Safe CAV lane changes using MARL and control barrier functions

Bharathkumar Hegde[1], Melanie Bouroche[1]

[1]*School of Computer Science and Statistics, Trinity College Dublin, Ireland*

Abstract

Connected and Autonomous Vehicles (CAVs) are expected to improve road safety and traffic efficiency in the near future. Recently, Multi-Agent Reinforcement Learning (MARL) algorithms have been applied to optimise lane change control decisions to improve the average speed of CAVs. The MARL algorithms, however, are limited by a lack of safety guarantees. Control Barrier Functions (CBFs) have been used for ensuring safety of a Reinforcement Learning (RL) agent performing safety-critical control tasks such as robotic navigation and autonomous driving. In this work, the CBF has been defined for a Multi-Agent System (MAS) of CAVs to ensure safety of a MARL lane change controller with three major contributions. The first is an architecture to integrate the high-level behavioural layer with a safe controller at the low-level motion planning layer. The high-level control layer implements a state-of-the-art MARL lane change controller, while the safe low-level motion planning layer constrains the vehicle to safe states using CBF functions. Secondly, multi-agent actor dependencies are defined to ensure that control decisions are made by CAVs in a specific order. Finally, decentralised CBF constraint formulations are defined to comply with the safety specifications. The proposed design, CBF-CAV, can guarantee safe manoeuvres while executing a behavioural control decision made by the MARL controller.

Keywords

Connected and Autonomous Vehicle (CAV), Lane change, Control Barrier Functions (CBF), Artificial Intelligence (AI), Multi-Agent Reinforcement Learning (MARL), Multi-Agent Systems (MAS), Deep learning (DL), Intelligent Transportation System (ITS)

1. Introduction

As a result of the increasing trend in private vehicle ownership, there are over a billion vehicles in the world's motor fleet currently, and this is expected to continue growing in the near future [1]. This trend is likely to cause increased congestion and road accidents. According to a report from the World Economic Forum (2018), road congestion cost 87 billion dollars to the US economy in 2018 due to loss of productivity [2]. Furthermore, a European Union (EU) report states that around 78% of road crashes are considered to be a result of human errors [3]. To minimise congestion and improve traffic safety, Autonomous Vehicles (AVs) are considered one of the main interventions in Intelligent Transportation Systems (ITS) [4].

AV technologies are evolving with the developments in communication technologies and Artificial Intelligence (AI). Connected Autonomous Vehicles (CAVs) are leveraging recent advancements in vehicular communication (V2X) technologies to make collaborated manoeuvres to improve traffic safety and efficiency. AI has been a popular option to solve some of the complex problems in AV technologies, such as localisation, mapping, perception, route planning, and motion control [5]. For CAV motion controllers specifically, our previous work shows that Multi-Agent Reinforcement Learning (MARL) is a popular choice [6]. Lane changing is one of the complex problems in motion control, as improper lane change may cause a collision that could damage the costly components in AVs or even cause loss of lives. Many forms of MARL using Deep Q-Networks (DQNs) [7, 8, 9], and Actor-Critic Networks (ACN) [10, 11, 12] has been applied for designing lane change controllers. Among them, MARL-CAV [12] is an open-sourced state-of-the-art MARL lane change controller designed for CAVs

ATT'24: Workshop Agents in Traffic and Transportation, October 19, 2024, Santiago de Compostela, Spain

hegdeb@tcd.ie (B. Hegde); melanie.bouroche@tcd.ie (M. Bouroche)
https://www.scss.tcd.ie/Melanie.Bouroche/ (M. Bouroche)
0000-0002-2085-7867 (B. Hegde); 0000-0002-5039-0815 (M. Bouroche)

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

[13]. The MARL-CAV significantly improves traffic efficiency and safety. This approach, however, uses a predication-based priority assignment to avoid collisions and encourage safe behaviour, and therefore safety is not guaranteed, which limits its applicability.

Our previous work identifies that Control Barrier Functions (CBFs) are suitable for ensuring the safety of CAV lane change controllers [13]. CBFs have been recently applied to ensure the safe operation of Reinforcement Learning (RL) based single-agent AV controllers [14]. This CBF implementation demonstrates a longitudinal safety constraint in a simple scenario. The CBF can be formulated by considering dynamic safety constraints relative to the surrounding vehicles [15]. This single-agent CBF, however, assumes that other agents make worst-case decisions. Such a safety constraint results in conservative behaviour, negatively affecting traffic efficiency.

Overall, CAV lane change controllers can be designed using MARL to improve traffic efficiency, but they do not ensure safety. This design aims to integrate the CBF safety constraints with the MARL-based lane change controller [12] to ensure safety by considering multi-agent vehicular dynamics to design safety constraints. The main contributions of this work are:

- The architecture to integrate CBF constraints to the high-level MARL-based lane change controllers (Section 3).
- The structure for defining the dynamics of multi-agent interaction between CAVs (Section 4.2).
- The specifications and formulations of the CBF constraints to ensure the safety of CAVs (Section 4.1 and Section 4.3).

2. Background

The background details related to AV control hierarchy, vehicle dynamics, RL, and CBFs are provided in this section. First, the scope of this research is explained based on control hierarchies. Next, the kinematic bicycle model is explained, and the assumptions related to vehicle dynamics are outlined. Then, notations used for RL formulations are discussed. Finally, a general form of a CBF is defined along with an optimisation problem for evaluating the safe control inputs.

2.1. Hierarchy of control layers

The control decisions of AVs can be separated into four hierarchical levels such as route planning, behavioural layer, motion planning, and local feedback control [16]. The route planning layer first identifies a feasible route to the destination provided by the user using the road network information. The route generated from this layer consists of a sequence of waypoints. While moving along these waypoints, the behavioural layer makes high-level driving decisions such as following a lane, performing a lane change, negotiating at the intersection, or moving in an unstructured environment. The motion planning layer generates reference control actions, such as acceleration and steering, to execute a specific manoeuvre from the high-level decision. In the last layer, a local feedback controller performs necessary actuation, such as steering, throttling, and braking, to follow the control references.

The lane change controller can be developed by engineering high-level behavioural and low-level motion planning layers. Specifically, a behavioural layer can be designed to make discrete decisions to change lanes or follow the current lane. Based on this decision, the motion planning layer can identify the control references to execute the desired driving manoeuvre. Therefore, this article mainly focuses on designing these two layers in the AV control hierarchy.

2.2. Vehicle dynamics

In this article, the kinematic bicycle model is considered to define vehicle dynamics. This model considers the two wheels in the front as one wheel and the same for the back wheels, as illustrated in Figure 1. The distance between the front and back wheels is denoted as V_l. The vehicle's position is defined using x and y, longitudinal and lateral coordinates along the road. The vehicle's velocity (v) is

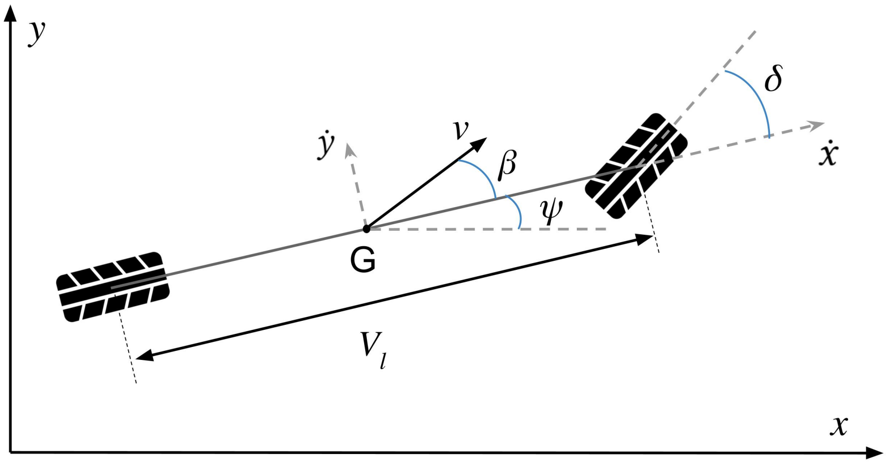


Figure 1: Kinematic bicycle model

controlled by adjusting the acceleration input (u_1), and the steering angle (δ) is controlled by adjusting the steering velocity (u_2). The steering velocity represents the rate of change in steering angle with time [17]. The steering angle is considered the same as the angle of the front wheels with respect to the current heading of the vehicle (ψ). The equations for the kinematic bicycle model that assumes the centre of gravity (G) on the axle with equal distance from the front and back wheels can be written as [18],

$$\begin{gathered} \dot{x} = v_x, \quad \dot{y} = v_y, \quad \dot{v}_x = u_1 \cos(\psi + \beta), \\ \dot{v}_y = u_1 \sin(\psi + \beta), \quad \dot{\psi} = \frac{v}{V_l} \sin\beta, \quad \dot{\delta} = u_2 \end{gathered} \tag{1}$$

where β is a slip angle at the centre of gravity:

$$\beta = \tan^{-1}\left(\frac{1}{2}\tan\delta\right) \tag{2}$$

2.3. Reinforcement learning

RL is a computational approach for learning a sequence of actions to achieve a specific goal. The RL problem is formulated using a Markov Decision Process (MDP) defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$. $\mathcal{S}$ is the state space, the set of state variables that an agent can observe. An agent observes a state $s_t \in \mathcal{S}$ at a time step t. $\mathcal{A}$ is the action space consisting of the set of actions that an agent can perform. At a time step t, an agent performs an action $a_t \in \mathcal{A}$. $\mathcal{P}$: $\mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$ is the state transition function that defines the likelihood of changes in the state observed from the environment based on an action $a_t \in \mathcal{A}$. $\mathcal{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ is a reward function that defines the agent's goal. At a time step t, the agent receives a reward R_t, which is a real number calculated for a transition from the previous state to the current state through an action. The reward function formulation plays a vital role in defining agents' behaviour in the system. $\gamma \in (0, 1]$ is a discount factor used to define the discounted reward G_t,

$$G_t = \sum_{k=t+1}^{\infty} \gamma^k R_k$$

The discounted reward can provide a measure to choose an action that has higher probability of getting better rewards in the future. Using the discounted reward, a state-action value function, also known as the Q-value, can be derived for a policy. A policy π is a mapping from states to the probability of selecting possible actions. The Q-function under policy π provides an expected future reward by choosing action a_t from state s_t. It can be defined as,

$$Q_\pi(s_t, a_t) = \mathbb{E}[G_t|s_t, a_t]$$

For simple problems with a small number of possible states and actions, Q-values can be calculated based on the transition probability $\mathscr{P}$ using the following equation:

$$Q_\pi(s_t, a_t) = \sum_{s_{t+1}} \mathscr{P}(s_t, a_t, s_{t+1})[R_t + \gamma \max_{a_{t+1}} Q_*(s_{t+1}, a_{t+1})]$$

For complex tasks with a large state and action space, such as autonomous driving, it is often very difficult or impossible to model the transition probability $\mathscr{P}$. Therefore, approximation methods are usually used to find a policy to achieve higher rewards in such tasks. These approximations can be implemented using deep neural networks [19]. Deep RL (DRL) approximation algorithms achieved impressive results in playing Atari games [20]. Some of the recent RL approximation algorithms include Deep Q-Networks (DQN) and policy gradient methods, such as Actor-Critic Networks (ACNs). The open-source ACN algorithms such as PPO [21] and ACKTR [22] have been developed and published on repositories like StableBaselines-3 [23] and OpenAI baselines [24]. These algorithms are applied to solve optimisation problems in various research areas, including manufacturing, robotics, large language models, and autonomous vehicles.

The RL algorithms extended to MAS considering various forms of learning and control components are known as MARL [25]. The learning components learn an approximate optimal policy, and the control components execute that policy. These components are integrated into an agent in single-agent tasks such as a robot cleaning the house. Many real-world tasks, however, can be considered to be MAS, as multiple agents may need to work together in the same environment. For example, systems such as multiplayer online games, cooperative robots in factories, traffic control systems, and CAVs can be considered as MAS [26]. However, MARL applications are limited to non-safety-critical tasks, as they can not ensure safety because of the blackbox property [27]. To overcome this limitation, CBF safety constraints can be used.

2.4. Control barrier functions

Consider a discrete time nonlinear control system defined by the following transition dynamics

$$\dot{s} = f(s_t) + g(s_t)u_t \tag{3}$$

where change in state variables $\dot{s}$ per unit time is defined using unactuated dynamics $f : S \to S$, and actuated dynamics $g : S \to \mathbb{R}^{n,m}$, n and m are number of variables in the state space S and action space U respectively, $s_t \in S$ is the system state, and $u_t \in U$ is the control action at time step t. The f and g are defined based on known system dynamics and they are locally Lipschitz continuous, in other words, continuous functions limited by a maximum rate of change. For example, the kinematic bicycle model defined in equation (1) can be defined as a discrete time nonlinear control system (3) as follows,

$$\dot{s} = \begin{bmatrix} v_x \\ v_y \\ 0 \\ 0 \\ \frac{v}{V_l}\sin\beta \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \cos(\psi + \beta) & 0 \\ \sin(\psi + \beta) & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{4}$$

Consider a safe set C defined as the super-level set of the continuously differentiable function $h : S \to \mathbb{R}$

$$C : \{s_t \in S : h(s_t) \geq 0\} \tag{5}$$

To ensure the safety of the control system (3), the safe set C must be forward invariant. When C is forward invariant, safe actions can be defined for each state $s_t \in C$ such that the system continues to stay in C. The safe set C is considered invariant if the function h is a *Control Barrier Function* (CBF) such that there exists $\eta \in [0, 1]$ for all $s_t \in C$ satisfying the following equation (6)

$$\sup_{u_t \in U} \left[h\left(f(s_t) + g(s_t)u_t \right) + (\eta - 1)h(s_t) \right] \geq 0 \tag{6}$$

where η defines the magnitude at which the system is pushed within the safe set C [14]. Using smaller values of η can enforce the constraints strictly, whereas higher values can relax the constraints. Therefore, η represents how strongly the barrier function pushes the states inwards within C. The existence of a CBF implies that for all $s_t \in C$, there exist u_t such that C is forward invariant [28]. Therefore, the goal is to find a minimal safe action u_t^{cbf} that satisfies (6) to ensure the safety of a control system (3).

Let us consider the affine barrier function of the form

$$h(s_t) = p^{\mathrm{T}} s_t + q \tag{7}$$

where $p \in \mathbb{R}^n$ and $q \in \mathbb{R}$ are the parameters used to define a safety constraint h on the state s_t. Combining the affine barrier function with the condition (6), the following constraint can be defined for the control action u_t,

$$-p^{\mathrm{T}} g(s_t) u_t \leq p^{\mathrm{T}} f(s_t) + p^{\mathrm{T}} (\eta - 1) s_t + \eta q \tag{8}$$

To consider multiple safety constraints defined using CBFs, C can be considered as the intersecting half spaces defined by k affine barrier functions [15]. The affine constraint on u_t can be defined by stacking all the constraints.

$$\begin{aligned} & Au_t \leq b, \\ \text{where,} \quad & A = [a_1, a_2, ..., a_k], \text{ with } a_i = -p_i^{\mathrm{T}} g(s_t) \\ & b = [b_1, b_2, ..., b_k], \text{ with } b_i = p_i^{\mathrm{T}} f(s_t) + p_i^{\mathrm{T}} (\eta - 1) s_t + \eta q_i \end{aligned} \tag{9}$$

This constraint can be used to reformulate the CBF given by (6) into the following optimisation problem

$$\begin{aligned} u_t = \arg \min_{u_t} & \|u_t\|_2 \\ \text{s.t} \quad & Au_t \leq b, \end{aligned} \tag{10}$$

which can be efficiently solved in each time step using quadratic program [29].

3. Integrating CBF with MARL for highway merging

In the AV control hierarchy (Section 2.1), the safety constraints can be integrated with the motion planning layer to ensure safe lane change manoeuvres [30]. The safety constraints act as a shield to override the control decisions from the motion planning layer to ensure that a vehicle stays in a safe state [31]. The architecture to integrate the MARL behavioural layer, motion planning layer, and safety constraints is presented in this section to develop a safe MARL lane change controller, illustrated in Figure 2.

3.1. MARL behavioural layer

A vehicle, referred to as the ego vehicle, makes behavioural decisions based on its state information, measured by onboard sensors such as LIDAR, RADAR, Camera, GPS, and IMU, as well as information about the states of the surrounding $\mathcal{N}$ vehicles. The ego vehicle can decide whether to change lanes, follow the lane, speed up, or slow down [12]. Since the ego vehicle can observe vehicles within the range of vehicular communication (V2X), the previously defined MDP (in Section 2.3) can be extended as a Partially Observable MDP(POMDP) for this MARL application. Moreover, V2X is assumed to be a

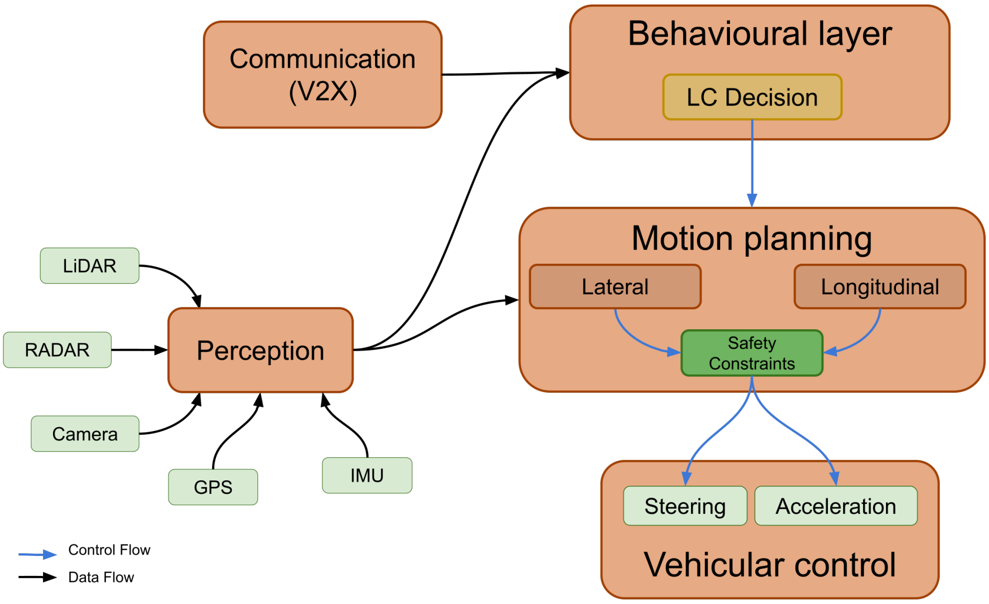


Figure 2: Architecture for integrating MARL with safety constraints

perfect communication interface without any delays or packet drops. The MARL formulation defined in this section is similar to the MARL-CAV formulation, [12], with minor changes in the state space and reward function.

The *state space* $\mathcal{S}_i$ of a vehicle i consists of state variables including,

- x : The longitudinal position of the vehicle.
- y : The lateral position of the vehicle.
- v_x : The longitudinal velocity of the vehicle.
- v_y : The lateral velocity of the vehicle.
- ψ : The vehicle heading with respect to the road.

These variables are observed with respect to a global coordinate system, while observed vehicle state variables are relative to the ego vehicle. As the ego vehicle observes states from $\mathcal{N}$ surrounding vehicles, the overall multi-agent state space is defined as a Cartesian product of the individual states, $\mathcal{S} = \mathcal{S}_0 \times \mathcal{S}_1 \times \mathcal{S}_2 \times \ldots \times \mathcal{S}_{\mathcal{N}}$. The $\mathcal{N} = 5$ is observed to achieve the best performance [12]. We have added the heading ψ to the state variables considered by MARL-CAV to capture the lane change intentions of the CAVs, which ensures safety.

The *action space* $\mathcal{A}$ is the same as defined in MARL-CAV, which consists of five discrete variables representing a specific behaviour, namely, *right lane change*, *left lane change*, *follow lane*, *speed up*, *slow down*. The behavioural layer chooses one of these high-level actions. The low-level controller explained in Section 3.2 further executes these decisions.

The *reward* function constitutes rewards for avoiding collision r_c, maintaining desirable speed r_s, maintaining desirable headway r_h, and feedback from the CBF evaluation r_f along with an associated weight w_* for each reward component. These weights can be tuned to prioritise the CAV objectives. Therefore, the reward for a CAV i at time t is defined as

$$r_{i,t} = w_c r_c + w_s r_s + w_h r_h + w_f r_f$$

The feedback from the CBF evaluation, r_f, is an additional component added to the reward formulation used in MARL-CAV. This can reward the agent for staying in the safe state, which minimises the control overrides required from the CBF layer. Further, this reward encourages the agent to explore within the safe states [14]. Note that the reward $r_{i,t}$ is the reward associated with an individual agent. To achieve collaborative goals, MARL-CAV combines rewards from the surrounding agents to define a local reward as

$$R_{i,t} = \frac{1}{\mathcal{N}} \sum_{j=0}^{\mathcal{N}} r_{j,t}$$

The MARL-CAV is demonstrated with multiple RL algorithms, such as Multi-Agent extensions (MA*) of PPO [21], ACKTR [22], and DQN [20], namely MAPPO, MAACKTR, and MADQN. The multi-agent extension of these algorithms is inspired by the parameter sharing approach proposed in the Multi-Agent Actor Critic (MA2C) RL algorithm [32]. Among them, MAPPO performed best compared to other algorithms in Chen et al. 2023's MARL benchmark analysis [12]. Therefore, this work considers the MAPPO algorithm to train the high-level behavioural layer.

3.2. Motion planning layer

For the low-level motion planning layer, a Proportional-Integral-Derivative (PID) controller can be used to generate control actions, such as acceleration and steering velocity, to execute a behavioural command (defined in Section 3.1). Because of its simplicity, the PID controller can generate control actions in real time. Moreover, it does not require any pre-defined model. While it is possible to integrate the behavioural and motion planning layer using learning-based methods to design end-to-end controllers, they have been criticised for the difficulty in training policies to perform complex tasks [33]. Especially for autonomous driving tasks with dynamic surroundings, end-to-end controllers suffer from poor sample efficiency, resulting in high resource requirements [27]. Another option to integrate the high-level and low-level control layers with learning-based methods is to use hierarchical RL [34]. However, this approach is difficult to reproduce because of the complex training process. Other model-based approaches, such as Model Predictive Controller (MPC), require a model for generating low-level control actions [35]. Estimating such a model for generating CAV control actions in a complex scenario can be difficult. Therefore, the PID controller is a viable option for the low-level control layer along with the high-level MARL controller.

Since the high-level MARL controller is not guaranteed to make safe control decisions, the low-level controller can generate unsafe control actions. The low-level control action u_t^{ll} at time t generated from the PID controller aims to execute the high-level behavioural decision. Therefore, the control action u_t^{ll} must be constrained to ensure safety.

3.3. Safety with CBF shield

Safety of a control system can be ensured by overriding the possibly unsafe lower-level control action u_t^{ll} with a correction u_t^{cbf} to comply with safety constraints defined using CBFs [14]. Therefore, the final control action u_t can be defined as

$$u_t = u_t^{\text{ll}} + u_t^{\text{cbf}} \tag{11}$$

With the updated definition for the action u_t (11), the constraints defined in (9) can be updated to modify the optimisation problem (10) as follows

$$\begin{aligned} u_t^{\text{cbf}} &= \arg\min_{u_t^{\text{cbf}}} \|u_t^{\text{cbf}}\|_2 \\ &\text{s.t} \;\; A u_t^{\text{cbf}} \le b^{\text{ll}}, \\ \text{where } b^{\text{ll}} = [b_1^{\text{ll}}, b_2^{\text{ll}}, ..., b_k^{\text{ll}}], &\;\; \text{with } b_i^{\text{ll}} = p_i^{\mathrm{T}} f(s_t) + p_i^{\mathrm{T}}(\eta - 1)s_t + \eta q_i + p_i^{\mathrm{T}} g(s_t) u_t^{\text{ll}} \end{aligned} \tag{12}$$

In the above optimisation problem (12), u_t^{cbf} is optimised to evaluate the minimal correction required to ensure the safety of the system.

4. Decentralised CBF for CAVs

In the previous sections, the CBF $h(s_t)$ has been defined based only on the ego vehicle's state. The safety constraints defined in this section consider MAS dynamics because CAVs depend on the control decisions of other vehicles to ensure their own safety. These safety constraints are defined for pure CAV traffic. The extension of the safety constraints to mixed CAV traffic, consisting of vehicles with varying levels of autonomy and connectivity, is left for future work. In this section, specifications for decentralised CBFs are defined first (Section 4.1). Then, the multi-agent actor dependencies are defined for CAVs(Section 4.2). In the end, CBF formulations are defined based on the actor dependencies to comply with the specifications (Section 4.3).

4.1. Specifications

The following specifications are considered to formulate decentralised CBFs for CAVs:

1. Ensure the safety of all CAVs in a MAS.
2. Safe acceleration control to avoid collision with the preceding vehicle.
3. Safe steering control to avoid collisions during lane change manoeuvres.
4. Respect the CAV controller's physical limits.

4.2. Multi-agent actor dependencies

As CAVs can share their states with the surrounding vehicles, the state s_t of the ego vehicle constitutes its own ego states, s_t^{e}, and observed states, s_t^{o}, of observed vehicles. With this consideration, a dynamic CBF h for two CAVs can be defined as

$$h(s_t) = h(s_t^{\mathrm{e}}) + h(s_t^{\mathrm{o}}) \tag{13}$$

Notice that each term in the previously-defined optimisation constraint (12) is defined based on the state and the action variables from a single agent. For MAS, each term can be separated into variables associated with the ego vehicle $*^{\mathrm{e}}$ and the observed vehicle $*^{\mathrm{o}}$ as follows

$$A^{\mathrm{e}} u_t^{\mathrm{e}} + A^{\mathrm{o}} u_t^{\mathrm{o}} \leq b^{\mathrm{e}} + b^{\mathrm{o}} \tag{14}$$

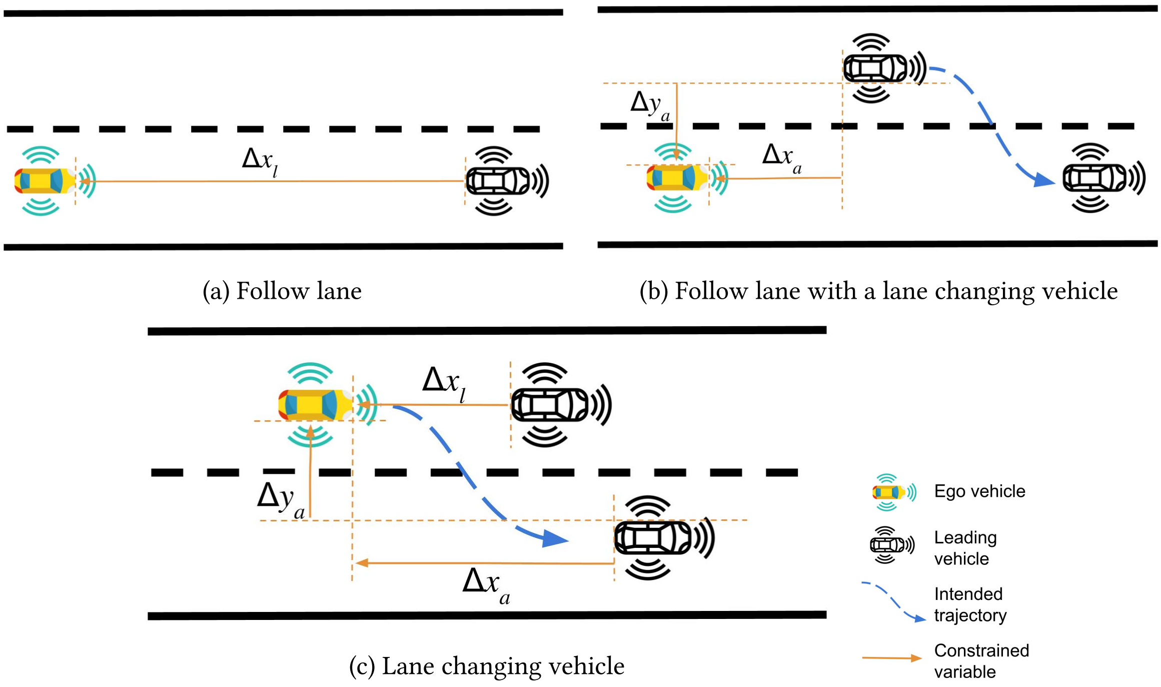
where A^{e} and b^{e} are equivalent to A and b^{ll} (from(12)), but evaluated using the state s_t and the action u_t^{ll} associated with the ego vehicle. Similarly, A^{o} and b^{o} are derived from the state and the action variables associated with the observed vehicle.

The safe action for an ego vehicle can be obtained by optimising the minimum control correction u_t^{e}, assuming that the observed vehicle shares its state variables and safe control decisions. Therefore, the multi-agent constraint in (14) is modified to update the quadratic program (12) for optimising u_t^{e} ,

$$u_t^{\mathrm{e}} = \underset{u_t^{\mathrm{e}}}{\arg\min} \|u_t^{\mathrm{e}}\|_2, \text{ s.t } A^{\mathrm{e}} u_t^{\mathrm{e}} \leq b^{\mathrm{ma}}, \text{and } b^{\mathrm{ma}} = b^{\mathrm{e}} + b^{\mathrm{o}} - A^{\mathrm{o}} u_t^{\mathrm{o}} \tag{15}$$

The actor dependency exists between ego vehicle and observed vehicles as the term b^{ma} in the above equation (15) requires the observed vehicles to make their control decision, u_t^{o}, before the ego vehicle. Based on the ego vehicle's high-level behaviour, the observed vehicles are identified to be considered in CBF constraints.

If the ego vehicle is following a lane, its control action depends on the immediate leading vehicle within the communication range, as illustrated in Figure 3a. In this case, the longitudinal distance between the ego and the leading vehicle Δx_{l} must be constrained to ensure safety. If a vehicle in the adjacent lane intends to change lanes to the ego vehicle's current lane behind the immediate leading vehicle, the ego vehicle's decision depends on the adjacent vehicle, as shown in Figure 3b. In this case, the longitudinal and lateral distances, Δx_{a} and Δy_{a}, with the adjacent vehicle are constrained. As the ego



(a) Follow lane

(b) Follow lane with a lane changing vehicle

(c) Lane changing vehicle

Figure 3: Actor dependency for CAVs with decentralised CBF shield

vehicle's decision depends on the adjacent vehicle's action, this dependency encourages collaborative lane change behaviour among CAVs.

While changing lanes, the ego vehicle depends on the actions of immediate leading vehicles in the current lane and the adjacent target lane. Similar to the previous case, longitudinal distance from the leading vehicle Δx_l is constrained. Moreover, the longitudinal and lateral distances from the adjacent vehicle, Δx_a and Δy_a, are constrained. This dependency is illustrated in Figure 3c.

Collectively, the safety of all the CAVs can be ensured by following the actor dependency as an individual CAV ensures safety with respect to its preceding vehicles. To honour this dependency, CAVs in a MAS are assumed to make control decisions sequentially in the decreasing order of their longitudinal position. Moreover, this actor dependency is applicable only to roads with two lanes. These limitations will be addressed in future work.

4.3. CBF formulation

CBF constraints for CAVs, CBF-CAV, are defined to ensure safe longitudinal and lateral motion without violating the physical control limits of the vehicles. This section defines CBFs for each type of safety constraint, along with the conditions for their applicability.

4.3.1. Longitudinal motion

The safety constraint for longitudinal motion allows the ego vehicle to maintain a safe headway with a preceding vehicle in the current lane. This constraint can be defined as,

$$h_\mathrm{lon} = \Delta x_\mathrm{l} - x_\mathrm{safe} \tag{16}$$

where Δx_l is the longitudinal distance from the rear end of the preceding vehicle and the front of the ego vehicle. The x_safe is the safe distance that must be maintained between two vehicles to ensure that the following vehicle has enough time to slowdown if the leading vehicle start slowing down abruptly. The safe distance threshold can be evaluated from the ego vehicle velocity v_e and time headway τ as shown below,

$$x_\mathrm{safe} = \tau * v_\mathrm{e} \tag{17}$$

Both lane following and lane changing vehicles use this constraint, as the ego vehicle is expected to maintain a safe distance from the leading vehicle in all driving scenarios. Moreover, the leading vehicle must be within the ego vehicle's communication range to enforce this constraint.

4.3.2. Lateral motion

This constraint ensures safety when a CAV moves laterally to change lanes. During the lane change, a vehicle must maintain a safe distance x_{safe} with a leading vehicle in the same lane. This constraint can be enforced with a previously defined CBF (17). As the vehicle moves laterally, either a safe lateral distance y_{safe} or a safe longitudinal distance x_{safe} must be maintained with a leading vehicle in the adjacent lane. This constraint is defined as h_{lat}

$$h_{\text{lat}} = \frac{\Delta x_{\text{a}}}{x_{\text{safe}}} + \frac{\Delta y_{\text{a}}}{y_{\text{safe}}} - 1 \tag{18}$$

where x_{safe} is the same variable defined in equation (17), y_{safe} is a constant defined based on lane width L_w and vehicle width V_w to ensure comfortable lateral distance when the adjacent leading vehicle is moving in parallel.

$$y_{\text{safe}} = L_w - V_w \tag{19}$$

A safe distance can be maintained with a vehicle in the adjacent lane with this constraint. Before changing a lane, this constraint ensures that the ego vehicle maintains a safe lateral distance from the adjacent vehicle. During the lane change, this constraint allows partial violations of lateral and longitudinal constraints while maintaining sufficient distance to avoid collision. During the execution of lane change manoeuvre, the partial violations allow a vehicle to gradually reduce the lateral distance Δy_{a} while gradually increasing the longitudinal distance Δx_{a} with the adjacent vehicle. The gradual increase in longitudinal distance ensures that a safe distance is maintained after completing the lane change manoeuvre. For example, if a vehicle in the adjacent lane is parallel to the ego vehicle, then the lateral safe distance must be maintained, $\Delta y_{\text{a}} \geq y_{\text{safe}}$. In another case, if the adjacent vehicle is about to enter the ego vehicle's lane, then the ego vehicle must gradually increase the longitudinal distance, such that $\Delta x_{\text{a}} \geq x_{\text{safe}}$ when the adjacent vehicle enters the current lane. This constraint is applied to lane changing vehicles and lane following vehicles if they are obstructed by the adjacent leading vehicle changing lanes.

4.3.3. Control limits

Given that vehicle control inputs are subject to physical limitations, they must be constrained. The physical constraints are defined on the steering angle (δ), which is constrained within the range [-$\delta^{\max}$, $\delta^{\max}$]. This physical constraint is defined using two CBFs $h_\delta^{\max}$ and $h_\delta^{\min}$. Note that the steering angle is one of the state variables, hence the constraints are defined using CBFs.

$$\begin{aligned} h_\delta^{\max} &= \delta - \delta^{\max} \\ h_\delta^{\min} &= -\delta - \delta^{\max} \end{aligned} \tag{20}$$

The acceleration range of the vehicle is defined as $[-u_1^{\max}, u_1^{\max}]$. This physical constraint can be enforced by including it in the constraints of the optimisation problem defined in (12).

$$-u_1^{\max} \leq u_1 \leq u_1^{\max} \tag{21}$$

These constraints are applied to all CAVs, as they must be honoured in all scenarios.

5. Conclusion

The proposed CBF formulations, CBF-CAV, integrated with the behavioural layer with MARL lane change controller can have minimal impact on the efficiency, because the CBF constraints override the actions only when a vehicle is about to go towards unsafe states. Moreover, by restricting agents to the safe states, the MARL controller can be trained efficiently by exploring the safe states only. Furthermore, the actor dependencies are used to consider MAS dynamics in the CAV traffic. These constraints ensure both lateral and longitudinal safe motions for all CAVs in a traffic scenario.

The provided formulations are suitable for pure CAV traffic, where a lower safe distance can be used. In the future, this can be extended to mixed traffic with dynamic CBFs to maintain higher safe distances with human driven vehicles, assuming they take worst-case control decisions.

Acknowledgments

The authors wish to thank the editors and anonymous reviewers for their valuable comments and helpful suggestions which greatly improved the paper's quality. 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] WHO, Global status report on road safety 2023, Technical Report, WHO, 2023. URL: https://www.who.int/publications-detail-redirect/9789240086517.

[2] WEF, Traffic congestion cost the US economy nearly $87 billion in 2018, Technical Report, World Economic Forum, 2019. URL: https://www.weforum.org/agenda/2019/03/traffic-congestion-cost-the-us-economy-nearly-87-billion-in-2018/.

[3] E. Commission, Road safety in the EU: fatalities below pre-pandemic levels but progress remains too slow, European Commission - European Commission (2023). URL: https://ec.europa.eu/commission/presscorner/detail/en/ip_23_953.

[4] E. Commission, D.-G. for Mobility andTransport, Next steps towards 'Vision Zero' – EU road safety policy framework 2021-2030, Publications Office, 2020. URL: https://data.europa.eu/doi/10.2832/391271. doi:doi/10.2832/391271.

[5] Y. Ma, Z. Wang, H. Yang, L. Yang, Artificial intelligence applications in the development of autonomous vehicles: a survey, IEEE/CAA Journal of Automatica Sinica 7 (2020) 315–329. doi:10.1109/JAS.2020.1003021, conference Name: IEEE/CAA Journal of Automatica Sinica.

[6] B. Hegde, 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. URL: http://ceur-ws.org/Vol-3173/7.pdf.

[7] C. Yu, X. Wang, X. Xu, M. Zhang, H. Ge, J. Ren, L. Sun, B. Chen, G. Tan, Distributed Multiagent Coordinated Learning for Autonomous Driving in Highways Based on Dynamic Coordination Graphs, IEEE Transactions on Intelligent Transportation Systems 21 (2020) 735–748. doi:10.1109/TITS.2019.2893683, conference Name: IEEE Transactions on Intelligent Transportation Systems.

[8] J. Dong, S. Chen, Y. Li, R. Du, A. Steinfeld, 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 128 (2021) 103192. URL: https://www.sciencedirect.com/science/article/pii/S0968090X21002084. doi:10.1016/j.trc.2021.103192.

[9] S. Chen, J. Dong, P. Y. J. Ha, Y. Li, S. Labi, Graph neural network and reinforcement learning for multi-agent cooperative control of connected autonomous vehicles, Computer-Aided Civil and Infrastructure Engineering 36 (2021) 838–857. URL: http://onlinelibrary.wiley.

com/doi/abs/10.1111/mice.12702. doi:10.1111/mice.12702, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/mice.12702.

[10] W. Zhou, D. Chen, J. Yan, Z. Li, H. Yin, W. Ge, Multi-agent reinforcement learning for cooperative lane changing of connected and autonomous vehicles in mixed traffic, Autonomous Intelligent Systems 2 (2022) 5. URL: https://doi.org/10.1007/s43684-022-00023-5. doi:10.1007/s43684-022-00023-5.

[11] Y. Hou, P. Graf, Decentralized Cooperative Lane Changing at Freeway Weaving Areas Using Multi-Agent Deep Reinforcement Learning, arXiv:2110.08124 [cs] (2021). URL: http://arxiv.org/abs/2110.08124, arXiv: 2110.08124.

[12] D. Chen, M. R. Hajidavalloo, Z. Li, K. Chen, Y. Wang, L. Jiang, Y. Wang, Deep Multi-Agent Reinforcement Learning for Highway On-Ramp Merging in Mixed Traffic, IEEE Transactions on Intelligent Transportation Systems (2023) 1–16. doi:10.1109/TITS.2023.3285442, conference Name: IEEE Transactions on Intelligent Transportation Systems.

[13] B. Hegde, M. Bouroche, Multi-agent reinforcement learning for safe lane changes by connected and autonomous vehicles: A survey, AI Communications 37 (2024) 203–222. URL: https://content.iospress.com/articles/ai-communications/aic220316. doi:10.3233/AIC-220316, publisher: IOS Press.

[14] R. Cheng, G. Orosz, R. M. Murray, J. W. Burdick, End-to-End Safe Reinforcement Learning through Barrier Functions for Safety-Critical Continuous Control Tasks, Proceedings of the AAAI Conference on Artificial Intelligence 33 (2019) 3387–3395. URL: https://ojs.aaai.org/index.php/AAAI/article/view/4213. doi:10.1609/aaai.v33i01.33013387, number: 01.

[15] X. Wang, Ensuring Safety of Learning-Based Motion Planners Using Control Barrier Functions, IEEE Robotics and Automation Letters 7 (2022) 4773–4780. doi:10.1109/LRA.2022.3152313, conference Name: IEEE Robotics and Automation Letters.

[16] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, E. Frazzoli, A Survey of Motion Planning and Control Techniques for Self-Driving Urban Vehicles, IEEE Transactions on Intelligent Vehicles 1 (2016) 33–55. doi:10.1109/TIV.2016.2578706, conference Name: IEEE Transactions on Intelligent Vehicles.

[17] A. De Luca, G. Oriolo, C. Samson, Feedback control of a nonholonomic car-like robot, in: M. Thoma, J. P. Laumond (Eds.), Robot Motion Planning and Control, volume 229, Springer Berlin Heidelberg, Berlin, Heidelberg, 1998, pp. 171–253. URL: http://link.springer.com/10.1007/BFb0036073. doi:10.1007/BFb0036073.PDFfoundinhttps://www.di.ens.fr/jean-paul.laumond/promotion/chap4.pdf, series Title: Lecture Notes in Control and Information Sciences.

[18] M. Althoff, M. Koschi, S. Manzinger, CommonRoad: Composable benchmarks for motion planning on roads, in: 2017 IEEE Intelligent Vehicles Symposium (IV), 2017, pp. 719–726. URL: https://ieeexplore.ieee.org/document/7995802. doi:10.1109/IVS.2017.7995802.

[19] R. S. Sutton, A. G. Barto, Reinforcement learning: An introduction, second edition ed., MIT press, 2018. Edition: Second edition. Publisher: MIT Press,.

[20] 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, D. Hassabis, Human-level control through deep reinforcement learning, Nature 518 (2015) 529–533. URL: http://www.nature.com/articles/nature14236. doi:10.1038/nature14236, number: 7540 Publisher: Nature Publishing Group.

[21] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal Policy Optimization Algorithms, 2017. URL: http://arxiv.org/abs/1707.06347. doi:10.48550/arXiv.1707.06347, arXiv:1707.06347 [cs].

[22] Y. Wu, E. Mansimov, R. B. Grosse, S. Liao, J. Ba, Scalable trust-region method for deep reinforcement learning using Kronecker-factored approximation, in: Advances in Neural Information Processing Systems, volume 30, Curran Associates, Inc., 2017.

[23] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, N. Dormann, Stable-Baselines3: Reliable Reinforcement Learning Implementations, Journal of Machine Learning Research 22 (2021) 1–8. URL: http://jmlr.org/papers/v22/20-1364.html.

[24] P. Dhariwal, C. Hesse, O. Klimov, A. Nichol, M. Plappert, A. Radford, J. Schulman, S. Sidor, Y. Wu, P. Zhokhov, OpenAI Baselines, 2017. URL: https://github.com/openai/baselines, publication Title: GitHub repository.

[25] J. K. Terry, N. Grammel, S. Son, B. Black, Parameter Sharing For Heterogeneous Agents in Multi-Agent Reinforcement Learning, 2022. URL: http://arxiv.org/abs/2005.13625. doi:10.48550/arXiv.2005.13625, arXiv:2005.13625 [cs, stat].

[26] T. T. Nguyen, N. D. Nguyen, S. Nahavandi, Deep Reinforcement Learning for Multiagent Systems: A Review of Challenges, Solutions, and Applications, IEEE Transactions on Cybernetics 50 (2020) 3826–3839. doi:10.1109/TCYB.2020.2977374, conference Name: IEEE Transactions on Cybernetics.

[27] S. Teng, X. Hu, P. Deng, B. Li, Y. Li, Y. Ai, D. Yang, L. Li, Z. Xuanyuan, F. Zhu, L. Chen, Motion Planning for Autonomous Driving: The State of the Art and Future Perspectives, IEEE Transactions on Intelligent Vehicles 8 (2023) 3692–3711. doi:10.1109/TIV.2023.3274536, conference Name: IEEE Transactions on Intelligent Vehicles.

[28] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, P. Tabuada, Control Barrier Functions: Theory and Applications, in: 2019 18th European Control Conference (ECC), 2019, pp. 3420–3431. URL: https://ieeexplore.ieee.org/abstract/document/8796030. doi:10.23919/ECC.2019.8796030.

[29] S. P. Boyd, L. Vandenberghe, Convex optimization, version 29 ed., Cambridge University Press, Cambridge New York Melbourne New Delhi Singapore, 2004.

[30] J. Li, L. Sun, J. Chen, M. Tomizuka, W. Zhan, A Safe Hierarchical Planning Framework for Complex Driving Scenarios based on Reinforcement Learning, in: 2021 IEEE International Conference on Robotics and Automation (ICRA), 2021, pp. 2660–2666. URL: https://ieeexplore.ieee.org/abstract/document/9561195. doi:10.1109/ICRA48506.2021.9561195, iSSN: 2577-087X.

[31] I. Elsayed-Aly, S. Bharadwaj, C. Amato, R. Ehlers, U. Topcu, L. Feng, Safe Multi-Agent Reinforcement Learning via Shielding, 2021. URL: http://arxiv.org/abs/2101.11196. doi:10.48550/arXiv.2101.11196, arXiv:2101.11196 [cs].

[32] K. Lin, R. Zhao, Z. Xu, J. Zhou, Efficient Large-Scale Fleet Management via Multi-Agent Deep Reinforcement Learning, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD '18, Association for Computing Machinery, New York, NY, USA, 2018, pp. 1774–1783. URL: http://doi.org/10.1145/3219819.3219993. doi:10.1145/3219819.3219993.

[33] A. Tampuu, T. Matiisen, M. Semikin, D. Fishman, N. Muhammad, A Survey of End-to-End Driving: Architectures and Training Methods, IEEE Transactions on Neural Networks and Learning Systems 33 (2022) 1364–1384. doi:10.1109/TNNLS.2020.3043505, conference Name: IEEE Transactions on Neural Networks and Learning Systems.

[34] J. Duan, S. Eben Li, Y. Guan, Q. Sun, B. Cheng, Hierarchical reinforcement learning for self-driving decision-making without reliance on labelled driving data, IET Intelligent Transport Systems 14 (2020) 297–305. URL: https://onlinelibrary.wiley.com/doi/abs/10.1049/iet-its.2019.0317. doi:10.1049/iet-its.2019.0317.

[35] Z. Zhao, Z. Wang, G. Wu, F. Ye, M. J. Barth, The State-of-the-Art of Coordinated Ramp Control with Mixed Traffic Conditions, in: 2019 IEEE Intelligent Transportation Systems Conference (ITSC), 2019, pp. 1741–1748. doi:10.1109/ITSC.2019.8917067.