Automatica 162 (2024) 111530

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Optimal control of connected automated vehicles with event/self-triggered control barrier functions☆

Ehsan Sabouni [a,*], Christos G. Cassandras [a], Wei Xiao [b], Nader Meskin [c]

[a] Division of Systems Engineering, Boston University, Brookline, MA, USA
[b] Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, USA
[c] Department of Electrical Engineering College of Engineering, Qatar University, Doha, Qatar

ARTICLE INFO

Article history:
Received 22 September 2022
Received in revised form 22 September 2023
Accepted 20 December 2023
Available online 25 January 2024

Keywords:
Optimal control
Connected and automated vehicles
Event-triggered control
Self-triggered control

ABSTRACT

We address the problem of controlling Connected and Automated Vehicles (CAVs) in conflict areas of a traffic network subject to hard safety constraints. It has been shown that such problems can be solved through a combination of tractable optimal control problem formulations and the use of Control Barrier Functions (CBFs) that guarantee the satisfaction of all constraints. These solutions can be reduced to a sequence of Quadratic Programs (QPs) which are efficiently solved on-line over discrete time steps. However, the feasibility of each such QP cannot be guaranteed over every time step. To overcome this limitation, we develop both an event-triggered approach and a self-triggered approach such that the next QP is triggered by properly defined events. We show that both approaches, each in a different way, eliminate infeasible cases due to time-driven inter-sampling effects, thus also eliminating the need for selecting the size of time steps. Simulation examples are included to compare the two new schemes and to illustrate how overall infeasibilities can be significantly reduced while at the same time reducing the need for communication among CAVs without compromising performance.

1. Introduction

The emergence of Connected and Automated Vehicles (CAVs) along with new traffic infrastructure technologies (Fagnant & Kockelman, 2015; Li, Wen, & Yao, 2014) over the past decade have brought the promise of resolving long-lasting problems in transportation networks such as accidents, congestion, and unsustainable energy consumption along with environmental pollution (de Waard, Dijksterhuis, & Brookhuis, 2009; Kavalchuk, Kolbasov, Karpukhin, Terenchenko, et al., 2020; Schrank, Eisele, Lomax, & Bak, 2015). Meeting this goal heavily depends on effective traffic management, specifically at the bottleneck points of a transportation network such as intersections, roundabouts, and merging roadways (Berg & Verhoef, 2016).

To date, both centralized and decentralized methods have been proposed to tackle the control and coordination problem of CAVs in conflict areas; an overview of such methods may be found in Rios-Torres and Malikopoulos (2017). Platoon formation (Rajamani, Tan, Law, & Zhang, 2000; Wu, Yan, & Abbas-Turki, 2013; Xu, Feng, Zhang, & Li, 2019) and reservation-based methods (Au & Stone, 2010; Dresner & Stone, 2004; Zhang, De La Fortelle, Zhang, & Wu, 2013) are among the centralized approaches, which are limited by the need for powerful central computation resources and are typically prone to disturbances and security threats. In contrast, in decentralized methods each CAV is responsible for its own on-board computation with information from other vehicles limited to a set of neighbors (Rios-Torres, Malikopoulos, & Pisu, 2015). Constrained optimal control problems can then be formulated with objectives usually involving minimizing acceleration or maximizing passenger comfort (measured as the acceleration derivative or jerk), or jointly minimizing travel time through conflict areas and energy consumption. These problems can be analytically solved in some cases, e.g., for optimal merging (Xiao & Cassandras, 2021) or crossing a signal-free intersection (Zhang & Cassandras, 2019). However, obtaining such solutions becomes computationally prohibitive for real-time applications when an optimal trajectory involves multiple constraints becoming active. Thus, on-line control methods such as Model Predictive Control (MPC) (Holkar & Waghmare, 2010) techniques or Control Barrier Functions (CBFs) (Ames et al., 2019; Ames, Xu, Grizzle, & Tabuada, 2017; Xiao & Belta, 2019) are often adopted for the handling of additional constraints.

☆ The material in this paper was not presented at any conference. This paper was recommended for publication in revised form by Associate Editor Antonella Ferrara under the direction of Editor Thomas Parisini.

* Corresponding author.
E-mail addresses: esabouni@bu.edu (E. Sabouni), cgc@bu.edu (C.G. Cassandras), weixy@mit.edu (W. Xiao), nader.meskin@qu.edu.qa (N. Meskin).

https://doi.org/10.1016/j.automatica.2024.111530
0005-1098/

In the MPC approach proposed in Garcia, Prett, and Morari (1989), the time is normally discretized and an optimization problem is solved at each time instant with the addition of appropriate inequality constraints; then, the system dynamics are updated. Since both control and state are considered as decision variables in the optimization problem, MPC is very effective for problems with simple (usually linear or linearized) dynamics, objectives, and constraints (Cao, Mukai, Kawabe, Nishira, & Fujiki, 2015). Alternatively, CBFs can overcome some shortcomings of the MPC method (Holkar & Waghmare, 2010) as they do not need states as decision variables, instead mapping state constraints onto new ones that only involve the decision variables in a linear fashion. Moreover, CBFs can be used with nonlinear (affine in control) system dynamics and they have a crucial forward invariance property which guarantees the satisfaction of safety constraints over all time as long as these constraints are initially satisfied.

An approach combining optimal control solutions with CBFs was recently presented in Xiao, Cassandras, and Belta (2021). In this combined approach (termed OCBF), the solution of an *unconstrained* optimal control problem is first derived and used as a reference control. Then, the resulting control reference trajectory is optimally tracked subject to a set of CBF constraints which ensure the satisfaction of all constraints of the original optimal control problem. Finally, this optimal tracking problem is efficiently solved by discretizing time and solving a simple Quadratic Problem (QP) at each discrete time step over which the control input is held constant (Ames et al., 2017). The use of CBFs in this approach exploits their forward invariance property to guarantee that all constraints they enforce are satisfied at all times if they are initially satisfied. In addition, CBFs are designed to impose *linear* constraints on the control which is what enables the efficient solution of the tracking problem through a sequence of QPs. This approach can also be shown to provide additional flexibility in terms of using nonlinear vehicle dynamics (as long as they are affine in the control), complex objective functions, and tolerating process and measurement noise (Xiao, Cassandras, & Belta, 2021).

However, in solving a sequence of QPs the control update interval in the time discretization process must be sufficiently small in order to always guarantee that every QP is feasible. In practice, such feasibility can often be seen to be violated due to the fact that it is extremely difficult to pick a proper discretization time which can be guaranteed to always work. An additional issue is that small control update intervals result in excessive communication resource consumption and computational costs.

The contribution of this paper is to resolve this problem for the decentralized constrained optimal control of CAVs since it limits the use of CBFs if the feasibility of the resulting controllers cannot always be guaranteed. We accomplish this by replacing the *time-driven* nature of the discretization process to solve a sequence of QPs in the OCBF approach by an *event-driven* mechanism. This allows us to achieve QP feasibility independent of a time step choice.

Such schemes, either *event-triggered* or *self-triggered*, have been considered in recent literature. For example, to avoid unnecessary communication in cooperative adaptive cruise control, Dolk, Ploeg, and Heemels (2017) employ an event-triggered approach while ensuring string stability. In Hu (2021), a self-triggered approach is proposed that ensures the stochastic stability of the vehicular network while efficiently utilizing communication resources. Event-triggered schemes used with CBFs have also been considered in Ong and Cortés (2018) where a Lyapunov function is combined with an event-triggered scheme with the goal of improving stability. In Taylor, Ong, Cortés, and Ames (2020), an input-to-state barrier function is proposed in an event-triggered scheme to impose safety under an input disturbance, while in Yang, Belta, and Tron (2019) a QP-based self-triggered scheme with minimum inter-event time is developed. However, these schemes have only been applied to *single-agent* systems, so that any consideration of communication in multi-agent systems has not been addressed and it is unclear how they can be extended to *multi-agent* environments. In contrast, our setting clearly involves multiple agents (CAVs). Thus, we extend the approach introduced in Xiao, Belta, and Cassandras (2021) for a multi-agent system and adapt it to the specifics of cooperating CAVs in conflict areas by defining events associated with the states of CAVs reaching a certain bound, at which point the next QP instance is triggered. On the other hand, in the self-triggering scheme, we provide a minimum inter-event time guarantee by predicting the first time instant that any of the CBF constraints in the QP problem is violated, hence we can determine the triggering time for the next QP instance. Both methods provide a guarantee for the forward invariance property of CBFs and eliminate infeasible cases due to time-driven inter-sampling effects (additional infeasibilities are still possible due to potentially conflicting constraints within a QP; this separate issue has been addressed in Xiao, Belta, and Cassandras (2022)).

The advantages of these event-driven schemes can be summarized as follows: (*i*) Infeasible QP instances due to inter-sampling effects are eliminated, (*ii*) There is no longer a need to determine a proper time step size required in the time-driven methods, (*iii*) The number of control updates under event-driven schemes is reduced, thereby reducing the overall computational cost, and (iv) Since the number of QPs that need to be solved is reduced, this also reduces the need for unnecessary communication among CAVs. This reduced need for communication, along with the unpredictability of event-triggering relative to a fixed time discretization approach, results in the system being less susceptible to malicious attacks (Ahmad, Sabouni, Xiao, Cassandras, & Li, 2023).

The paper is organized as follows. In Section 2, we provide an overview of the decentralized constrained optimal control for CAVs in any conflict area setting, along with a brief review of CBFs to set the stage for the OCBF approach. We also review the time-driven approach to solve such optimal control problems, motivating the proposed solutions to the problem. In Section 3, event-triggered and self-triggered approaches are separately presented, including the formulation and solution of QPs in both frameworks and their associated communication schemes. In Section 4, simulation results compare time-driven, event-triggered, and self-triggered schemes, in terms of their performance metrics, computational load, and infeasible cases to show how constraint violations are reduced through the proposed approaches.

2. Problem formulation and time-driven control solutions

In this section, we review the setting as for CAVs whose motion is cooperatively controlled at conflict areas of a traffic network. This includes merging roads, signal-free intersections, roundabouts, and highway segments where lane change maneuvers take place. We define a Control Zone (CZ) to be an area within which CAVs can communicate with each other or with a coordinator (e.g., a Road-Side Unit (RSU)) which is responsible for facilitating the exchange of information (but not controlling individual vehicles) within this CZ. As an example, Fig. 1 shows a conflict area due to vehicles merging from two single-lane roads and there is a single Merging Point (MP) which vehicles must cross from either road. The problem formulation here is the same as the one discussed in Xiao, Cassandras, and Belta (2021).

In such a setting, assuming all traffic consists of CAVs, a finite horizon constrained optimal control problem can be formulated

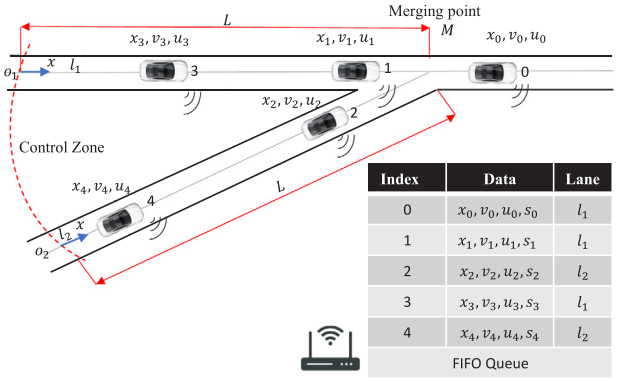


Fig. 1. The merging problem (based on Fig. 1 in Xiao, Cassandras, and Belta (2021)).

aiming to determine trajectories that jointly minimize travel time and energy consumption through the CZ while also ensuring passenger comfort (by minimizing jerk or centrifugal forces) and guaranteeing safety constraints are always satisfied. Let $F(t)$ be the set of indices of all CAVs located in the CZ at time t. A CAV enters the CZ at one of several origins (e.g., O and O' in Fig. 1) and leaves at one of possibly several exit points (e.g., M in Fig. 1). The index 0 is used to denote a CAV that has just left the CZ. Let $N(t)$ be the cardinality of $F(t)$. Thus, if a CAV arrives at time t, it is assigned the index $N(t)+1$. All CAV indices in $F(t)$ decrease by one when a CAV passes over the MP and the vehicle whose index is -1 is dropped.

The vehicle dynamics for each CAV $i \in F(t)$ along the lane to which it belongs in a given CZ are assumed to be of the form

$$\dot{x}_i(t) = v_i(t), \quad \dot{v}_i(t) = u_i(t) \tag{1}$$

where $x_i(t)$ denotes the distance from the origin at which CAV i arrives, $v_i(t)$ denotes the velocity, and $u_i(t)$ denotes the control input (acceleration). There are two objectives for each CAV, as detailed next.

Objective 1 (Minimize travel time): Let t_i^0 and t_i^f denote the time that CAV $i \in F(t)$ arrives at its origin and leaves the CZ at its exit point, respectively. We wish to minimize the travel time $t_i^f - t_i^0$ for CAV i.

Objective 2 (Minimize energy consumption): We also wish to minimize the energy consumption for each CAV i:

$$J_i(u_i(t), t_i^f) = \int_{t_i^0}^{t_i^f} \mathcal{L}_i(|u_i(t)|)dt, \tag{2}$$

where $\mathcal{L}_i(\cdot)$ is a strictly increasing function of its argument.

Constraint 1 (Safety constraints): Let i_p denote the index of the CAV which physically immediately precedes i in the CZ (if one is present). We require that the distance $z_{i,i_p}(t) := x_{i_p}(t) - x_i(t)$ be constrained by:

$$z_{i,i_p}(t) \geq \varphi v_i(t) + \delta, \quad \forall t \in [t_i^0, t_i^f], \tag{3}$$

where φ denotes the reaction time (as a rule, $\varphi = 1.8s$ is used, e.g., Vogel (2003)) and δ is a given minimum safe distance. If we define z_{i,i_p} to be the distance from the center of CAV i to the center of CAV i_p, then δ depends on the length of these two CAVs (generally dependent on i and i_p but taken to be a constant over all CAVs for simplicity).

Constraint 2 (Safe merging): Whenever a CAV crosses a MP, a lateral collision is possible and there must be adequate safe space for the CAV at this MP to avoid such collision, i.e.,

$$z_{i,i_c}(t_i^m) \geq \varphi v_i(t_i^m) + \delta, \tag{4}$$

where i_c is the index of the CAV that may collide with CAV i at merging point $m = \{1, \ldots, n_i\}$ where n_i is the total number of MPs that CAV i passes in the CZ. The determination of CAV i_c depends on the policy adopted for sequencing CAVs through the CZ, such as First-In-First-Out (FIFO) based on the arrival times of CAVs, or any other desired policy. It is worth noting that this constraint only applies at a certain time t_i^m which obviously depends on how the CAVs are controlled. As an example, in Fig. 1 under FIFO, we have $i_c = i - 1$ and $t_i^m = t_i^f$ since the MP defines the exit from the CZ.

Constraint 3 (Vehicle limitations): Finally, there are constraints on the speed and acceleration for each $i \in F(t)$:

$$v_{\min} \leq v_i(t) \leq v_{\max}, \forall t \in [t_i^0, t_i^f] \tag{5}$$

$$u_{\min} \leq u_i(t) \leq u_{\max}, \forall t \in [t_i^0, t_i^f], \tag{6}$$

where $v_{\max} > 0$ and $v_{\min} \geq 0$ denote the maximum and minimum speed allowed in the CZ for CAV i, $u_{\min} < 0$ and $u_{\max} > 0$ denote the minimum and maximum control for CAV i, respectively.

Optimal Control Problem formulation. Our goal is to determine a control law achieving objectives 1–2 subject to constraints 1–3 for each $i \in F(t)$ governed by the dynamics (1). Choosing $\mathcal{L}_i(u_i(t)) = \frac{1}{2}u_i^2(t)$ and normalizing travel time and $\frac{1}{2}u_i^2(t)$, we use the weight $\alpha \in [0, 1]$ to construct a convex combination as follows:

$$\min_{u_i(t), t_i^f} J_i(u_i(t), t_i^f) = \int_{t_i^0}^{t_i^f} \left(\alpha + \frac{(1-\alpha)\frac{1}{2}u_i^2(t)}{\frac{1}{2}\max\{u_{\max}^2, u_{\min}^2\}} \right) dt. \tag{7}$$

Letting $\beta := \frac{\alpha \max\{u_{\max}^2, u_{\min}^2\}}{2(1-\alpha)}$, to obtain a simplified form we multiply (7) by $\frac{\beta}{\alpha}$ which results in:

$$\min_{u_i(t), t_i^f} J_i(u_i(t), t_i^f) := \beta(t_i^f - t_i^0) + \int_{t_i^0}^{t_i^f} \frac{1}{2}u_i^2(t)dt, \tag{8}$$

where $\beta \geq 0$ is an adjustable weight to penalize travel time relative to the energy cost. Note that the solution is *decentralized* in the sense that CAV i requires information only from CAVs i_p and i_c required in (3) and (4).

Problem (8) subject to (1), (3), (4), (5) and (6) can be analytically solved in some cases, e.g., the merging problem in Fig. 1, (Xiao & Cassandras, 2021) and, a signal-free intersection, (Zhang & Cassandras, 2019). However, obtaining solutions for real-time applications becomes prohibitive when an optimal trajectory involves multiple constraints becoming active. This has motivated an approach which combines a solution of the unconstrained problem (8), which can be obtained very fast, with the use of Control Barrier Functions (CBFs) which provide guarantees that (3), (4), (5) and (6) are always satisfied through constraints that are linear in the control, thus rendering solutions to this alternative problem obtainable by solving a sequence of computationally efficient QPs. This approach is termed Optimal Control with Control Barrier Functions (OCBF) (Xiao, Cassandras, & Belta, 2021).

The OCBF approach. The OCBF approach consists of three steps: (*i*) the solution of the *unconstrained* optimal control problem (8) is used as a reference control, (*ii*) the resulting control reference trajectory is optimally tracked subject to the constraint (6), as well as a set of CBF constraints enforcing (3), (4) and (5). (*iii*) This optimal tracking problem is efficiently solved by discretizing time and solving a simple QP at each discrete time step. The significance of CBFs in this approach is twofold: first, their forward invariance property (Xiao, Cassandras, & Belta, 2021) guarantees that all constraints they enforce are satisfied at all times if they are initially satisfied; second, CBFs impose *linear*

constraints on the control which is what enables the efficient solution of the tracking problem through the sequence of QPs in (*iii*) above.

The reference control in step (*i*) above is denoted by $u_i^{\text{ref}}(t)$. The unconstrained solution to (8) is denoted by $u_i^*(t)$, thus we usually set $u_i^{\text{ref}}(t) = u_i^*(t)$. However, $u_i^{\text{ref}}(t)$ may be chosen to be any desired control trajectory and, in general, we use $u_i^{\text{ref}}(t) = h(u_i^*(t), x_i^*(t), \mathbf{x}_i(t))$ where $\mathbf{x}_i(t) \equiv (x_i(t), v_i(t))^T$, $\mathbf{x}_i \in \mathbf{X}$ ($\mathbf{X} \subset \mathbb{R}^2$ is the state space). Thus, in addition to the unconstrained optimal control and position $u_i^*(t), x_i^*(t)$, observations of the actual CAV state $\mathbf{x}_i(t)$ provide direct feedback as well.

To derive the CBFs that ensure the constraints (3), (4), and (5) are always satisfied, we use the vehicle dynamics (1) to define $f(\mathbf{x}_i(t)) = [v_i(t), 0]^T$ and $g(\mathbf{x}_i(t)) = [0, 1]^T$. Each of these constraints can be easily written in the form of $b_q(\mathbf{x}(t)) \geq 0$, $q \in \{1, \ldots, n\}$ where n stands for the number of constraints and $\mathbf{x}(t) = [\mathbf{x}_1(t), \mathbf{x}_2(t), \ldots, \mathbf{x}_{N(t)}(t)]$. The CBF method (details provided in Xiao, Cassandras, and Belta (2021)) maps a constraint $b_q(\mathbf{x}(t)) \geq 0$ onto a new constraint which is linear in the control input $u_i(t)$ and takes the general form

$$L_f b_q(\mathbf{x}(t)) + L_g b_q(\mathbf{x}(t)) u_i(t) + \gamma(b_q(\mathbf{x}(t))) \geq 0, \tag{9}$$

where L_f, L_g denote the Lie derivatives of $b_q(\mathbf{x}(t))$ along f and g, respectively and $\gamma(\cdot)$ stands for any class-$\mathcal{K}$ function (Xiao, Cassandras, & Belta, 2021). It has been established (Xiao, Cassandras, & Belta, 2021) that satisfaction of (9) implies the satisfaction of the original problem constraint $b_q(\mathbf{x}(t)) \geq 0$ because of the forward invariance property. It is worth observing that the newly obtained constraints are *sufficient* conditions for the original problem constraints, therefore, potentially conservative.

We now apply (9) to obtain the CBF constraint associated with the safety constraint (3). By setting

$$\begin{aligned} b_1(\mathbf{x}_i(t), \mathbf{x}_{i_p}(t)) &= z_{i,i_p}(t) - \varphi v_i(t) - \delta \\ &= x_{i_p}(t) - x_i(t) - \varphi v_i(t) - \delta, \end{aligned} \tag{10}$$

and since $b_1(\mathbf{x}_i(t), \mathbf{x}_{i_p}(t))$ is differentiable, the CBF constraint for (3) is

$$\underbrace{v_{i_p}(t) - v_i(t)}_{L_f b_1(\mathbf{x}_i(t), \mathbf{x}_{i_p}(t))} + \underbrace{-\varphi}_{L_g b_1(\mathbf{x}_i(t))} u_i(t) + \underbrace{k_1(z_{i,i_p}(t) - \varphi v_i(t) - \delta)}_{\gamma_1(b_1(\mathbf{x}_i(t), \mathbf{x}_{i_p}(t)))} \geq 0, \tag{11}$$

where the class-$\mathcal{K}$ function $\gamma_1(x) = k_1 x$ is chosen here to be linear.

Deriving the CBF constraint for the safe merging constraint (4) poses a technical challenge due to the fact that it only applies at a certain time t_i^m, whereas a CBF is required to be in a continuously differentiable form. To tackle this problem, we apply a technique used in Xiao, Cassandras, and Belta (2021) to convert (4) to a continuous differentiable form as follows:

$$z_{i,i_c}(t) - \Phi(x_i(t)) v_i(t) - \delta \geq 0, \ \forall t \in [t_i^0, t_i^m], \tag{12}$$

where $\Phi : \mathbb{R} \to \mathbb{R}$ may be any continuously differentiable function as long as it is strictly increasing and satisfies the boundary conditions $\Phi(x_i(t_i^0)) = 0$ and $\Phi(x_i(t_i^m)) = \varphi$. In this case, a linear function can satisfy both conditions: $\Phi(x_i(t)) = \varphi \frac{x_i(t)}{L}$, where L is the length of road traveled by the CAV from its entry to the CZ to the MP of interest in (4). Then by setting

$$\begin{aligned} b_2(\mathbf{x}_i(t), \mathbf{x}_{i_c}(t)) &= z_{i,i_c}(t) - \varphi v_i(t) - \delta \\ &= x_{i_c}(t) - x_i(t) - \Phi(x_i(t)) v_i(t) - \delta, \end{aligned} \tag{13}$$

proceeding as in the derivation of (11), we obtain:

$$\underbrace{v_{i_c}(t) - v_i(t) - \frac{\varphi}{L} v_i^2(t)}_{L_f b_2(\mathbf{x}_i(t), \mathbf{x}_{i_c}(t))} + \underbrace{-\varphi \frac{x_i(t)}{L}}_{L_g b_2(\mathbf{x}_i(t))} u_i(t) +$$

$$\underbrace{k_2(z_{i,i_c}(t) - \varphi \frac{x_i(t)}{L} v_i(t) - \delta)}_{\gamma_2(b_2(\mathbf{x}_i(t), \mathbf{x}_{i_c}(t)))} \geq 0. \tag{14}$$

The speed constraints in (5) are also easily transformed into CBF constraints using (9) by defining

$$b_3(\mathbf{x}_i(t)) = v_{\max} - v_i(t), \tag{15}$$

$$b_4(\mathbf{x}_i(t)) = v_i(t) - v_{\min}. \tag{16}$$

This yields:

$$\underbrace{-1}_{L_g b_3(\mathbf{x}_i(t))} u_i(t) + \underbrace{k_3(v_{\max} - v_i(t))}_{\gamma_3(b_3(\mathbf{x}_i(t)))} \geq 0 \tag{17}$$

$$\underbrace{1}_{L_g b_4(\mathbf{x}_i(t))} u_i(t) + \underbrace{k_4(v_i(t) - v_{\min})}_{\gamma_4(b_4(\mathbf{x}_i(t)))} \geq 0, \tag{18}$$

for the maximum and minimum velocity constraints, respectively.

Inclusion of soft constraints in (8). As a last step in the OCBF approach, we can exploit the versatility of the CBF method to include soft constraints expressed as terminal state costs in (8), e.g., the CAV achieving a desired terminal speed. This is accomplished by using a Control Lyapunov Function (CLF) to track specific state variables in the reference trajectory if desired. A CLF $V(\mathbf{x}_i(t))$ is similar to a CBF (see Xiao, Cassandras, and Belta (2021)). In our problem, letting $V(\mathbf{x}_i(t)) = (v_i(t) - v_i^{\text{ref}}(t))^2$ we can express the CLF constraint associated with tracking the CAV speed to a desired value $v_i^{\text{ref}}(t)$ (if one is provided) as follows:

$$L_f V(\mathbf{x}_i(t)) + L_g V(\mathbf{x}_i(t)) u_i(t) + \epsilon V(\mathbf{x}_i(t)) \leq e_i(t), \tag{19}$$

where $\epsilon > 0$ and $e_i(t)$ makes this a soft constraint.

Now that all the original problem constraints have been transformed into CBF constraints, we can formulate the OCBF problem as follows:

$$\min_{u_i(t), e_i(t)} J_i(u_i(t), e_i(t)) := \int_{t_i^0}^{t_i^f} \left[\frac{1}{2}(u_i(t) - u_i^{\text{ref}}(t))^2 + \lambda e_i^2(t)\right] dt \tag{20}$$

subject to vehicle dynamics (1), the CBF constraints (11), (14), (17), (18), the control constraint (6), and CLF constraint (19). Note that this is a decentralized optimization problem, as it only requires information sharing with a small number of "neighbor" CAVs, i.e. CAV i_p and i_c (if they exist). We denote this set of CAV neighbors by $\mathcal{R}_i(t)$ at time t:

$$\mathcal{R}_i(t) = \{i_p(t), i_c(t)\}. \tag{21}$$

Note that $\mathcal{R}_i(t)$ in general can change over time, i.e., $i_c(t)$ changes when dynamic "resequencing" (discussed in Xiao and Cassandras (2020)) is carried out and $i_p(t)$ changes in the case of lane changing maneuvers. It is worth mentioning that in the single lane merging example in Fig. 1 i_p cannot change.

A common way to solve this dynamic optimization problem is to discretize $[t_i^0, t_i^f]$ into intervals $[t_i^0, t_i^0 + \Delta), \ldots, [t_i^0 + k\Delta, t_i^0 + (k+1)\Delta), \ldots$ with equal length Δ and solving (20) over each time interval. The decision variables $u_{i,k} = u_i(t_{i,k})$ and $e_{i,k} = e_i(t_{i,k})$ are assumed to be constant on each interval and can be easily calculated at time $t_{i,k} = t_i^0 + k\Delta$ through solving a QP at each time step:

$$\min_{u_{i,k}, e_{i,k}} \left[\frac{1}{2}(u_{i,k} - u_i^{\text{ref}}(t_{i,k}))^2 + \lambda e_{i,k}^2\right], \tag{22}$$

subject to the CBF constraints (11), (14), (17), (18), and control input bounds (6) and CLF constraint (19) where all constraints are linear in the decision variables. We refer to this as the *time-driven* approach, which is fast and can be readily used in real time.

The main problem with this approach is that a QP may become infeasible at any time instant because the decision variable $u_{i,k}$ is held constant over a given time period Δ. Since this is externally defined, there is no guarantee that it is small enough to ensure the forward invariance property of a CBF, thereby also failing to ensure the satisfaction of the safety constraints. In other words, in this time-driven approach, there is a critical (and often restrictive) assumption that the control update rate is high enough to avoid such a problem. There are several additional issues worth mentioning: (*i*) imposing a high update rate makes the solution of multiple QPs inefficient since it increases the computational burden, (*ii*) using a common update rate across all CAVs renders their synchronization difficult, and (*iii*) the predictability of a time-driven communication mechanism across CAVs makes the whole system susceptible to malicious attacks. As we will show next, the two event-driven solutions proposed in this paper alleviate these problems by eliminating the need to select a time step Δ.

3. Event-driven solutions

There are several possible event-driven mechanisms one can adopt to invoke the solution of the QPs in (22) subject to the CBF constraints (11), (14), (17), (18) along with control input bounds (6). One approach is to adopt an *event-triggering* scheme such that we only need to solve a QP (with its associated CBF constraints) when one of two possible events (as defined next) is detected. We will show that this provides a guarantee for the satisfaction of the safety constraints which cannot be offered by the time-driven approach described earlier. The key idea is to ensure that the safety constraints are satisfied while the state remains within some bounds and define events which coincide with the state reaching these bounds, at which point the next instance of the QP in (22) is triggered. Another idea is to create a *self-triggering* framework with a minimum inter-event time guarantee by predicting at $t_{i,k}$ the first time instant that any of the CBF constraints in the QP problem (22) is subsequently violated. We then select that as the next time instant $t_{i,k+1}$ when CAV i communicates with the coordinator and updates the control. A comparison of the two mechanisms is given in Section 3.4.

3.1. Event-triggered control

Let $t_{i,k}$, $k = 1, 2, \ldots$, be the time instants when the QP in (22) is solved by CAV i. Our goal is to guarantee that the state trajectory does not violate any safety constraints within any time interval $[t_{i,k}, t_{i,k+1})$ where $t_{i,k+1}$ is the next time instant when the QP is solved. Define a subset of the state space of CAV i at time $t_{i,k}$ such that:

$$\mathbf{x}_i(t_{i,k}) - \mathbf{s}_i \leq \mathbf{x}_i(t) \leq \mathbf{x}_i(t_{i,k}) + \mathbf{s}_i, \tag{23}$$

where $\mathbf{s}_i = \begin{bmatrix} s_{i_x} & s_{i_v} \end{bmatrix}^T \in \mathbb{R}^2_{>0}$ is a parameter vector whose choice will be discussed later. Intuitively, this choice reflects a trade-off between *computational efficiency* (when the $\mathbf{s}_i$ values are large and there are fewer instances of QPs to be solved) and *conservativeness* (when the values are small). We denote the set of states of CAV i that satisfy (23) at time $t_{i,k}$ by

$$S_i(t_{i,k}) = \left\{ \mathbf{y}_i \in \mathbf{X} : \ \mathbf{x}_i(t_{i,k}) - \mathbf{s}_i \leq \mathbf{y}_i \leq \mathbf{x}_i(t_{i,k}) + \mathbf{s}_i \right\}. \tag{24}$$

In addition, let $C_{i,1}$ be the feasible set of our original constraints (3), (4) and (5) defined as

$$C_{i,1} := \left\{ \mathbf{x}_i \in \mathbf{X} : \ b_q(\mathbf{x}_i) \geq 0, \ q \in \{1, 2, 3, 4\} \right\}. \tag{25}$$

Next, we seek a bound and a control law that satisfies the safety constraints within this bound. This can be accomplished by considering the minimum value of each component in (9) for every $q \in \{1, 2, 3, 4\}$ as shown next.

Modified CBF constraints: Let us start with the first of the three terms in (9), $L_f b_q(\mathbf{x}(t))$. Observing that not all state variables are generally involved in a constraint $b_q(\mathbf{x}(t)) \geq 0$, we can rewrite this term as $L_f b_q(\mathbf{y}_i(t), \mathbf{y}_r(t))$ with $\mathbf{y}_i(t)$ as in (24) and where r stands for "relevant" CAVs affecting the specific constraint of i, i.e., $r \in \mathcal{R}_i(t)$ in (21). Let $b^{\min}_{q,f_i}(t_{i,k})$ be the minimum possible value of the term $L_f b_q(\mathbf{y}_i(t), \mathbf{y}_r(t))$ over the time interval $[t_{i,k}, t_{i,k+1})$ for each $q = \{1, 2, 3, 4\}$ over the set $\bar{S}_i(t_{i,k}) \cap \bar{S}_r(t_{i,k})$:

$$b^{\min}_{q,f_i}(t_{i,k}) = \min_{\substack{\mathbf{y}_i \in \bar{S}_i(t_{i,k}) \\ \mathbf{y}_r \in \bar{S}_r(t_{i,k})}} L_f b_q(\mathbf{y}_i(t), \mathbf{y}_r(t)), \tag{26}$$

where $\bar{S}_i(t_{i,k})$ is defined as follows:

$$\bar{S}_i(t_{i,k}) := \{\mathbf{y}_i \in C_{i,1} \cap S_i(t_{i,k})\} \tag{27}$$

Similarly, we can define the minimum value of the third term in (9):

$$b^{\min}_{\gamma_q}(t_{i,k}) = \min_{\substack{\mathbf{y}_i \in \bar{S}_i(t_{i,k}) \\ \mathbf{y}_r \in \bar{S}_r(t_{i,k})}} \gamma_q(\mathbf{y}_i(t), \mathbf{y}_r(t)). \tag{28}$$

For the second term in (9), note that $L_g b_q(\mathbf{x}_i)$ is a constant for $q = \{1, 3, 4\}$, as seen in (11), (17) and (18), therefore there is no need for any minimization. However, $L_g b_2(\mathbf{x}_i) = -\varphi \frac{x_i(t)}{L}$ in (14) is state-dependent and needs to be considered for the minimization. Since $x_i(t) \geq 0$, note that $L_g b_2(\mathbf{x}_i)$ is always negative, therefore, we can determine the limit value $b^{\min}_{2,g_i}(t_{i,k}) \in \mathbb{R}$, as follows:

$$b^{\min}_{2,g_i}(t_{i,k}) = \begin{cases} \min\limits_{\substack{\mathbf{y}_i \in \bar{S}_i(t_{i,k}) \\ \mathbf{y}_r \in \bar{S}_r(t_{i,k})}} L_g b_2(\mathbf{x}_i(t)), & \text{if } u_{i,k} \geq 0 \\ \max\limits_{\substack{\mathbf{y}_i \in \bar{S}_i(t_{i,k}) \\ \mathbf{y}_r \in \bar{S}_r(t_{i,k})}} L_g b_2(\mathbf{x}_i(t)), & \text{otherwise,} \end{cases} \tag{29}$$

where the sign of $u_{i,k}$, $i \in F(t_{i,k})$ can be determined by simply solving the CBF-based QP (22) at time $t_{i,k}$.

Thus, the condition that can guarantee the satisfaction of (11), (14) and (17), (18) in the time interval $\left[t_{i,k}, t_{i,k+1}\right)$ is given by

$$b^{\min}_{q,f_i}(t_{i,k}) + b^{\min}_{q,g_i}(t_{i,k}) u_{i,k} + b^{\min}_{\gamma_q}(t_{i,k}) \geq 0, \tag{30}$$

for $q = \{1, 2, 3, 4\}$. In order to apply this condition to the QP (22), we just replace (9) by (30) as follows:

$$\min_{u_{i,k}, e_{i,k}} \left[\frac{1}{2}(u_{i,k} - u^{\text{ref}}_i(t_{i,k}))^2 + \lambda e^2_{i,k} \right]$$
$$\text{s.t.} (19), (30), (6) \tag{31}$$

Event-triggered control execution: It is important to note that each instance of the QP (31) is now triggered by one of the following two events where $k = 1, 2, \ldots$ is a local event (rather than time step) counter:

- **Event 1:** the state of CAV i reaches the boundary of $S_i(t_{i,k-1})$.
- **Event 2:** the state of CAV $r \in \mathcal{R}_i(t_{i,k-1})$ reaches the boundary of $S_r(t_{i,k-1})$, if $\mathcal{R}_i(t_{i,k-1})$ is nonempty. In this case either $r = i_p$ or $r = i_c$ (e.g., in the merging problem $i_c = i - 1 \neq i_p$ if such a CAV exists). Thus, Event 2 is further identified by the CAV which triggers it and denoted accordingly by **Event 2**(r), $r \in \mathcal{R}_i(t_{i,k-1})$.

As a result, $t_{i,k}$, $k = 1, 2, \ldots$ is unknown in advance but can be determined by CAV i through:

$$t_{i,k} = \min \left\{ t > t_{i,k-1} : |\mathbf{x}_i(t) - \mathbf{x}_i(t_{i,k-1})| = \mathbf{s}_i \right. \tag{32}$$

$$\text{or } |\mathbf{x}_{i_p}(t) - \mathbf{x}_{i_p}(t_{i,k-1})| = \mathbf{s}_{i_p}$$
$$\text{or } |\mathbf{x}_{i_c}(t) - \mathbf{x}_{i_c}(t_{i,k-1})| = \mathbf{s}_{i_c} \Big\},$$

where $t_{i,0} = t_i^0$. Note that k is a *local* event counter for each i so, strictly speaking, we should use k_i. Instead, the index k can be dropped and we can write $\mathbf{x}_i(t_{i,\text{last}})$ rather than $\mathbf{x}_i(t_{i,k-1})$. However, when there is no ambiguity, we will simply write $\mathbf{x}_i(t_i)$ to indicate that t_i is the "last event" occurring at i.

The definition above is based on events which directly affect CAV i (leading to i solving a QP) whether they are triggered by i or $r \neq i$. Alternatively, we may think of any CAV i as generating Event 1 leading to a new QP solution by CAV i itself and Event 2(i) which affects some $j \in \{l | i \in \mathcal{R}_l(t)\}$, i.e., i is relevant to some $j \neq i$. In this case, a violation of the bound of $S_i(t_{i,k-1})$ or $S_i(t_{j,k-1})$ by the evolving state of CAV i triggers events relevant to CAV i or j, respectively.

Events 1,2(r) can be detected through the dynamics in (1) or from on-board state measurements, if available, along with state information from relevant other CAVs (e.g., CAVs i_p and i_c in Fig. 1) through the coordinator. Finally, note that because of the Lipschitz continuity of the dynamics in (1) and the fact that the control is constant within an inter-event interval, Zeno behavior does not occur in this framework.

The following theorem formalizes our analysis by showing that if new constraints of the general form (30) hold, then our original CBF constraints (11), (14) and (17), (18) hold.

Theorem 1. *Given a CBF $b_q(\mathbf{x}(\mathbf{t}))$ with relative degree one, let $t_{i,k}$, $k = 1, 2, \ldots$ be determined by (32) with $t_{i,0} = t_i^0$ and $b^{\min}_{q,f_i}(t_{i,k})$, $b^{\min}_{\gamma_q}(t_{i,k})$, $b^{\min}_{q,g_i}(t_{i,k})$ for $q = \{1, 2, 3, 4\}$ obtained through (26), (28), and (29). Then, any control input $u_{i,k}$ that satisfies (30) for all $q \in \{1, 2, 3, 4\}$ within the time interval $[t_{i,k}, t_{i,k+1})$ renders the set $C_{i,1}$ forward invariant for the dynamic system defined in (1).*

The proof follows along similar lines as Theorem 2 in Xiao, Belta, and Cassandras (2021). By (24), we can write:

$$\mathbf{y}_i(t) \in S_i(t_{i,k}),\ \mathbf{y}_r(t) \in S_r(t_{i,k}),\ \mathbf{y}_i(t) \in C_{i,1} \tag{33}$$

for all $t \in [t_{i,k}, t_{i,k+1})$, $k = 1, 2, \ldots$.

$$L_f b_q(\mathbf{x}_i(t)) \geq b^{\min}_{q,f_i}(t_k), \tag{34}$$

$$\gamma_q(\mathbf{x}_i(t)) \geq b^{\min}_{\gamma_q}(t_k), \tag{35}$$

$$L_g b_q(\mathbf{x}_i(t)) u_i(t_k) \geq b^{\min}_{q,g_i}(t_k) u_i(t_k), \tag{36}$$

for $q \in \{1, 2, 3, 4\}$, by adding these inequalities which have the same direction it follows that

$$\begin{aligned} &L_f b_q(\mathbf{x}_i(t)) + L_g b_q(\mathbf{x}_i(t)) u_i(t_k) + \gamma_q(\mathbf{x}_i(t)) \\ &\geq b^{\min}_{q,f_i}(t_k) + b^{\min}_{q,g_i}(t_k) u_i(t_k) + b^{\min}_{\gamma_q}(t_k) \geq 0. \end{aligned} \tag{37}$$

i.e., (9) is satisfied. By Theorem 1 of Xiao, Cassandras, and Belta (2021) applied to (9), if $x_i(0) \in C_{i,1}$, then any Lipschitz continuous controller $u_i(t)$ that satisfies (9) $\forall t \geq 0$ renders $C_{i,1}$ forward invariant for system (1). Therefore, $C_{i,1}$ is forward invariant for the dynamic system defined in (1).

Remark 1. Expressing (30) in terms of the minimum value of each component separately may become overly conservative if each minimum value corresponds to different points in the decision variable space. Therefore, an alternative approach is to calculate the minimum value of the whole term.

Selection of parameters $\mathbf{s}_i$. The importance of properly selecting the parameters $\mathbf{s}_i$ is twofold. First, it is necessary to choose them such that all events are observed, i.e., given the sensing capabilities and limitations of a CAV i, the value of $\mathbf{s}_i$ must be large enough to ensure that no events will go undetected. In particular, the variation of the states of CAV i within the sensor sampling time must not be greater than bounds $\mathbf{s}_i$. Therefore, letting T_s be a given sensor sampling time, the maximum state (position and speed) variation during this sampling time must satisfy:

$$x_i(t + T_s) - x_i(t) \leq v_{\max} T_s \tag{38}$$

$$v_i(t + T_s) - v_i(t) \leq \max(u_{\max} T_s, |u_{\min}| T_s) \tag{39}$$

where $v_{\max}$, $u_{\max}$, and $u_{\min}$ are given CAV i specifications. Therefore, we need to pick lower bounds given by the maximum state variations in (38) and (39) as follows:

$$\mathbf{s}_i = \begin{bmatrix} s_{ix} \\ s_{iv} \end{bmatrix} \geq \begin{bmatrix} v_{\max} T_s \\ \max(u_{\max} T_s, |u_{\min}| T_s) \end{bmatrix}. \tag{40}$$

Note that these lower bounds hold for every CAV $i \in F(t)$. Moreover the right hand side only depends on the CAV specification (we assume all CAVs are the same), therefore it follows that the same lower bounds can also be used for $\mathbf{s}_{i_c}$ and $\mathbf{s}_{i_p}$.

Second, the choice of $\mathbf{s}_i$ captures the trade-off between computational cost and conservativeness: the larger the value of each component of $\mathbf{s}_i$ is, the smaller the number of events triggering instances of the QPs becomes, thus reducing the total computational cost. At the same time, the control law must satisfy the safety constraints over a longer time interval as we take the minimum values in (26)–(29), rendering the approach more conservative.

Remark 2. Network delays can also be dealt with through a proper choice of $\boldsymbol{s}_r$, where $r \in \mathcal{R}_i(t)$ as defined in (21) (i.e., assuming there is no delay in obtaining CAV i state information, $\boldsymbol{s}_i$ will remain the same). Given a bounded delay in the network, the state's bound parameters can be selected in such a way as to ensure safety. Let τ_D denote the upper bound of the network delay, we can write: $\boldsymbol{s}_r^{\text{new}}(t_{i,k}) = \boldsymbol{s}_r + \boldsymbol{s}_r^D(t_{i,k})$ where

$$\mathbf{s}_r^D(t_{i,k}) = \begin{bmatrix} s^D_{rx}(t_{i,k}) \\ s^D_{rv}(t_{i,k}) \end{bmatrix} = \begin{bmatrix} \max\left(\frac{1}{2} u_{\min} \tau_D^2 + v_r(t_{i,k}) \tau_D, 0\right) \\ |u_{\min}| \tau_D \end{bmatrix}, \tag{41}$$

can be defined given $u_{\min}$. Then the set $S_r(t_{i,k})$ can be modified accordingly as follows:

$$\begin{aligned} &S_r(t_{i,k}) = \\ &\left\{ \boldsymbol{y}_r \in \mathbf{X} :\ \boldsymbol{x}_r(t_{i,k}) - \boldsymbol{s}_r^{\text{new}}(t_{i,k}) \leq \boldsymbol{y}_r \leq \boldsymbol{x}_r(t_{i,k}) + \boldsymbol{s}_r^{\text{new}}(t_{i,k}) \right\}. \end{aligned} \tag{42}$$

Even though we want to calculate the maximum deviation in the states given an upper bound on the delay, we only use $u_{\min}$ instead of $\max(|u_{\min}|, u_{\max})$. Due to the fact that delays become critical to CAV i in terms of safety only when other relevant vehicles r start slowing down, only the maximum rate of deceleration is used in (41). Last but not least, this is clearly not the only way to deal with network delays, as there are other methods involving, for example, carefully designed communication protocols.

3.2. Self-triggered control

As an alternative to event-triggered control, a self-triggered asynchronous control scheme can be used where each CAV i communicates with the coordinator at specified time instants $\{t_{i,k}\}$, $k \in \mathbb{Z}^+$. At each such instant $t_{i,k}$, CAV i uploads its own state information $\mathbf{x}_i(t_{i,k})$, the calculated control input $u_i(t_{i,k})$ that is going to be applied over the time interval $[t_{i,k}, t_{i,k+1})$, and the next time when CAV i will communicate with the coordinator and resolve its QP, denoted as $t_{i,\text{next}}$. The data stored at the coordinator

Table 1
Data stored on the coordinator for self-triggered control.

$t_{i,\text{last}}$	Last time CAV i communicated.
$t_{i,\text{next}}$	Next time CAV i will communicate.
$x_i(t_{i,\text{last}})$	Last updated position of CAV i.
$v_i(t_{i,\text{last}})$	Last updated velocity of CAV i.
$u_i(t_{i,\text{last}})$	Last control input of CAV i.

for all vehicles are shown in Table 1. We denote the most recent stored information of the ith CAV at the coordinator as $\mathcal{I}_i = [t_{i,\text{last}}, t_{i,\text{next}}, x_i(t_{i,\text{last}}), v_i(t_{i,\text{last}}), u_i(t_{i,\text{last}})]$.

The goal is to develop a self-triggered asynchronous algorithm to determine the sequence of time instants $t_{i,k}$ and the control input $u_i(t), t \in [t_{i,k}, t_{i,k+1})$ for each CAV to solve the problem formed in (20). Providing a lower bound for the inter-event time interval is an imperative feature in a self-triggered scheme. It is worth mentioning that since Zeno behavior never occurs under Lipschitz continuity, such a guarantee is not necessary for the event-triggered control algorithm. To provide such a guarantee for the generated time instants $t_{i,k}$, there should exist some $T_d > 0$ such that $|t_{i,k+1} - t_{i,k}| \geq T_d$. This is a design parameter which depends on the sensor sampling rate, as well as the clock of the on-board embedded system on each CAV. For the same reason, the time-instants $t_{i,k}$ are calculated such that $(t_{i,k} \bmod T_d) = 0$ where mod denotes the modulo operator. In contrast to the time-driven scheme with a fixed sampling time Δ, each CAV $i \in F(t_{i,k})$ calculates the time instant $t_{i,k}$ in which the QP problem must be solved in a self-triggered fashion. As in the event-triggered scheme, at each time instant $t_{i,k}$, CAV i solves its QP problem to obtain $u_i(t_{i,k})$. However, unlike the event-triggered scheme, CAV i also calculates the next time instant $t_{i,k+1}$ at which it should resolve the QP problem. Note that similar to the time-driven scheme, the newly obtained control input, $u_i(t_{i,k})$ is held constant over the time interval $[t_{i,k}, t_{i,k+1})$ for CAV i. We address two problems in the following. First, it will be shown how a lower bound T_d on the inter-event time interval can be ensured. Second, we will show how each CAV $i \in F(t_{i,k})$ specifies the time instants $t_{i,k}$.

3.2.1. Minimum inter-event time, T_d

In this subsection, it is shown how the CBF constraints (11), (14), (17), and (18) for the CAV i should be modified to ensure a minimum inter-event time T_d. This is achieved by adding extra positive terms to the right hand side of these constraints.

Modified maximum speed CBF constraints: first, consider the maximum speed CBF constraint (17) to be satisfied when solving the QP problem at $t_{i,k}$ with feasible solution $u_i(t_{i,k})$. Thus, we have:

$$\mathcal{C}_{i,1}(t_{i,k}, u_i(t_{i,k})) := -u_i(t_{i,k}) + k_3 b_3(\mathbf{x}_i(t_{i,k})) \geq 0. \tag{43}$$

However, the CBF constraint should be satisfied over the entire time interval $[t_{i,k}, t_{i,k} + T_d]$ to ensure the minimum inter-event time. Therefore, for all $t \in [t_{i,k}, t_{i,k} + T_d]$:

$$\mathcal{C}_{i,1}(t, u_i(t_{i,k})) = -u_i(t_{i,k}) + k_3 b_3(\mathbf{x}_i(t)) \geq 0. \tag{44}$$

By defining $\tau = t - t_{i,k}$ as the elapsed time after $t_{i,k}$, and recalling that the acceleration is kept constant over the inter-event time, we can derive an expression for the velocity $v_i(t)$ as follows:

$$v_i(\tau) = v_i(t_{i,k}) + u_i(t_{i,k})\tau, \quad \tau \in [0, T_d]. \tag{45}$$

Now by using (17), (43), and (45) we can rewrite (44) as follows:

$$\mathcal{C}_{i,1}(t, u_i(t_{i,k})) = \mathcal{C}_{i,1}(t_{i,k}, u_i(t_{i,k})) - k_3 u_i(t_{i,k})\tau \quad \tau \in [0, T_d], \tag{46}$$

In what follows, we show that if $\mathcal{C}_{i,1}(t_{i,k}, u_i(t_{i,k})) \geq \sigma_{i,1}(T_d)$ holds, then:

$$\mathcal{C}_{i,1}(t, u_i(t_{i,k})) \geq 0, \quad \forall t \in [t_{i,k}, t_{i,k} + T_d], \tag{47}$$

where $\sigma_{i,1}(T_d) := k_3 u_M T_d$ and $u_M = \max(|u_{\min}|, u_{\max}) > 0$. To prove (47), we can rewrite $\mathcal{C}_{i,1}(t_{i,k}, u_i(t_{i,k})) \geq \sigma_{i,1}(T_d)$ as follows:

$$\mathcal{C}_{i,1}(t_{i,k}, u_i(t_{i,k})) - \mathcal{C}_{i,1}(t, u_i(t_{i,k})) + \mathcal{C}_{i,1}(t, u_i(t_{i,k})) \geq \sigma_{i,1}(T_d) \tag{48}$$

By combining (48) with (46), for all $t \in [t_{i,k}, t_{i,k} + T_d]$ and $\tau \in [0, T_d]$ we have:

$$\mathcal{C}_{i,1}(t, u_i(t_{i,k})) \geq \sigma_{i,1}(T_d) - k_3 u_i(t_{i,k})\tau \geq 0, \tag{49}$$

where the non-negativity of the inequality follows from the definition of $\sigma_{i,1}(t_{i,k}, T_d) = k_3 u_M T_d$, i.e., $k_3 u_M T_d - k_3 u_i(t_{i,k})\tau \geq 0$ for $\tau \in [0, T_d]$. Hence, in order to ensure the minimum inter-event interval T_d, the CBF constraint (17) should be modified to:

$$\mathcal{C}_{i,1}(t, u_i(t)) \geq \sigma_{i,1}(T_d). \tag{50}$$

Modified minimum speed CBF constraints: Following a similar derivation of the modified CBF constraint for the minimum speed (17), it follows that (18) should be modified to:

$$\mathcal{C}_{i,2}(t, u_i(t)) \geq \sigma_{i,2}(T_d), \tag{51}$$

where

$$\mathcal{C}_{i,2}(t, u_i(t)) := u_i(t) + k_4 b_4(\mathbf{x}_i(t))$$
$$\sigma_{i,2}(T_d) := k_4 u_M T_d. \tag{52}$$

Modified safety CBF constraint: let us consider the safety CBF constraint (11) to be satisfied when solving the QP problem at $t_{i,k}$ with a feasible solution $u_i(t_{i,k})$. It follows that

$$\mathcal{C}_{i,3}(t_{i,k}, u_i(t_{i,k})) := v_{i_p}(t_{i,k}) - v_i(t_{i,k}) - \varphi u_i(t_{i,k}) + k_1 b_1(\mathbf{x}_i(t_{i,k}), \mathbf{x}_{i_p}(t_{i,k})) \geq 0. \tag{53}$$

Once again, we need to ensure that the CBF constraint is satisfied over the entire time interval $[t_{i,k}, t_{i,k} + T_d]$ as follows:

$$\mathcal{C}_{i,3}(t, u_i(t_{i,k})) = v_{i_p}(t) - v_i(t) - \varphi u_i(t_{i,k}) + k_1 b_1(\mathbf{x}_i(t), \mathbf{x}_{i_p}(t)) \geq 0, \quad t \in [t_{i,k}, t_{i,k} + T_d]. \tag{54}$$

For ease of notation, we use the following definitions:

$$\Delta v_{i,i_p}(t_{i,k}) = v_{i_p}(t_{i,k}) - v_i(t_{i,k}), \tag{55}$$

$$\Delta u_{i,i_p}(t_{i,k}) = u_{i_p}(t_{i,k}) - u_i(t_{i,k}). \tag{56}$$

Similar to the procedure of deriving the lower bound for constraints (17) and (18), by using (11), (55), (56), and

$$x_i(\tau) = x_i(t_{i,k}) + v_i(t_{i,k})\tau + 0.5u_i(t_{i,k})\tau^2, \tag{57}$$
$$x_{i_p}(\tau) = x_{i_p}(t_{i,k}) + v_{i_p}(t_{i,k})\tau + 0.5u_{i_p}(t_{i,k})\tau^2, \tag{58}$$

we rewrite (54) as follows:

$$\mathcal{C}_{i,3}(t, u_i(t_{i,k})) = \mathcal{C}_{i,3}(t_{i,k}, u_i(t_{i,k})) + \Delta u_{i,i_p}(t_{i,k})\tau + k_1\big(0.5\Delta u_{i,i_p}(t_{i,k})\tau^2 + \Delta v_{i,i_p}(t_{i,k})\tau - \varphi u_i(t_{i,k})\tau\big) \geq 0, \quad \tau \in [0, T_d]. \tag{59}$$

To further ease up the notation, we define

$$\mathcal{M}_{i,3}(t, t_{i,k}, u_i(t_{i,k})) := \mathcal{C}_{i,3}(t_{i,k}, u_i(t_{i,k})) - \mathcal{C}_{i,3}(t, u_i(t_{i,k})), \tag{60}$$

which will be used later on. Similarly, in the following we intend to show that if $\mathcal{C}_{i,3}(t_{i,k}, u_i(t_{i,k})) \geq \sigma_{i,3}(t_{i,k}, T_d)$ holds, it follows:

$$\mathcal{C}_{i,3}(t, u_i(t_{i,k})) \geq 0, \quad t \in [t_{i,k}, t_{i,k} + T_d], \tag{61}$$

where

$$\sigma_{i,3}(t_{i,k}, T_d) := |u_{i_p}(t_{i,k})| + k_1\big(0.5T_d^2(|u_{i_p}(t_{i,k})| + u_M) + (|\Delta v_{i,i_p}(t_{i,k})| + (1+\varphi)u_M)T_d\big). \quad (62)$$

To demonstrate (61), we follow the same procedure as before by starting with

$$\mathcal{C}_{i,3}(t_{i,k}, u_i(t_{i,k})) \geq \sigma_{i,3}(t_{i,k}, T_d), \quad (63)$$

and then rewrite (63) in the following form:

$$\mathcal{C}_{i,3}(t_{i,k}, u_i(t_{i,k})) - \mathcal{C}_{i,3}(t, u_i(t_{i,k})) + \mathcal{C}_{i,3}(t, u_i(t_{i,k})) \geq \sigma_{i,3}(t_{i,k}, T_d) \quad (64)$$

Then, combining (60) and (64), for $t \in [t_{i,k}, t_{i,k}+T_d]$ follows that:

$$\mathcal{C}_{i,3}(t, u_i(t_{i,k})) \geq \sigma_{i,3}(t_{i,k}, T_d) - \mathcal{M}_{i,3}(t, t_{i,k}, u_i(t_{i,k})) \quad (65)$$

where $\sigma_{i,3}(t_{i,k}, T_d)$, i.e. the upper bound of $\mathcal{M}_{i,3}(t, t_{i,k}, u_i(t_{i,k}))$, is chosen such that the left hand side of the inequality is always positive:

$$\sigma_{i,3}(t_{i,k}, T_d) - \mathcal{M}_{i,3}(t, t_{i,k}, u_i(t_{i,k})) \geq 0, \quad (66)$$

hence, by modifying the CBF constraint (11) to:

$$\mathcal{C}_{i,3}(t, u_i(t)) \geq \sigma_{i,3}(t, T_d), \quad (67)$$

one can enforce (54).

Modified safe merging CBF constraint: Following a similar approach, to provide a minimum inter-event time T_d, the CBF constraint (14) should be modified to,

$$\mathcal{C}_{i,4}(t, u_i(t)) \geq \sigma_{i,4}(t, T_d), \quad (68)$$

where $\mathcal{C}_{i,4}(t, u_i(t)) = v_{i_c}(t) - v_i(t) - \frac{\varphi}{L}v_i^2(t) - \varphi\frac{x_i(t)}{L}u_i(t) + k_2(b_2(\mathbf{x}_i(t), \mathbf{x}_{i_c}(t))$,

$$\sigma_{i,4}(t, T_d) := 0.5\frac{\varphi}{L}u_M^2T_d^3 + |v_{i_c}(t)| + |v_i(t)| + \frac{\varphi}{L}v_i^2(t)\Big)\, k_4T_d + k_4\Big(\frac{3\varphi}{2L}(u_M^2 + |v_i(t)|u_M) + 0.5(|u_{i_c}(t)| + u_M)\Big)T_d^2 + \Big(|u_{i_c}(t)| + (\frac{3\varphi}{L}|v_i(t)| + \frac{\varphi}{L}|x_i(t)| + 1)u_M. \quad (69)$$

Finally, since the CLF constraint (19) is added optionally for an optimal trajectory, it can be relaxed in the presence of safety constraints and there is generally no need to ensure that it is satisfied during the whole time-interval $t \in [t_{i,k}, t_{i,k}+T_d]$ with the same relaxation variable value $e_i(t_{i,k})$. Therefore, there is no need to modify it as was necessary for the CBF constraints. In conclusion, to ensure the minimum inter-event time T_d, at each time instant $t_{i,k}$, CAV i needs to solve the following QP:

$$\min_{u_{i,k}, e_{i,k}} \ \frac{1}{2}(u_{i,k} - u_i^{\text{ref}}(t_{i,k}))^2 + \lambda e_{i,k}^2 \quad (70)$$

subject to the modified CBF constraints (50), (51), (67), and (68), the control input bounds (6) and the CLF constraint (19). In the next subsection, it will be shown how the time-instant $t_{i,k}$ should be obtained for CAV i.

3.2.2. Self-triggered time instant calculation

The key idea in the self-triggered framework is to predict the first time instant that any of the CBF constraints (11), (14), (17) or (18), is violated and select that as the next time instant $t_{i,k+1}$. CAV i then communicates with the coordinator and requests the necessary information to solve its next QP and obtain a new control input $u_i(t_{i,k+1})$ and update its stored data in the coordinator table. Note that it is not required to consider the modified CBF constraints (50), (51), (67), and (68) here, since these are obtained purely for ensuring the minimum inter-event time T_d, while the original CBF constraints (11), (14), (17), and (18) are sufficient for satisfying constraints 1, 2, and 3 (state limitation constraint) in problem (8).

For the speed constraint (17), it is clear that if $u_i(t_{i,k}) \leq 0$ (decelerating), then this constraint always holds, hence there is no need to check it. However, for $u_i(t_{i,k}) > 0$ (accelerating), the constraint (17) can be violated. To calculate the time instant, $t_{i,k}^1$, when this occurs we need to solve the following equation:

$$-u_i(t_{i,k}) + k_3(v_{\max} - v_i(t)) = 0, \quad t > t_{i,k}. \quad (71)$$

Recalling that the acceleration is held constant in the inter-event time, (71) can be rewritten as

$$-u_i(t_{i,k}) + k_3(v_{\max} - v_i(t_{i,k}) - u_i(t_{i,k})(t - t_{i,k})) = 0 \quad (72)$$

and its solution yields:

$$t_{i,k}^1 = t_{i,k} + \frac{-u_i(t_{i,k}) + k_3v_{\max} - k_3v_i(t_{i,k})}{k_3u_i(t_{i,k})}.$$

Observe that at $t_{i,k}$, the QP in (70) is solved, therefore the constraint (50) is satisfied at $t = t_{i,k}$ and we have $-u_i(t_{i,k}) + k_3(v_{\max} - v_i(t_{i,k})) \geq \sigma_{i,1}(T_d) > 0$. It follows that $t_{i,k}^1 \geq t_{i,k} + T_d$.

For the second speed constraint (18), it is clear that if $u_i(t_{i,k}) \geq 0$ (accelerating), then this constraint is satisfied, hence there is no need to check it. However, for $u_i(t_{i,k}) < 0$ (decelerating), the constraint (18) can be violated. Similar to the previous case, we can solve the following equation for t to obtain $t_{i,k}^2$ as the first time instant that constraint (18) is violated:

$$u_i(t_{i,k}) + k_4(v_i(t_{i,k}) + u_i(t_{i,k})(t - t_{i,k}) - v_{\min}) = 0 \quad t > t_{i,k}. \quad (73)$$

Solving (73) leads to

$$t_{i,k}^2 = t_{i,k} + \frac{-u_i(t_{i,k}) + k_4v_{\min} - k_4v_i(t_{i,k})}{k_4u_i(t_{i,k})},$$

and it can be shown, similar to the previous case, that $t_{i,k}^2 \geq t_{i,k} + T_d$.

For the rear-end safety constraint (11), we need to find the first time instant $t > t_{i,k}$ such that $\mathcal{C}_{i,3}(t, u_i(t_{i,k})) = 0$ in (59). This leads to the following quadratic equation:

$$k_1\big(0.5\Delta u_{i,i_p}(t_{i,k})\big)\tau^2 + \big(\Delta u_{i,i_p}(t_{i,k}) + k_1(\Delta v_{i,i_p}(t_{i,k}) - \varphi u_i(t_{i,k}))\big)\tau + \mathcal{C}_{i,3}(t_{i,k}, u_i(t_{i,k})) = 0.$$

The least positive root of the above equation is denoted as $\tau_{i,3}$ and we define $t_{i,k}^3 = t_{i,k} + \tau_{i,3}$. The case of having both roots negative corresponds to the constraint (11) not being violated, hence $t_{i,k}^3 = \infty$. Moreover, due to the added term in (67), it follows that $t_{i,k}^3 \geq t_{i,k} + T_d$.

Similarly for the safe merging constraint (14), the first time instant $t > t_{i,k}$ such that $\mathcal{C}_{i,4}(t, u_i(t_{i,k})) = 0$ can be obtained by solving the following cubic equation:

$$-k_4\frac{\varphi}{2L}u_i^2(t_{i,k})\tau^3 + \big(0.5\Delta u_{i,i_c}(t_{i,k}) - k_4\frac{3\varphi}{2L}u_i^2(t_{i,k}) + - k_4\frac{3\varphi}{2L}v_i(t_{i,k})u_i(t_{i,k})\big)\tau^2 + k_4\Big(\Delta u_{i,i_c}(t_{i,k}) - \frac{3\varphi}{L}v_i(t_{i,k})u_i(t_{i,k}) + (\Delta v_{i,i_c}(t_{i,k}) - \frac{\varphi}{L}v_i^2(t_{i,k}) - \frac{\varphi}{L}u_i(t_{i,k})x_i(t_{i,k}))\Big)\tau + \mathcal{C}_{i,4}(t_{i,k}, u_i(t_{i,k})) = 0,$$

where $\Delta v_{i,i_c}(t_{i,k}) = v_{i_c}(t_{i,k}) - v_i(t_{i,k})$, $\Delta u_{i,i_c}(t_{i,k}) = u_{i_c}(t_{i,k}) - u_i(t_{i,k})$.

The least positive root is denoted as $\tau_{i,4}$ and we define $t_{i,k}^4 = t_{i,k} + \tau_{i,4}$. Moreover, due to solving QP in (70) subject to the modified CBF constraint derived in (68), it follows that $t_{i,k}^4 \geq t_{i,k} + T_d$. The case of having all roots negative corresponds to the constraint (14) not being violated, hence $t_{i,k}^4 = \infty$.

Self-triggered control execution: First, it should be noted that the time instants $t_{i,k}^q$, $q = 1, \ldots, 4$ are obtained based on the safety constraints (3) and (4), as well as the vehicle state limitations (5). However, this choice can compromise the optimal performance of CAVs in the CZ. In particular, it is possible that the acceleration of a CAV stays constant for a long period of time if there are no safety constraints or vehicle state limit violations, whereas, as shown in Xiao, Cassandras, and Belta (2021), the optimal acceleration trajectory of the CAV in fact changes linearly. Therefore, in order to avoid this issue and minimize deviations from the optimal acceleration trajectory, one can impose a maximum allowable inter-event time, denoted by $T_{\max}$. To accomplish this, we can define

$$t_{i,k}^{\min} = \min\left\{t_{i,k}^1, t_{i,k}^2, t_{i,k}^3, t_{i,k}^4, t_{i,k} + T_{\max}\right\}. \tag{74}$$

The next update time instant for CAV i, i.e. $t_{i,k+1} = t_{i,\text{next}}$ should now be calculated. Towards this goal, consider the case where $t_{i,k}^{\min} \leq \min(t_{i_p,\text{next}}, t_{i_c,\text{next}})$, which corresponds to the next update time instant of CAV i occurring before the next control update of the preceding vehicle i_p or the conflict CAV i_c. Then, we can set $t_{i,k+1} = t_{i,\text{next}} = t_{i,k}^{\min}$ from (74).

The only remaining case is when $t_{i,k}^{\min} > \min(t_{i_p,\text{next}}, t_{i_c,\text{next}})$, which corresponds to either CAV i_p or i_c updating its control input sooner than CAV i, hence CAV i does not have access to their updated control input. Consequently, checking the constraints (11) and (14) is no longer valid. In this case, $t_i^{\text{next}} = \min(t_{i_p,\text{next}}, t_{i_c,\text{next}}) + T_d$ which implies that CAV i's next update time will be immediately after the update time of CAV i_p or i_c with a minimum inter-event time interval T_d.

By setting $t_{r,\text{next}}^{\min} = \min(t_{i_p,\text{next}}, t_{i_c,\text{next}})$, we can summarize the selection of the next self-triggered time instant as follows:

$$t_{i,\text{next}} = \begin{cases} t_{i,k}^{\min}, & t_{i,k}^{\min} \leq t_{r,\text{next}}^{\min} \\ t_{r,\text{next}}^{\min} + T_d, & \text{otherwise,} \end{cases} \tag{75}$$

Finally, in order to have $(t_{i,k} \bmod T_d) = 0$, we set $t_{i,\text{next}} = \lfloor \frac{t_{i,\text{next}}}{T_d} \rfloor \times T_d$.

It should be noted that the case of $t_{i,\text{next}} = t_{i_c,\text{next}}$ or $t_{i,\text{next}} = t_{i_p,\text{next}}$ corresponds to having identical next update times for CAV i and CAV i_c or CAV i_p so that they need to solve their QPs at the same time instant. However, in order for CAV i to solve its QP at the time instant $t_{i,k+1} = t_{i,\text{next}}$, it requires the updated control input of CAV i_c or CAV i_p, i.e. $u_{i_c}(t_{i,k+1})$ or $u_{i_p}(t_{i,k+1})$; this is practically not possible. In order to remedy this issue, whenever $t_{i,\text{next}} = t_{i_c,\text{next}}$ or $t_{i,\text{next}} = t_{i_p,\text{next}}$, CAV i solves its QP at $t_{i,k+1}$ by using u_M instead of $u_{i_c}(t_{i,k+1})$ and $u_{i_p}(t_{i,k+1})$ in (67) and (68). This corresponds to considering the worst case in $\sigma_{i,3}(t, T_d)$ and $\sigma_{i,4}(t, T_d)$. Moreover, since calculating the next update time $t_{i,k+2}$ also depends on $u_{i_c}(t_{i,k+1})$ and $u_{i_p}(t_{i,k+1})$, CAV i in this case acts similar to the time-driven case with assigned $t_{i,k+2} = t_{i,k+1} + T_d$. Then, at the next time instant $t_{i,k+2}$, CAV i can obtain the updated control inputs of CAV i_c and CAV i_p from the coordinator and follows the proposed self-triggered scheme.

3.3. Communication schemes

C.1. **Event-Triggered Communication Scheme.**

As mentioned earlier, a coordinator is responsible for exchanging information among CAVs (but does not exert any control). To accommodate event-triggered communication, the coordinator table in Fig. 1 is extended as shown in Table 2 so that it includes "relevant CAV info" data for each CAV i. In particular, in addition to the states of CAV i in column 2, denoted by $\mathbf{x}_i(t_i)$, the states of CAVs $r \in \mathcal{R}_i(t_i)$ are included, denoted by $\mathbf{x}_r(t_i)$, in column 3, where CAV r affects the constraints of CAV i, i.e., $r \in \mathcal{R}_i(t_i)$. In an event-driven scheme, frequent communication is generally not needed, since it occurs only when an event is triggered. CAV i updates its state in the coordinator table and re-solves a QP in two cases depending on which event occurs: (i) Event 1 triggered by i. The first step is state synchronization: CAV i requests current states from all relevant CAVs and the coordinator updates these (column 3), as well as the state $\mathbf{x}_i(t_i)$ (column 2). CAV i then solves its QP while the coordinator notifies all CAVs $r \in \mathcal{R}_i(t_i)$ of the new CAV i state so they can update their respective boundary set $S_r(t_i)$. This may trigger an Event 2(r) to occur at some future event time as in (32); such an event cannot be triggered instantaneously, as it takes some finite time for a bound in $S_r(t_i)$ to be reached because of Lipschitz continuity in the dynamics. In addition, the coordinator notifies all CAVs j such that $i \in \mathcal{R}_j(t_i)$ (i.e. i is relevant to j) so that they can update their bounds $S_j(t_i)$ respectively. (ii) Event 2(r) is triggered by $r \in \mathcal{R}_i(t_i)$. When CAV r reaches the boundary set $S_r(t_i)$ it notifies the coordinator to update its state (column 2). The coordinator passes on this information to all CAVs j where $r \in \mathcal{R}_j(t_i)$, which includes i since $r \in \mathcal{R}_i(t_i)$, and the corresponding state of r is updated (column 3). Then, CAV i re-solves its QP and the coordinator updates t_i to the current time and the state $\mathbf{x}_i(t_i)$ (column 2) and the state $\mathbf{x}_r(t_i)$(column 3). The rest of the process is the same as in case (i). Note that any update in CAV i's state due to a triggered event can immediately affect only CAVs $l > i$ such that i is relevant to l. If an "event chain" ensues, the number of events is bounded by $N(t_i)$.

Table 2
Extended coordinator table from Fig. 1 for event triggered control.

Extended coordinator table			
Index	CAV info	Relevant CAV info	Lane
0	$\mathbf{x}_0(t_0)$	–	Main
1	$\mathbf{x}_1(t_1)$	–	Main
2	$\mathbf{x}_2(t_2)$	$\mathbf{x}_1(t_2)$	Merging
3	$\mathbf{x}_3(t_3)$	$\mathbf{x}_1(t_3)$, $\mathbf{x}_2(t_3)$	Merging
4	$\mathbf{x}_4(t_4)$	$\mathbf{x}_1(t_4)$, $\mathbf{x}_3(t_4)$	Main
5	$\mathbf{x}_5(t_5)$	$\mathbf{x}_4(t_5)$	Main

Remark 3. It is possible to simplify the communication scheme by assuming that each CAV can measure (through local sensors) the state of its relevant CAVs (i.e. in the case of CAV i, the states of the CAV $r \in \mathcal{R}_i(t_i)$). Thus, CAVs can check for violations not only in their own state boundaries $S_i(t_i)$ but also in their relevant CAV state boundaries, $S_r(t_i)$. The same applies to the case where CAVs have a direct vehicle-to-vehicle (V2V) communication capability.

C.2. **Self-Triggered Communication Scheme**.

In view of the constraints (11) and (14), CAV i requires knowledge of $t_{i_p,\text{last}}$, $v_{i_p}(t_{i,k})$, $x_{i_p}(t_{i,k})$, $t_{i_c,\text{last}}$, $v_{i_c}(t_{i,k})$, and $x_{i_c}(t_{i,k})$ at time instant $t_{i,k}$. Hence, at each time instant that it accesses the coordinator, it needs to download the recorded data of CAV i_p and i_c. Then, the required updated information at $t_{i,k}$ for CAV i_p can be calculated as $v_{i_p}(t_{i,k}) = v_{i_p}(t_{i_p,\text{last}}) + (t_{i,k} - t_{i_p,\text{last}})u_{i_p}(t_{i_p,\text{last}})$, $x_{i_p}(t_{i,k}) = x_{i_p}(t_{i_p,\text{last}}) + (t_{i,k} - t_{i_p,\text{last}})v_{i_p}(t_{i_p,\text{last}}) + \frac{1}{2}(t_{i,k} - t_{i_p,\text{last}})^2 u_{i_p}(t_{i_p,\text{last}})$, with similar information calculated for CAV i_c. Note that the information for CAV i_p may also be obtained from the on-board sensors at CAV i if such are available. There are two key differences between the event-triggered and self-triggered approaches in the communication scheme as follows: (i). In the self-triggered approach in addition to the states of the CAVs $\mathbf{x}_i(t_{i,\text{last}})$, control input $u_i(t_{i,\text{last}})$, current time instant $t_{i,\text{last}}$, and the next time instant of solving QP $t_{i,\text{next}}$ have to be shared. Whereas in the event-triggered only states of the CAVs and the states of the relevant CAVs at the time of the QP solving are needed. (ii). In this scheme, unlike the event-triggered scheme, the coordinator does

not notify other relevant CAVs when a particular CAV solves QP and updates its data since the next QP solving time is known and stored in the coordinator table. For example, when CAV i solves its QP there is no need for the CAVs j where $i \in R_j(t_i)$ to be notified as they are already aware. Instead, the coordinator only receives and stores the current time instant, states, control input, and the next time instant of solving QP of the CAV i. Also upon download request from a particular CAV at the time of solving QP, access to the data of the relevant CAVs r will be given to that particular CAV by the coordinator.

3.4. Comparison of control schemes

We briefly discuss the similarities and differences between the event-triggered and self-triggered control schemes. The infeasibility issue and excessive computational burden of solving QPs through a time-driven method are addressed in both approaches. In both approaches, the original CBFs are replaced with modified CBFs (i.e. (30) for $q = \{1, 2, 3, 4\}$ in the event-triggered approach and (50), (51), (67), and (68) in the self-triggered approach) to ensure the feasibility of the QP at the next time instant. In both approaches events determine the next time instant of QP. In the event-triggered scheme, the event occurrence is unknown to the vehicles and the onboard sensors are responsible for detecting such events, as in (32), whereas in the self-triggered approach the next event time can be analytically calculated as in (74). In both approaches, there are some parameters that can be adjusted to avoid conservativeness and potentially deal with delays and noise, such as $\boldsymbol{s}_i$ in the event-triggered approach, and T_d, T_{max} in the self-triggered approach. Due to the unpredictability of the events in the event-triggered approach, the communication scheme becomes slightly more complicated compared to the self-triggered approach. Finally in terms of results, the self-triggered approach was found to be more conservative compared to the event-triggered approach as will be discussed in the next section.

4. Simulation results

All algorithms in this section have been implemented in MATLAB. We used QUADPROG for solving QPs of the form (22), (31) and (70), LINGPROG for solving the linear programming in (26), (28) and (29), FMINCON for a nonlinear optimization problem arising when (26) and (28) become nonlinear, and ODE45 to integrate the vehicle dynamics.

We have considered the merging problem shown in Fig. 1 where CAVs are simulated according to Poisson arrival processes with an arrival rate which is fixed for the purpose of comparing the time-driven approach and the event-driven schemes (over different bound values in (32) for the event-triggered scheme and with different $T_{\max}$ for the self-triggered scheme). The initial speed $v_i(t_{i,0})$ is also randomly generated with a uniform distribution over [15 m/s, 20 m/s] at the origins O and O', respectively. The parameters for (20), (31), and (70) are: $L = 400$ m, $\varphi = 1.8$ s, $\delta = 0$ m, $u_{\max} = 4.905$ m/s^2, $u_{\min} = -5.886$ m/s^2, $v_{\max} = 30$ m/s, $v_{\min} = 0$ m/s, $k_1 = k_2 = k_3 = k_4 = 1, \lambda = 10$ and $T_d = 0.05$. The sensor sampling rate is 20 Hz, sufficiently high to avoid missing any triggering event as discussed earlier. The control update period for the time-driven control is $\Delta t = 0.05$ s. For the event-triggered scheme, we let the bounds $S = [s_x, s_v]$ be the same for the all CAVs in the network and vary them between the values of {[0.5, 1.5], [0.5, 2], [0.5, 2.5]}. For the self-triggered scheme, we set $T_{\max} \in \{0.5, 1, 1.5, 2\}$.

In our simulations, we included the computation of a more realistic energy consumption model (Kamal, Mukai, Murata, & Kawabe, 2012) to supplement the simple surrogate L_2-norm (u^2) model in our analysis: $f_v(t) = f_{\text{cruise}}(t) + f_{\text{accel}}(t)$ with $f_{\text{cruise}}(t) = \omega_0 + \omega_1 v_i(t) + \omega_2 v_i^2(t) + \omega_3 v_i^3(t), f_{\text{accel}}(t) = (r_0 + r_1 v_i(t) + r_2 v_i^2(t))u_i(t)$. where we used typical values for parameters $\omega_1, \omega_2, \omega_3, r_0, r_1$ and, r_2 as reported in Kamal et al. (2012).

Our results from several simulations corresponding to three different methods under the same conditions with different values for the relative weight of energy vs time are shown in Tables 3 and 4: the time-driven method, the event-triggered scheme, and, the self-triggered scheme. We observe that by using the event-triggered and self-triggered approaches we are able to significantly reduce the number of infeasible QP cases (up to 95%) compared to the time-driven approach. At the same time, the overall number of instances when a QP needs to be solved has also decreased up to 68% and 80% in the event-triggered and self-triggered approaches, respectively. Note that the large majority of infeasibilities is due to holding acceleration constant over an inappropriate sampling time, which can invalidate the forward invariance property of CBFs over the entire time interval. These infeasible cases were eliminated by the event-triggered and self-triggered schemes. However, another source of infeasibility is due to conflicts that may arise between the CBF constraints and the control bounds in a QP. This cannot be remedied through the proposed event-triggered or self-triggered QPs; it can, however, be dealt with by the introduction of a sufficient condition that guarantees no such conflict, as described in Xiao et al. (2022).

In Tables 3 and 4, we can also observe some loss of performance (i.e. average travel time increases hence road throughput decreases) in both approaches as the values of the bound parameters in the event-triggered approach and $T_{\max}$ in the self-triggered approach increase, hence increasing conservativeness. On the other hand, this decreases the computational load expressed in terms of the number of QPs that are solved in both methods, illustrating the trade-off discussed in previous sections. For instance, when $\alpha = 0.25$, the number of QPs (i.e. the indicator of computation load) and thereby the number of communications between the CAVs is reduced by 49% and 80% in the event-triggered and self-triggered scheme respectively, compared to the time-driven approach. There is also an apparent discrepancy in the energy consumption results: when the L_2-norm of the control input is used as a simple metric for energy consumption, the values are higher under event-triggered and self-triggered control, whereas the detailed fuel consumption model shows lower values compared to time-driven control. This is due to the fact that u_i^2 penalizes CAVs when they decelerate, whereas this is not actually the case under a realistic fuel consumption model.

We can also visualize the results presented in Tables 3 and 4 by showing the variation of the average objective functions in (7) with respect to α for different choices of $[s_x, s_v]$ and $T_{\max}$, shown in Fig. 2 and Fig. 3, respectively. As seen in Fig. 2, by selecting higher values for bounds in the event-triggered scheme and for $T_{\max}$ in the self-triggered scheme (being more conservative) the objective functions will also attain higher values, while the lowest cost (best performance) is reached under time-driven control.

Constraint violation. It is worth noting that an "infeasible" QP does not necessarily imply a constraint violation, since violating a CBF constraint does not always imply the violation of an original constraint in (3), (4), and (5). This is due to the conservative nature of a CBF whose intent is to *guarantee* the satisfaction of our original constraints. In order to explicitly show how an infeasible case may lead to a constraint violation and how this can be alleviated by the event-triggered and self-triggered schemes, we simulated 12 CAVs in the merging framework of Fig. 1 with the exact same parameter settings as before and with $S = [0.5, 1.5]$ in the event-triggered scheme, $T_{\max} = 1$ in the self-triggered scheme and $\beta = 5$. Fig. 4 shows the values of the rear-end safety constraint over time. One can see that the satisfaction of safety constraints is always guaranteed with the event-triggered

Table 3
CAV metrics under self-triggered (see Section 3.1) and time-driven control.

	Item	Self-Triggered				Time-driven modified CBF	Time-driven
	$T_{\max}$	0.5	1	1.5	2	$T_s = T_d = 0.05$	$T_s = 0.05$
$\alpha = 0.1$	Ave. Travel time	19.5	19.48	19.48	19.49	19.5	19.42
	Ave. $\frac{1}{2}u^2$	4.27	5.00	5.93	7.2	3.37	3.18
	Ave. Fuel consumption	31.86	32.21	32.64	33.23	31.32	31.61
	Computation load (Num of QPs solved)	20.46% (7252)	11.9% (4218)	10.87% (3854)	10.32%(3658)	100.5% (35636)	100% (35443)
	Num of infeasible cases	42	42	43	32	190	315
$\alpha = 0.25$	Ave. Travel time	15.57	15.56	15.57	15.62	15.58	15.44
	Ave. $\frac{1}{2}u^2$	14.33	15.10	15.68	16.68	13.38	13.34
	Ave. Fuel consumption	54.45	53.51	52.57	52.94	54.17	55.81
	Computation load (Num of QPs solved)	19.5% (5495)	13.68% (3857)	12.34% (3479)	12.72% (3588)	100.9% (28461)	100% (28200)
	Num of infeasible cases	27	27	28	24	249	341
$\alpha = 0.4$	Ave. Travel time	15.15	15.15	15.18	15.2	15.16	15.01
	Ave. $\frac{1}{2}u^2$	18.5	19.32	19.73	20.36	17.64	17.67
	Ave. Fuel consumption	55.23	53.35	52.67	52.95	54.93	56.5
	Computation load (Num of QPs solved)	20.4% (5591)	14.85% (4071)	13.69% (3754)	13.60% (3727)	101.0% (27695)	100% (27412)
	Num of infeasible cases	25	25	25	20	220	321
$\alpha = 0.5$	Ave. Travel time	14.79	14.79	14.82	14.89	14.8	14.63
	Ave. $\frac{1}{2}u^2$	25.5	25.84	26.43	27.5	24.86	25.08
	Ave. Fuel consumption	55.5	53.15	52.9	53.45	55.5	56.93
	Computation load (Num of QPs solved)	21.8% (5841)	16.7% (4322)	15.09% (4034)	15.17% (4054)	101.1% (27033)	100% (26726)
	Num of infeasible cases	19	20	20	20	250	341

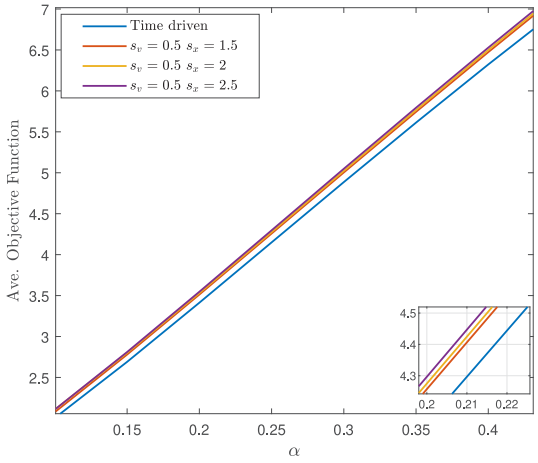


Fig. 2. Average objective function value with respect to α (time weight with respect to energy in (7)) for different selection of bounds in event-triggered approach (see Section 3.1).

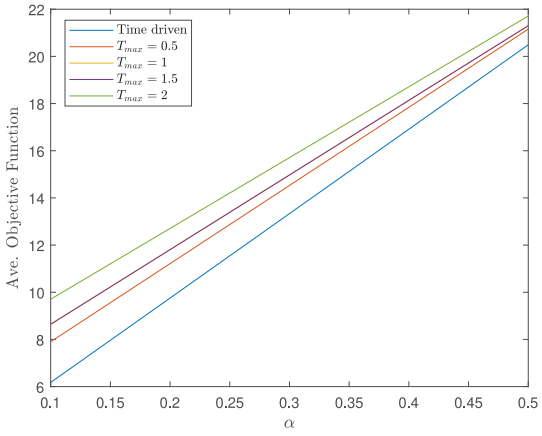


Fig. 3. Average objective function value with respect to α (time weight with respect to energy in (7)) for different selection of bounds in self-triggered approach (see Section 3.2).

and self-triggered approach as there is no infeasible case and the value of the constraint $b_1(\mathbf{x}(t))$) is well above zero. In contrast, we see a clear violation of the constraint in the time-driven scheme in the cases of CAVs 8 depicted by the blue line (see Fig. 2).

Robustness. We have investigated the robustness of both schemes with respect to different forms of uncertainty, such as modeling and computational errors, by adding two noise terms to the vehicle dynamics: $\dot{x}_i(t) = v_i(t) + w_1(t)$, $\dot{v}_i(t) = u_i(t) + w_2(t)$, where $w_1(t)$, $w_2(t)$ denote two random processes defined in an appropriate probability space which, in our simulation, are set to be uniformly distributed over $[-2, 2]$ and $[-0.2, 0.2]$, respectively. We repeated the prior simulation experiment with added noise and results shown in Figs. 5 and 6. We can see that the event-triggered and self-triggered schemes with almost similar performance because of their conservativeness keep the functions well away from the unsafe region (below 0) in contrast to the time-driven approach where we observe constraint violations due to noise, e.g., CAVs 3, 4, and, 9 in Fig. 5 and CAV 8 in Fig. 6 and .

Remark 4. With the aim of validating our controllers in the presence of noise, delays, and system uncertainty, we designed a laboratory test bed using small mobile robots to emulate CAVs. The results of the implementation demonstrate how the event-triggered scheme is computationally efficient and can handle measurement uncertainties and noise compared to time-driven control while guaranteeing safety. For instance, in this setup, the main sources of noise are the Optitrack localization system we use and the IMU of the robots (to measure their velocity). Based on the IMU datasheet, the total RMS noise is 0.05 deg/s and for the Optitrack system the noise can be up to 1 mm depending on the position of the markers installed on the robots (Nagymáté & Kiss, 2018). A video of the implementation of our event-triggered scheme for the safe and optimal merging of mobile robots may be

Table 4
CAV metrics under event-triggered (see Section 3.1) and time-driven control.

	Item	Event triggered			Time driven
	Bounds	$s_v = 0.5, s_x = 1.5$	$s_v = 0.5, s_x = 2$	$s_v = 0.5, s_x = 2.5$	$\Delta t = 0.05$
$\alpha = 0.1$	Ave. Travel time	19.61	19.73	19.65	19.42
	Ave. $\frac{1}{2}u^2$	4.45	4.81	5.16	3.18
	Ave. Fuel consumption	31.77	31.51	31.04	31.61
	Computation load (Num of QPs solved)	50% (17853)	47% (16778)	34% (12168)	100% (35443)
	Num of infeasible cases	42	42	43	315
$\alpha = 0.25$	Ave. Travel time	15.82	15.88	15.95	15.44
	Ave. $\frac{1}{2}u^2$	13.93	14.06	14.25	13.34
	Ave. Fuel consumption	52.12	51.69	51.42	55.81
	Computation load (Num of QPs solved)	51% (14465)	51% (14403)	48% (13707)	100% (28200)
	Num of infeasible cases	27	27	28	341
$\alpha = 0.4$	Ave. Travel time	15.4	15.46	15.53	15.01
	Ave. $\frac{1}{2}u^2$	18.04	18.13	18.22	17.67
	Ave. Fuel consumption	53.155	52.77	52.42	56.5
	Computation load (Num of QPs solved)	54% (14089)	53% (14072)	49% (13573)	100% (27412)
	Num of infeasible cases	25	25	25	321
$\alpha = 0.5$	Ave. Travel time	15.05	15.11	15.17	14.63
	Ave. $\frac{1}{2}u^2$	24.94	24.88	24.93	25.08
	Ave. Fuel consumption	53.65	53.41	53.21	56.93
	Computation load (Num of QPs solved)	51% (13764)	51% (13758)	50% (13415)	100% (26726)
	Num of infeasible cases	20	20	20	341

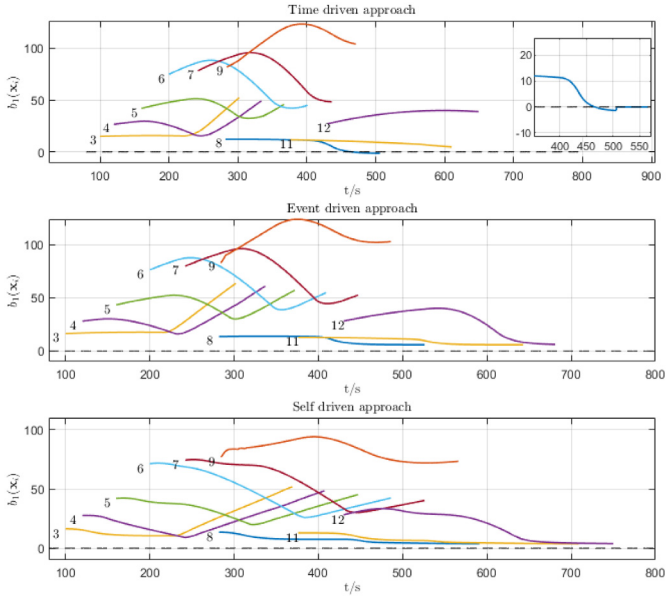


Fig. 4. The variation of rear-end safety constraints for the time-driven, event-triggered and self-triggered approaches. As can be seen, the blue line associated with CAV 8 violates the rear-end safety constraint by becoming negative in the time-driven approach, whereas in both event and self-triggered approaches it stays well above the zero level. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

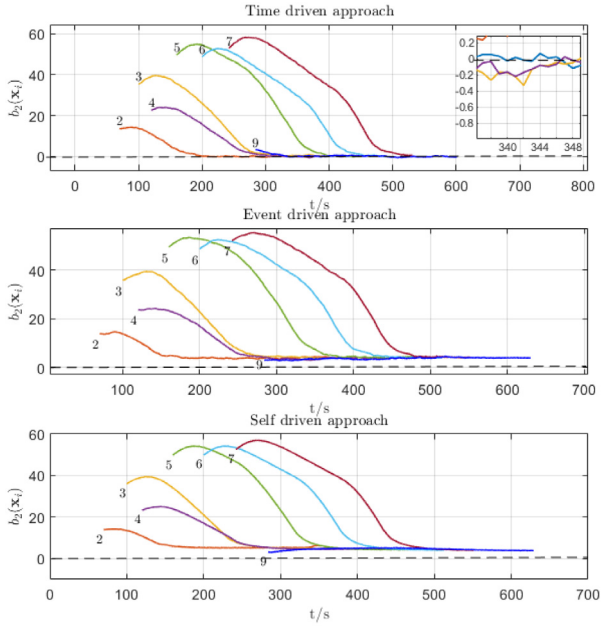


Fig. 5. The variation of safe merging constraints for the time-driven, event-triggered and self-triggered approaches in the presence of noise. It can be observed that the yellow, purple, and blue lines corresponding to safe merging constraints for CAVs 3, 4 and 9 violate these constraints by becoming negative, whereas in the proposed event and self-triggered schemes they stay well above zero. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

found at https://www.youtube.com/watch?v=qwhLjEskPS8. More detailed results and analysis can be found in Sabouni, Ahmad, Xiao, Cassandras, and Li (2023).

5. Conclusions

The problem of controlling CAVs in conflict areas of a traffic network subject to hard safety constraints can be solved through a combination of tractable optimal control problems and the use of CBFs. These solutions can be derived by discretizing time and solving a sequence of QPs. However, the feasibility of each QP cannot be guaranteed over every time step. When this is due to the lack of a sufficiently high control update rate, we have shown that this problem can be alleviated through either an event-triggered scheme or a self-triggered scheme, while at the same time reducing the need for communication among CAVs, thus

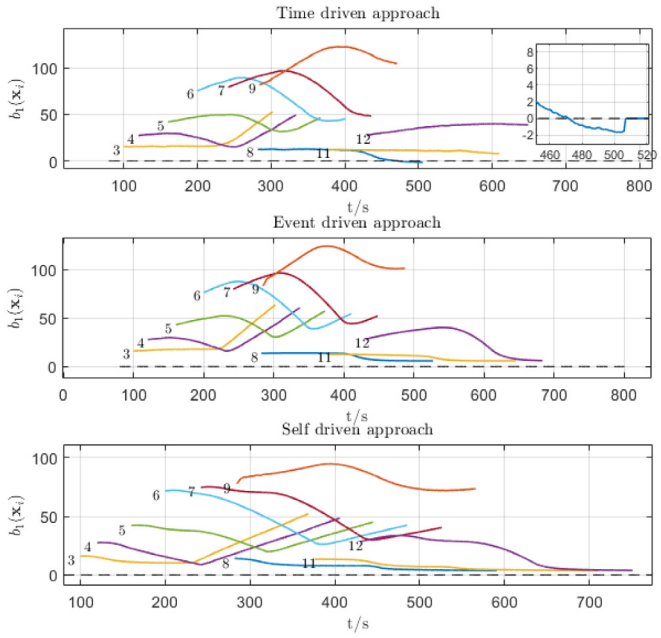


Fig. 6. The variation of rear-end safety constraints for the time-driven, event-triggered, and self-triggered approaches in the presence of noise. It can be observed that the blue line corresponding to rear-end safety constraints for CAV 8 violates these constraints by becoming negative, whereas in the proposed event and self-triggered schemes it stays well above zero. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

lowering computational costs and the chance of security threats. Ongoing work is targeted at eliminating all possible infeasibilities through the use of sufficient conditions based on the work in Xiao et al. (2022) added to the QPs, leading to complete solutions of CAV control problems with full safety constraint guarantees. As unnecessary communication is avoided, the proposed schemes are less likely to suffer from communication-related issues. However, imperfect communication can still be a problem and, as part of our future work, we will investigate how packet loss and delays affect the proposed methods and how they can possibly be mitigated. The first step in this direction has been the successful validation of these schemes in a laboratory-scale test-bed where noise, delays, packet loss, and uncertainty are inevitable.

Acknowledgments

This work was supported in part by U.S. National Science Foundation under grants ECCS-1931600, DMS-1664644, CNS-1645681, CNS-2149511, by AFOSR, United States under grant FA9550-19-1-0158, by ARPA-E, United States under grant DE-AR0001282, by the MathWorks, United States, and by NPRP grant (12S-0228-190177) from the Qatar National Research Fund, a member of the Qatar Foundation (the statements made herein are solely the responsibility of the authors). The authors are with the Division of Systems Engineering and Center for Information and Systems Engineering, Boston University, Brookline, MA, 02446, USA, CSAIL, MIT, USA and Electrical Engineering Department, Qatar University, Doha, Qatar.

References

Ahmad, H., Sabouni, E., Xiao, W., Cassandras, C. G., & Li, W. (2023). Evaluations of cyber attacks on cooperative control of connected and autonomous vehicles at bottleneck points. In *Symposium on vehicles security and privacy*.

Ames, A. D., Coogan, S., Egerstedt, M., Notomista, G., Sreenath, K., & Tabuada, P. (2019). Control barrier functions: Theory and applications. In *2019 18th European control conference* (pp. 3420–3431). IEEE.

Ames, A., Xu, X., Grizzle, J., & Tabuada, P. (2017). Control barrier function based quadratic programs for safety critical systems. *IEEE Transactions on Automatic Control*, *62*(8), 3861–3876. http://dx.doi.org/10.1109/TAC.2016.2638961.

Au, T., & Stone, P. (2010). Motion planning algorithms for autonomous intersection management. In *Workshops at the twenty-fourth AAAI conference on artificial intelligence*.

Berg, V. A. V. D., & Verhoef, E. T. (2016). Autonomous cars and dynamic bottleneck congestion: The effects on capacity, value of time and preference heterogeneity. *Transportation Research, Part B (Methodological)*, *94*, 43–60. http://dx.doi.org/10.1016/j.trb.2016.08.018, URL https://www.sciencedirect.com/science/article/pii/S0191261515300643.

Cao, W., Mukai, M., Kawabe, T., Nishira, H., & Fujiki, N. (2015). Cooperative vehicle path generation during merging using model predictive control with real-time optimization. *Control Engineering Practice*, *34*, 98–105.

de Waard, D., Dijksterhuis, C., & Brookhuis, K. (2009). Merging into heavy motorway traffic by young and elderly drivers. *Accident Analysis and Prevention*, *41*(3), pp. 588–597.

Dolk, V. S., Ploeg, J., & Heemels, W. P. M. H. (2017). Event-triggered control for string-stable vehicle platooning. *IEEE Transactions on Intelligent Transportation Systems*, *18*(12), 3486–3500. http://dx.doi.org/10.1109/TITS.2017.2738446.

Dresner, K., & Stone, P. (2004). Multiagent traffic management: a reservation-based intersection control mechanism. In *Proceedings of the third international joint conference on autonomous agents and multiagent systems, 2004* AAMAS 2004, (pp. 530–537).

Fagnant, D. J., & Kockelman, K. (2015). Preparing a nation for autonomous vehicles: Opportunities, barriers and policy recommendations. *Transportation Research Part A: Policy and Practice*, *77*, 167–181. http://dx.doi.org/10.1016/j.tra.2015.04.003, URL https://www.sciencedirect.com/science/article/pii/S0965856415000804.

Garcia, C. E., Prett, D. M., & Morari, M. (1989). Model predictive control: Theory and practice—A survey. *Automatica*, *25*(3), 335–348.

Holkar, K., & Waghmare, L. M. (2010). An overview of model predictive control. *International Journal of control and automation*, *3*(4), 47–63.

Hu, B. (2021). Stochastic stability analysis for vehicular networked systems with state-dependent bursty fading channels: A self-triggered approach. *Automatica*, *123*, Article 109352. http://dx.doi.org/10.1016/j.automatica.2020.109352, URL https://www.sciencedirect.com/science/article/pii/S0005109820305549.

Kamal, M. A. S., Mukai, M., Murata, J., & Kawabe, T. (2012). Model predictive control of vehicles on urban roads for improved fuel economy. *IEEE Transactions on Control Systems Technology*, *21*(3), 831–841.

Kavalchuk, I., Kolbasov, A., Karpukhin, K., Terenchenko, A., et al. (2020). The performance assessment of low-cost air pollution sensor in city and the prospect of the autonomous vehicle for air pollution reduction. In *IOP conference series: Materials science and engineering: vol. 819*, (no. 1), IOP Publishing, Article 012018.

Li, L., Wen, D., & Yao, D. (2014). A survey of traffic control with vehicular communications. *IEEE Transactions on Intelligent Transportation Systems*, *15*(1), 425–432. http://dx.doi.org/10.1109/TITS.2013.2277737.

Nagymáté, G., & Kiss, R. M. (2018). Motion capture system validation with surveying techniques. *Materials Today: Proceedings*, *5*(13, Part 2), 26501–26506. http://dx.doi.org/10.1016/j.matpr.2018.08.107, URL https://www.sciencedirect.com/science/article/pii/S2214785318320789. 34th Danubia Adria Symposium on Advances in Experimental Mechanics, 19 - 22 September 2017, Trieste, Italy.

Ong, P., & Cortés, J. (2018). Event-triggered control design with performance barrier. In *2018 IEEE conf. on decision and control* (pp. 951–956). IEEE.

Rajamani, R., Tan, H., Law, B. K., & Zhang, W. (2000). Demonstration of integrated longitudinal and lateral control for the operation of automated vehicles in platoons. *IEEE Transactions on Control Systems Technology*, *8*(4), 695–708.

Rios-Torres, J., & Malikopoulos, A. A. (2017). A survey on the coordination of connected and automated vehicles at intersections and merging at highway on-ramps. *IEEE Transactions on Intelligent Transportation Systems*, *18*(5), pp. 1066–1077. http://dx.doi.org/10.1109/TITS.2016.2600504.

Rios-Torres, J., Malikopoulos, A., & Pisu, P. (2015). Online optimal control of connected vehicles for efficient traffic flow at merging roads. In *2015 IEEE 18th international conf. on intelligent transportation systems* (pp. 2432–2437). IEEE.

Sabouni, E., Ahmad, H. S., Xiao, W., Cassandras, C. G., & Li, W. (2023). Optimal control of connected automated vehicles with event-triggered control barrier functions: A test bed for safe optimal merging. In *2023 IEEE conference on control technology and applications* (pp. 321–326). http://dx.doi.org/10.1109/CCTA54093.2023.10253379.

Schrank, D., Eisele, B., Lomax, T., & Bak, J. (2015). *2015 Urban mobility scorecard*.

Taylor, A. J., Ong, P., Cortés, J., & Ames, A. D. (2020). Safety-critical event triggered control via input-to-state safe barrier functions. *IEEE Control Systems Letters*, *5*(3), 749–754.

Vogel, K. (2003). A comparison of headway and time to collision as safety indicators. *Accident Analysis and Prevention*, *35*(3), 427–433.

Wu, J., Yan, F., & Abbas-Turki, A. (2013). Mathematical proof of effectiveness of platoon-based traffic control at intersections. In *16th international IEEE conference on intelligent transportation systems* (pp. 720–725). IEEE.

Xiao, W., & Belta, C. (2019). Control barrier functions for systems with high relative degree. In *Proc. of 58th IEEE conf. on decision and control* (pp. 474–479). Nice, France.

Xiao, W., Belta, C., & Cassandras, C. G. (2021). Event-triggered safety-critical control for systems with unknown dynamics. In *2021 60th IEEE conf. on decision and control* (pp. 540–545). http://dx.doi.org/10.1109/CDC45484.2021.9682792.

Xiao, W., Belta, C. A., & Cassandras, C. G. (2022). Sufficient conditions for feasibility of optimal control problems using control barrier functions. *Automatica*, *135*, Article 109960. http://dx.doi.org/10.1016/j.automatica.2021.109960, URL https://www.sciencedirect.com/science/article/pii/S0005109821004866.

Xiao, W., & Cassandras, C. (2020). Decentralized optimal merging control for connected and automated vehicles with optimal dynamic resequencing. In *2020 American control conference* (pp. 4090–4095). http://dx.doi.org/10.23919/ACC45564.2020.9147805.

Xiao, W., & Cassandras, C. (2021). Decentralized optimal merging control for connected and automated vehicles with safety constraint guarantees. *Automatica*, *123*, Article 109333. http://dx.doi.org/10.1016/j.automatica.2020.109333, URL https://www.sciencedirect.com/science/article/pii/S0005109820305331.

Xiao, W., Cassandras, C., & Belta, C. (2021). Bridging the gap between optimal trajectory planning and safety-critical control with applications to autonomous vehicles. *Automatica*, *129*, Article 109592. http://dx.doi.org/10.1016/j.automatica.2021.109592, URL https://www.sciencedirect.com/science/article/pii/S0005109821001126.

Xu, H., Feng, S., Zhang, Y., & Li, L. (2019). A grouping-based cooperative driving strategy for CAVs merging problems. *IEEE Transactions on Vehicular Technology*, *68*(6), pp. 6125–6136.

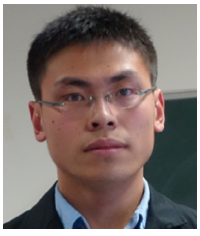
Yang, G., Belta, C., & Tron, R. (2019). Self-triggered control for safety critical systems using control barrier functions. In *2019 American control conference* (pp. 4454–4459). http://dx.doi.org/10.23919/ACC.2019.8814657.

Zhang, Y., & Cassandras, C. (2019). Decentralized optimal control of connected automated vehicles at signal-free intersections including comfort-constrained turns and safety guarantees. *Automatica*, *109*, p. 108563. http://dx.doi.org/10.1016/j.automatica.2019.108563.

Zhang, K., De La Fortelle, A., Zhang, D., & Wu, X. (2013). Analysis and modeled design of one state-driven autonomous passing-through algorithm for driverless vehicles at intersections. In *2013 IEEE 16th international conference on computational science and engineering* (pp. 751–757). IEEE.

Ehsan Sabouni is a Ph.D. candidate in System Engineering at Boston University, focusing his research on multi-agent safety-critical control algorithms for multi-agent systems. He received his B.Sc. in Electrical Engineering from Isfahan University of Technology in Iran, graduating with honors in 2018. For his academic achievements in connected and autonomous vehicle controls, Ehsan has received awards including the Outstanding Student-Paper award at the 2023 IEEE Conference on Control Technology and Applications.

Christos G. Cassandras received a B.S. degree from Yale University, a M.S.E.E. degree from Stanford University, and S.B. and Ph.D. degrees from Harvard University. In 1982–1984, he was with ITP Boston, Inc. In 1984–1996, he was a Faculty Member with the Department of Electrical and Computer Engineering, University of Massachusetts/Amherst. He is currently a Distinguished Professor of Engineering with Boston University, Brookline, MA, USA, the Head of the Division of Systems Engineering, and Professor of Electrical and Computer Engineering. He has authored or coauthored seven books, and over 500 refereed papers in the areas of discrete event and hybrid systems, cooperative control, stochastic optimization, and computer simulation, with applications to computer and sensor networks, manufacturing systems, and transportation systems. Dr. Cassandras serves on several journal Editorial Boards and was the Editor-in-Chief of the IEEE Transactions on Automatic Control (1998–2009). He was the 2012 President of the IEEE Control Systems Society (CSS). He has been a plenary/keynote speaker at numerous international conferences and has also been an IEEE Distinguished Lecturer. He is the recipient of several awards, including the 2011 IEEE Control Systems Technology Award, the Distinguished Member Award of the IEEE Control Systems Society (2006), the 1999 Harold Chestnut Prize (IFAC Best Control Engineering Textbook) among others. He is a Member of Phi Beta Kappa and Tau Beta Pi. He is a Fellow of the IEEE and a Fellow of the IFAC.



Wei Xiao (Member, IEEE) received the B.Sc. degree in mechanical engineering and automation from the University of Science and Technology Beijing, Beijing, China, the M.Sc. degree in robotics from the Chinese Academy of Sciences (Institute of Automation), Beijing, China, and the Ph.D. degree in systems engineering from Boston University, Boston, MA, USA, in 2013, 2016, and 2021, respectively. He is currently a Postdoctoral Associate with the Massachusetts Institute of Technology, Cambridge, MA, USA. His current research interests include control theory and machine learning, with particular emphasis on robotics and traffic control. Dr. Xiao was the recipient of an Outstanding Student Paper Award at the 2020 IEEE Conference on Decision and Control.

Nader Meskin received the B.Sc. degree from Sharif University of Technology, Tehran, Iran, in 1998, the M.Sc. degree from the University of Tehran, Tehran, in 2001, and the Ph.D. degree in electrical and computer engineering in 2008 from Concordia University, Montreal, QC, Canada., He was a Postdoctoral Fellow at Texas A&M University at Qatar, Doha, Qatar, from January 2010 to December 2010. He is currently a Professor at Qatar University, Doha, and an Adjunct Associate Professor at Concordia University. He has published more than 280 refereed journal and conference papers. His research interests include fault diagnosis and health monitoring of complex industrial systems, multiagent systems, mathematical modeling and control of biomedical systems, and intelligent transportation systems.