: Emerging Technologies*, vol. 133, p. 103441, Dec. 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X21004319

[31] S. Chen, J. Dong, P. Y. J. Ha, Y. Li, and S. Labi, "Graph neural network and reinforcement learning for multi-agent cooperative control of connected autonomous vehicles," *Computer-Aided Civil and Infrastructure Engineering*, vol. 36, no. 7, pp. 838–857, 2021, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/mice.12702. [Online]. Available: http://onlinelibrary.wiley.com/doi/abs/10.1111/mice.12702

[32] Y. Hou and P. Graf, "Decentralized Cooperative Lane Changing at Freeway Weaving Areas Using Multi-Agent Deep Reinforcement Learning," *arXiv:2110.08124 [cs]*, Oct. 2021, arXiv: 2110.08124. [Online]. Available: http://arxiv.org/abs/2110.08124

[33] P. Y. J. Ha, S. Chen, J. Dong, R. Du, Y. Li, and S. Labi, "Leveraging the Capabilities of Connected and Autonomous Vehicles and Multi-Agent Reinforcement Learning to Mitigate Highway Bottleneck Congestion," *arXiv:2010.05436 [cs, eess]*, Oct. 2020, arXiv: 2010.05436. [Online]. Available: http://arxiv.org/abs/2010.05436

[34] T. Ren, Y. Xie, and L. Jiang, "Cooperative Highway Work Zone Merge Control Based on Reinforcement Learning in a Connected and Automated Environment," *Transportation Research Record*, vol. 2674, no. 10, pp. 363–374, Oct. 2020, publisher: SAGE Publications Inc. [Online]. Available: https://doi.org/10.1177/0361198120935873

[35] D. Weyns, A. Omicini, and J. Odell, "Environment as a first class abstraction in multiagent systems," *Autonomous Agents and Multi-Agent Systems*, vol. 14, no. 1, pp. 5–30, Feb. 2007. [Online]. Available: https://doi.org/10.1007/s10458-006-0012-0

[36] A. Helleboogh, G. Vizzari, A. Uhrmacher, and F. Michel, "Modeling dynamic environments in multi-agent simulation," *Autonomous Agents and Multi-Agent Systems*, vol. 14, no. 1, pp. 87–116, Feb. 2007. [Online]. Available: https://doi.org/10.1007/s10458-006-0014-y

[37] B. Häfner, V. Bajpai, J. Ott, and G. A. Schmitt, "A Survey on Cooperative Architectures and Maneuvers for Connected and Automated Vehicles," *IEEE Communications Surveys Tutorials*, pp. 1–1, 2021, conference Name: IEEE Communications Surveys Tutorials.

[38] M. Veres and M. Moussa, "Deep Learning for Intelligent Transportation Systems: A Survey of Emerging Trends," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 8, pp. 3152–3168, Aug. 2020, conference Name: IEEE Transactions on Intelligent Transportation Systems.

[39] Y. Ma, Z. Wang, H. Yang, and L. Yang, "Artificial intelligence applications in the development of autonomous vehicles: a survey," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 2, pp. 315–329, Mar. 2020, conference Name: IEEE/CAA Journal of Automatica Sinica.

[40] F. L. D. Silva and A. H. R. Costa, "A Survey on Transfer Learning for Multiagent Reinforcement Learning Systems," *Journal of Artificial Intelligence Research*, vol. 64, pp. 645–703, Mar. 2019. [Online]. Available: https://www.jair.org/index.php/jair/article/view/11396

[41] K. Zhang, Z. Yang, and T. Başar, "Multi-Agent Reinforcement Learning: A Selective Overview of Theories and Algorithms," *arXiv:1911.10635 [cs, stat]*, Apr. 2021, arXiv: 1911.10635. [Online]. Available: http://arxiv.org/abs/1911.10635

[42] K. Zhang, Z. Yang, H. Liu, T. Zhang, and T. Basar, "Fully Decentralized Multi-Agent Reinforcement Learning with Networked Agents," in *Proceedings of the 35th International Conference on Machine Learning*. PMLR, Jul. 2018, pp. 5872–5881, iSSN: 2640-3498. [Online]. Available: https://proceedings.mlr.press/v80/zhang18n.html

[43] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015, number: 7540 Publisher: Nature Publishing Group. [Online]. Available: http://www.nature.com/articles/nature14236

[44] X. Liao, X. Zhao, Z. Wang, K. Han, P. Tiwari, M. J. Barth, and G. Wu, "Game Theory-Based Ramp Merging for Mixed Traffic With Unity-SUMO Co-Simulation," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–12, 2021, conference Name: IEEE Transactions on Systems, Man, and Cybernetics: Systems. [Online]. Available: https://doi-org.elib.tcd.ie/10.1109/TSMC.2021.3131431

[45] I. Grondman, L. Busoniu, G. A. D. Lopes, and R. Babuska, "A Survey of Actor-Critic Reinforcement Learning: Standard and Natural Policy Gradients," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 6, pp. 1291–1307, Nov. 2012, conference Name: IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews).

[46] H. Xu, Y. Zhang, C. G. Cassandras, L. Li, and S. Feng, "A bi-level cooperative driving strategy allowing lane changes," *Transportation Research Part C: Emerging Technologies*, vol. 120, p. 102773, Nov. 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X20306835

[47] X. Liao, Z. Wang, X. Zhao, Z. Zhao, K. Han, P. Tiwari, M. Barth, and G. Wu, "Online Prediction of Lane Change with a Hierarchical Learning-Based Approach," May 2022.

[48] M. Garg, C. Johnston, and M. Bouroche, "Can Connected Autonomous Vehicles really improve mixed traffic efficiency in realistic scenarios?" in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, Sep. 2021, pp. 2011–2018.

[49] P. Polack, F. Altché, B. d'Andréa Novel, and A. de La Fortelle, "The kinematic bicycle model: A consistent model for planning feasible trajectories for autonomous vehicles?" in *2017 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2017, pp. 812–818.

[50] B. Lehmann, H.-J. Günther, and L. Wolf, "A Generic Approach towards Maneuver Coordination for Automated Vehicles," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, Nov. 2018, pp. 3333–3339, iSSN: 2153-0017.

[51] S. Hwang, K. Lee, H. Jeon, and D. Kum, "Autonomous Vehicle Cut-In Algorithm for Lane-Merging Scenarios via Policy-Based Reinforcement Learning Nested Within Finite-State Machine," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–13, 2022, conference Name: IEEE Transactions on Intelligent Transportation Systems. [Online]. Available: https://ieeexplore-ieee-org.elib.tcd.ie/document/9729796

[52] I. Elsayed-Aly, S. Bharadwaj, C. Amato, R. Ehlers, U. Topcu, and L. Feng, "Safe Multi-Agent Reinforcement Learning via Shielding," Feb. 2021, arXiv:2101.11196 [cs]. [Online]. Available: http://arxiv.org/abs/2101.11196

[53] J. Zhang and P. Weng, "Safe Distributional Reinforcement Learning," in *Distributed Artificial Intelligence*, ser. Lecture Notes in Computer Science, J. Chen, J. Lang, C. Amato, and D. Zhao, Eds. Cham: Springer International Publishing, 2022, pp. 107–128.

[54] H. Krasowski, X. Wang, and M. Althoff, "Safe Reinforcement Learning for Autonomous Lane Changing Using Set-Based Prediction," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, Sep. 2020, pp. 1–7.

[55] G. Li, Y. Yang, S. Li, X. Qu, N. Lyu, and S. E. Li, "Decision making of autonomous vehicles in lane change scenarios: Deep reinforcement learning approaches with risk awareness," *Transportation Research Part C: Emerging Technologies*, vol. 134, p. 103452, Jan. 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X21004411

[56] G. Dalal, K. Dvijotham, M. Vecerik, T. Hester, C. Paduraru, and Y. Tassa, "Safe Exploration in Continuous Action Spaces," Jan. 2018, number: arXiv:1801.08757 arXiv:1801.08757 [cs]. [Online]. Available: http://arxiv.org/abs/1801.08757

[57] Z. Sheebaelhamd, K. Zisis, A. Nisioti, D. Gkouletsos, D. Pavllo, and J. Kohler, "Safe Deep Reinforcement Learning for Multi-Agent Systems with Continuous Action Spaces," Aug. 2021, arXiv:2108.03952 [cs]. [Online]. Available: http://arxiv.org/abs/2108.03952

[58] X. Wang, "Ensuring Safety of Learning-Based Motion Planners Using Control Barrier Functions," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4773–4780, Apr. 2022, conference Name: IEEE Robotics and Automation Letters.

[59] Z. Cai, H. Cao, W. Lu, L. Zhang, and H. Xiong, "Safe Multi-Agent Reinforcement Learning through Decentralized Multiple Control Barrier Functions," Mar. 2021, arXiv:2103.12553 [cs]. [Online]. Available: http://arxiv.org/abs/2103.12553

[60] E. Altman, *Constrained Markov Decision Processes*. Routledge, 1999. [Online]. Available: https://www.taylorfrancis.com/books/mono/10.1201/9781315140223/constrained-markov-decision-processes-eitan-altman

[61] Y. Chow, O. Nachum, A. Faust, E. Duenez-Guzman, and M. Ghavamzadeh, "Lyapunov-based Safe Policy Optimization for Continuous Control," Feb. 2019, arXiv:1901.10031 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1901.10031

[62] S. Shalev-Shwartz, S. Shammah, and A. Shashua, "Safe, Multi-Agent, Reinforcement Learning for Autonomous Driving," arXiv, Tech. Rep. arXiv:1610.03295, Oct. 2016, arXiv:1610.03295 [cs, stat] type: article. [Online]. Available: http://arxiv.org/abs/1610.03295

[63] M. G. Bellemare, W. Dabney, and R. Munos, "A Distributional Perspective on Reinforcement Learning," in *Proceedings of the 34th International Conference on Machine Learning*. PMLR, Jul. 2017, pp. 449–458, iSSN: 2640-3498. [Online]. Available: https://proceedings.mlr.press/v70/bellemare17a.html

[64] X. Ma, L. Xia, Z. Zhou, J. Yang, and Q. Zhao, "DSAC: Distributional Soft Actor Critic for Risk-Sensitive Reinforcement Learning," Jun. 2020, number: arXiv:2004.14547 arXiv:2004.14547 [cs]. [Online]. Available: http://arxiv.org/abs/2004.14547

[65] S. Herbert, J. J. Choi, S. Sanjeev, M. Gibson, K. Sreenath, and C. J. Tomlin, "Scalable Learning of Safety Guarantees for Autonomous Systems using Hamilton-Jacobi Reachability," Apr. 2021, arXiv:2101.05916 [cs, eess]. [Online]. Available: http://arxiv.org/abs/2101.05916

[66] J. F. Fisac, N. F. Lugovoy, V. Rubies-Royo, S. Ghosh, and C. J. Tomlin, "Bridging Hamilton-Jacobi Safety Analysis and Reinforcement Learning," in *2019 International Conference on Robotics and Automation (ICRA)*, May 2019, pp. 8550–8556, iSSN: 2577-087X.

[67] S. M. Richards, F. Berkenkamp, and A. Krause, "The Lyapunov Neural Network: Adaptive Stability Certification for Safe Learning of Dynamical Systems," in *Proceedings of The 2nd Conference on Robot Learning*. PMLR, Oct. 2018, pp. 466–476, iSSN: 2640-3498. [Online]. Available: https://proceedings.mlr.press/v87/richards18a.html

[68] M. Jin and J. Lavaei, "Stability-Certified Reinforcement Learning: A Control-Theoretic Perspective," *IEEE Access*, vol. 8, pp. 229 086–229 100, 2020, conference Name: IEEE Access.

[69] Z. Zhou, O. S. Oguz, M. Leibold, and M. Buss, "A General Framework to Increase Safety of Learning Algorithms for Dynamical Systems Based on Region of Attraction Estimation," *IEEE Transactions on Robotics*, vol. 36, no. 5, pp. 1472–1490, Oct. 2020, conference Name: IEEE Transactions on Robotics.