

Optimal Control of Connected Automated Vehicles with Event-Triggered Control Barrier Functions: a Test Bed for Safe Optimal Merging

Ehsan Sabouni^{*,1}, H.M. Sabbir Ahmad^{*,1}, Wei Xiao², Christos G. Cassandras¹ and Wenchao Li¹

Abstract—We address the problem of safely coordinating a network of Connected and Automated Vehicles (CAVs) in conflict areas of a traffic network. Such problems can be solved through a combination of tractable optimal control problems and 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 online over discrete time steps. However, guaranteeing the feasibility of the CBF-based QP method within each discretized time interval requires the careful selection of time steps which need to be sufficiently small. This creates computational requirements and communication rates between agents which may limit the controller's application to real CAVs. We tackle this limitation by adopting an event-triggered control approach for CAVs such that the next QP is triggered by properly defined events with a safety guarantee. We present a laboratory-scale test bed developed to emulate merging roadways using mobile robots as CAVs. We present results to demonstrate how the event-triggered scheme is computationally efficient and can handle measurement uncertainties and noise compared to time-driven control while guaranteeing safety.

I. INTRODUCTION

The effective traffic management of Connected and Automated Vehicles (CAVs) through control and coordination has brought the promise of resolving long-lasting problems in transportation networks such as accidents, congestion, and unsustainable energy consumption [1]. To date, both centralized [2] and decentralized [3] methods have been proposed to tackle the problem of CAVs in conflict area such as intersections, roundabouts, and merging roadways; an overview of such methods may be found in [4]. In decentralized methods, as opposed to centralized ones, each CAV is responsible for its own on-board computation with information from other vehicles limited to a set of neighbors [5]. One approach is to formulate a constrained optimal control problem jointly minimizing travel time and energy consumption, e.g., for optimal merging [3] or crossing intersections [6]. The complexity of obtaining the solution, however, necessitates the use of online control methods like

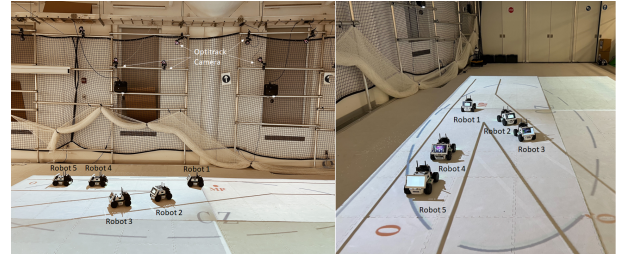


Fig. 1. The merging experiment setup where the control zone (CZ) starts from point O in the main road and point O' in the merging road and ends at the merging point MP .

Control Barrier Functions (CBFs) [7], and Model Predictive Control (MPC) techniques [8] for real-world applications.

An approach combining optimal control solutions with CBFs (termed OCBF) was recently presented in [9]. This optimal tracking problem can be 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 [7]. However, 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 be often seen to be violated.

One approach is to use an *event-triggered* scheme instead. A general event-triggered framework for CBFs has been proposed in [10]. It remains to be shown how this framework can be used for real CAVs with event-triggered communication, optimization, and control in the presence of measurement uncertainties, noise, and multiple safety constraints. As part of this paper, we adapt the framework in [10] to CAVs and implement it in a laboratory-scale test bed using mobile robots to emulate CAVs. Several optimal control methods have been developed for CAVs and implemented on actual mobile robots [11]–[14]. To our knowledge, there is no general *safety-guaranteed* control algorithm for CAVs implemented on real robots for merging roadways.

Our contribution: The contributions of the paper are as follows. (i) We follow the approach presented in [10] and introduce an *event-triggered* framework for coordination and control of CAVs in merging roadways. This approach provides manifold benefits, namely (a) guaranteeing safety constraints are satisfied in the presence of measurement noise and model uncertainties which are inevitable in practical applications, as opposed to the time-driven scheme, and (b) significant reduction in the number of QPs solved, thereby reducing the need for unnecessary communication among CAVs and overall computational overload. (ii) we have also

^{*}These authors contributed equally to this paper.

¹Division of Systems Engineering and Department of Electrical & Computer Engineering, Boston University, Boston, MA, USA, {esabouni, sabbir92, cgc, wenchao}@bu.edu

²Computer Science & Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA, USA {weixy@mit.edu}

This work was supported in part by ONR grant N000114-19-1-2496, NSF under grants CNS-1932162, ECCS-1931600, DMS-1664644, CNS-1645681, CNS-2149511, by AFOSR under grant FA9550-19-1-0158, by ARPA-E under grant DE-AR0001282, by the MathWorks 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).

implemented it in a laboratory-scale test bed using mobile robots as shown in Fig 1. The test bed allows us to test a variety of cooperative control algorithms for CAVs under multiple safety constraints in the presence of disturbances that cannot be captured in a simulated setting. In this test bed, all computation, including checking for events and resolving a QP to obtain a new control input, are performed on-board in a real-time manner.

II. PROBLEM FORMULATION AND EVENT-TRIGGERED CONTROL

In this section, we review the problem of cooperation of CAVs at conflict road networks focusing on merging roadways. Following [9], we define a *Control Zone* (CZ) to be an area within which CAVs can communicate with each other or with a coordinator responsible for facilitating the exchange of information within this CZ. As an example, Fig. 2 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 [9]. In such a setting, assuming all traffic consists of CAVs, a finite horizon constrained optimal control problem can be formulated aiming to determine trajectories that jointly minimize travel time and energy consumption through the CZ and guaranteeing safety constraints are always satisfied.

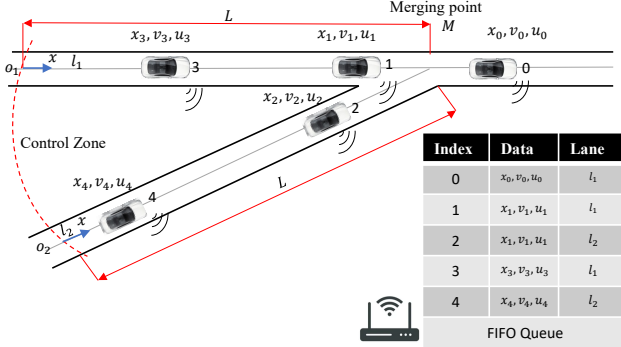


Fig. 2. Illustration of cooperative control of CAVs at merging roadways.

Let $S(t)$ be the set of First-In-First-Out (FIFO)-ordered indices of all CAVs located in the CZ at time t along with the CAV that has just left the CZ (whose index is 0 as shown in Fig.2). $N(t)$ is defined as the cardinality of $S(t)$. Thus, a CAV arriving at time t is assigned $N(t)$ as its index. All CAV indices in the CZ $S(t)$ decrease by one when a CAV leaves the CZ and the CAV whose index is -1 is dropped.

The vehicle dynamics for each CAV $i \in S(t)$ take the following form

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

where $x_i(t)$ denotes the distance from the origin at which CAV i arrives, $v_i(t)$, and $u_i(t)$ denotes the velocity and control input (acceleration) of CAV i , respectively.

Let t_i^0 and t_i^f denote the time that CAV i arrives at the origin and leaves the CZ at its exit point, respectively. Constraints for any conflict area can be listed as follows:

Constraint 1 (Safety constraints): CAV i must maintain adequate spacing from the CAV i_p immediately preceding it (if there is any) in the same lane, denoted as $z_{i,i_p}(t) = x_{i_p}(t) - x_i(t)$, hence CAV i must satisfy:

$$z_{i,i_p}(t) \geq \varphi v_i(t) + \delta, \quad \forall t \in [t_i^0, t_i^f], \quad (2)$$

where φ denotes the reaction time and δ is a given minimum safe distance and depends on the length of these two CAVs.

Constraint 2 (Safe merging): At the MP M , a CAV i should maintain adequate safe space to avoid any collision, i.e.,

$$z_{i,i_m}(t_i^m) \geq \varphi v_i(t_i^m) + \delta, \quad (3)$$

where i_m is the index of the CAV that may collide with CAV i at the merging point M as shown in Fig. 2. The determination of CAV i_m depends on the policy adopted for sequencing CAVs through the CZ. For the merging roadway in Fig. 2 under FIFO, we have $i_m = i - 1$ if $i - 1 \neq i_p$ and $t_i^f = t_i^m$ since the MP defines the exit from the CZ.

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

$$v_{min} \leq v_i(t) \leq v_{max}, \quad \forall t \in [t_i^0, t_i^f], \quad (4)$$

$$u_{min} \leq u_i(t) \leq u_{max}, \quad \forall t \in [t_i^0, t_i^f], \quad (5)$$

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

Decentralized optimal control problem formulation.

Our goal is to determine a control law jointly minimizing the travel time and energy consumption subject to constraints 1-3 for each $i \in S(t)$ governed by the dynamics (1). Expressing energy through $\frac{1}{2}u_i^2(t)$ and normalizing travel time and energy, the objective function can be formulated as follows:

$$\min_{u_i(t), t_i^f} J_i(u_i(t), t_i^f) := \beta_i(t_i^f - t_i^0) + \int_{t_i^0}^{t_i^f} \frac{1}{2}u_i^2(t)dt, \quad (6)$$

where $\beta_i := \frac{\alpha_i \max\{u_{max}^2, u_{min}^2\}}{2(1-\alpha_i)}$, is a positive adjustable weight to penalize travel time relative to the energy cost of CAV i and $\alpha_i \in [0, 1]$. Note that the solution is *decentralized* in the sense that CAV i requires information only from CAVs i_p and i_m required in (2) and (3).

The OCBF approach. Firstly, we derive the CBFs that ensure the constraints (2), (3), and (4) are always satisfied, subject to the vehicle dynamics (1) by defining $f(\mathbf{x}_i(t)) = [v_i(t), 0]^T$ and $g(\mathbf{x}_i(t)) = [0, 1]^T$. Each of these constraints can be written in the form of $b_q(\mathbf{x}(t)) \geq 0$, $q \in \{1, \dots, n\}$ where n stands for the number of constraints and $\mathbf{x}(t) = [x_1(t), x_2(t), \dots, x_{N(t)}(t)]$. The CBF method (details provided in [9]) maps a constraint $b_q(\mathbf{x}(t)) \geq 0$ onto a new constraint which is *linear* in the control input and takes the general form

$$L_f b_q(\mathbf{x}(t)) + L_g b_q(\mathbf{x}(t))u_i(t) + \gamma_q(b_q(\mathbf{x}(t))) \geq 0. \quad (7)$$

To obtain the CBF constraint for the safety constraint (2) we set $b_1(\mathbf{x}_i(t), \mathbf{x}_{i_p}(t)) = z_{i,i_p}(t) - \varphi v_i(t) - \delta$ and since it is differentiable, the CBF constraint for (2) is

$$v_{i_p}(t) - v_i(t) - \varphi u_i(t) + k_1 b_1(\mathbf{x}_i(t), \mathbf{x}_{i_p}(t)) \geq 0, \quad (8)$$

where the class- $\mathcal{K}$ function $\gamma_1(\mathbf{x}) = k_1(b_1(\mathbf{x}_i(t), \mathbf{x}_{i_p}(t)))$ is chosen here to be linear. Deriving the CBF constraint for the safe merging constraint (3) 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, by following the technique used in [9], we replace φ in (3) by $\Phi(x_i(t)) = \varphi \frac{x_i(t)}{L}$, where L is the length of the road traveled by the CAV from the CZ entry to the MP, and use $b_2(\mathbf{x}_i(t), \mathbf{x}_{i_m}(t)) = z_{i,i_m}(t) - \Phi(x_i(t))v_i(t) - \delta$ so that the CBF constraint for safe merging (3) becomes

$$v_{i_m}(t) - v_i(t) - \frac{\varphi}{L}v_i^2(t) - \varphi \frac{x_i(t)}{L}u_i(t) + k_2(b_2(\mathbf{x}_i(t), \mathbf{x}_{i_m}(t))) \geq 0. \quad (9)$$

The speed constraints in (4) are also easily transformed into CBF constraints by defining $b_3(\mathbf{x}_i(t)) = v_{max} - v_i(t)$ and $b_4(\mathbf{x}_i(t)) = v_i(t) - v_{min}$. This yields:

$$-u_i(t) + k_3 b_3(\mathbf{x}_i(t)) \geq 0, \quad u_i(t) + k_4 b_4(\mathbf{x}_i(t)) \geq 0, \quad (10)$$

for the maximum and minimum velocity, respectively.

As a last step in the OCBF approach, we use a Control Lyapunov Function (CLF) to track the CAV speed to a desired value $v_i^{ref}(t)$ setting $V(\mathbf{x}_i(t)) = (v_i(t) - v_i^{ref}(t))^2$ and express the CLF constraint as follows:

$$L_f V(\mathbf{x}_i(t)) + L_g V(\mathbf{x}_i(t))u_i(t) + c_3 V(\mathbf{x}_i(t)) \leq e_i(t), \quad (11)$$

where $e_i(t)$ makes this a soft constraint.

Finally, we can formulate the OCBF problem as follows:

$$\min_{u_i(t), e_i(t)} J_i(u_i(t), e_i(t)) := \int_{t_i^0}^{t_i^f} \left[\frac{1}{2} (u_i(t) - u_i^{ref}(t))^2 + \lambda e_i^2(t) \right] dt \quad (12)$$

subject to vehicle dynamics (1), the CBF constraints (8), (9), (10), and CLF constraint (11). In this approach, (i) we solve the unconstrained optimal control problem in (6) which yields u_i^{ref} , (ii) the resulting u_i^{ref} is optimally tracked such that constraints including CBF constraints (8), (9), (10) satisfied. A common way to solve this dynamic optimization problem is to discretize $[t_i^0, t_i^f]$ into intervals $[t_i^0, t_i^0 + \Delta], \dots, [t_i^0 + k\Delta, t_i^0 + (k+1)\Delta], \dots$ with equal length Δ and solving (12) 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:

$$\min_{u_{i,k}, e_{i,k}} \left[\frac{1}{2} (u_{i,k} - u_i^{ref}(t_{i,k}))^2 + \lambda e_{i,k}^2 \right], \quad (13)$$

subject to the constraints (8), (9), (10), (11), and (5) where all constraints are linear in the decision variables. We refer to this as the *time-driven* approach. As pointed out earlier, the main problem with this approach is that there is no

guarantee for the feasibility of each CBF-based QP, as it requires a small enough discretization time which is not always possible to achieve. Also it is worth mentioning that synchronization is required amongst all CAVs which can be difficult to impose in real-world applications.

III. EVENT-TRIGGERED CONTROL

In this paper, we adopt an *event-triggered control* scheme whereby a QP is solved when one of two possible events (as defined next) is detected. We will show that this can guarantee the satisfaction of the safety constraints which cannot be offered by the time-driven approach.

Let $t_{i,k}$, $k = 1, 2, \dots$, be the time instants when CAV i solves the QP in (13). 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. We define a subset of the state space of CAV i at time $t_{i,k}$ such that:

$$\mathbf{x}_i(t_{i,k}) - \mathbf{s}_i \leq \mathbf{x}_i(t) \leq \mathbf{x}_i(t_{i,k}) + \mathbf{s}_i, \quad (14)$$

where $\mathbf{s}_i \in \mathbb{R}_{\geq 0}^2$ is a parameter vector whose choice will be discussed later. We denote the set of states of CAV i that satisfy (14) at time $t_{i,k}$ by

$$S_i(t_{i,k}) = \{\mathbf{y}_i \in \mathbf{X} : \mathbf{x}_i(t_{i,k}) - \mathbf{s}_i \leq \mathbf{y}_i \leq \mathbf{x}_i(t_{i,k}) + \mathbf{s}_i\}. \quad (15)$$

In addition, let C_i be the feasible set of our original constraints (2),(3) and (4) defined as

$$C_i := \{\mathbf{x}_i \in \mathbf{X} : b_q(\mathbf{x}_i) \geq 0, q \in \{1, 2, 3, 4\}\}, \quad (16)$$

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 of (7), where $q \in \{1, 2, 3, 4\}$, as shown next.

Let us start with the first term in (7), $L_f b_q(\mathbf{x}_i(t), \mathbf{x}_r(t))$ for CAV i , rewritten as $L_f b_q(\mathbf{y}_i(t), \mathbf{y}_r(t))$ with $\mathbf{y}_i(t)$ as in (15) and $r \in \mathcal{R}_i(t)$ in $\mathbf{y}_r(t)$ and $\mathbf{x}_r(t)$ stands for “relevant” CAVs affecting the constraint of i (i.e., $r \in \{i_p, i_m\}$ in Fig. 2). Let $b_{q,f_i}^{min}(t_{i,k})$ be the minimum possible value of the term $L_f b_q(\mathbf{x}_i(t), \mathbf{x}_r(t))$ over the interval $(t_{i,k}, t_{i,k+1}]$ for each $q \in \{1, 2, 3, 4\}$ over the set $\bar{S}_i(t_{i,k}) \cap \bar{S}_r(t_{i,k})$:

$$b_{q,f_i}^{min}(t_{i,k}) = \min_{\substack{\mathbf{y}_i \in \bar{S}_i(t_{i,k}) \\ \mathbf{y}_r \in \bar{S}_r(t_{i,k})}} L_f b_q(\mathbf{y}_i(t), \mathbf{y}_r(t)), \quad (17)$$

where $\bar{S}_i(t_{i,k}) := \{\mathbf{y}_i \in C_i \cap S_i(t_{i,k})\}$. Similarly, we can define the minimum value of the third term in (7):

$$b_{\gamma_q}^{min}(t_{i,k}) = \min_{\substack{\mathbf{y}_i \in \bar{S}_i(t_{i,k}) \\ \mathbf{y}_r \in \bar{S}_r(t_{i,k})}} \gamma_q(\mathbf{y}_i(t), \mathbf{y}_r(t)). \quad (18)$$

For the second term in (7), note that $L_g b_q(\mathbf{x}_i(t), \mathbf{x}_r(t))$ is a constant for $q \in \{1, 3, 4\}$, therefore there is no need for any minimization. However, $L_g b_2(\mathbf{x}_i(t))$ in (9) is state dependent and needs to be considered for the minimization. Since $x_i(t) \geq 0$, note that $L_g b_2(\mathbf{x}_i(t), \mathbf{x}_r(t))$ is always negative,

therefore, the limit value $b_{2,g_i}^{min}(t_{i,k}) \in \mathbb{R}$, can be determined as follows:

$$b_{2,g_i}^{min}(t_{i,k}) = \begin{cases} \min_{\substack{\mathbf{u}_i \in \tilde{S}_i(t_{i,k}) \\ \mathbf{y}_r \in S_r(t_{i,k})}} L_g b_2(\mathbf{x}_i(t)), & \text{if } u_{i,k} \geq 0 \\ \max_{\substack{\mathbf{u}_i \in \tilde{S}_i(t_{i,k}) \\ \mathbf{y}_r \in S_r(t_{i,k})}} L_g b_2(\mathbf{x}_i(t)), & \text{otherwise,} \end{cases} \quad (19)$$

where the sign of $u_{i,k}$, $i \in S(t_{i,k})$ can be determined by simply solving the CBF-based QP (12) at time $t_{i,k}$.

Thus, the condition that can guarantee the satisfaction of (8),(9) and (10) in the interval $(t_{i,k}, t_{i,k+1}]$ is given by

$$b_{q,f_i}^{min}(t_{i,k}) + b_{q,g_i}^{min}(t_{i,k})u_{i,k} + b_{\gamma_q}^{min}(t_{i,k}) \geq 0, \quad (20)$$

for $q \in \{1, 2, 3, 4\}$. In order to apply this condition to the QP (12), we just replace (7) by (20) as follows:

$$\min_{u_{i,k}, e_{i,k}} \left[\frac{1}{2}(u_{i,k} - u_i^{ref}(t_{i,k}))^2 + \lambda e_{i,k}^2 \right] \text{ s.t. (11), (20), (5)} \quad (21)$$

It is important to note that each instance of the QP (21) is now triggered by one of the following two events where $k = 1, 2, \dots$ is an event (rather than time step) counter:

Event 1: the state measurement 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 $r = i_m$. Thus, Event 2 is further identified by the CAV which triggers it and denoted accordingly by **Event 2(r)**, $r \in \mathcal{R}_i(t_{i,k-1})$.

The state measurements are obtained from external sensors which introduce noise in the state values. Thus, we denote the state measurements of CAV i by $\tilde{\mathbf{x}}_i(t) = \mathbf{x}_i(t) + \mathbf{w}(t)$, where $\mathbf{w}(t)$ is a random but bounded noise term. As a result, $t_{i,k}, k = 1, 2, \dots$ is unknown in advance but can be determined by CAV i through:

$$t_{i,k} = \min \left\{ t > t_{i,k-1} : |\tilde{\mathbf{x}}_i(t) - \tilde{\mathbf{x}}_i(t_{i,k-1})| = \mathbf{s}_i \right. \\ \text{or } |\tilde{\mathbf{x}}_{i_p}(t) - \tilde{\mathbf{x}}_{i_p}(t_{i,k-1})| = \mathbf{s}_{i_p} \\ \left. \text{or } |\tilde{\mathbf{x}}_{i_m}(t) - \tilde{\mathbf{x}}_{i_m}(t_{i,k-1})| = \mathbf{s}_{i_m} \right\}, \quad t_{i,1} = 0 \quad (22)$$

The following theorem formalizes our analysis by showing that if new constraints of the general form (20) holds, then our original CBF constraints (8),(9) and (10) also hold. The proof follows the same lines as that of a more general theorem in [10] and, therefore, is omitted.

Theorem 1: Given a CBF $b_q(\mathbf{x}_i(t), \mathbf{x}_r(t))$ with relative degree one, let $t_{i,k+1}$, $k = 1, 2, \dots$ be determined by (22) with $t_{i,1} = 0$ and $b_{q,f_i}^{min}(t_{i,k})$, $b_{q,g_i}^{min}(t_{i,k})$, $b_{\gamma_q}^{min}(t_{i,k})$ for $q = \{1, 2, 3, 4\}$ obtained through (17), (18), and (19). Then, any control input $u_{i,k}$ that satisfies (20) for all $q \in \{1, 2, 3, 4\}$ within the time interval $[t_{i,k}, t_{i,k+1})$ renders the set C_i forward invariant for the dynamic system defined in (1).

The selection of the parameters $\mathbf{s}_i$ 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 that trigger instances of the QPs becomes,

thus reducing the total computational cost. The choice of parameter $\mathbf{s}_i$ can also render our algorithm more robust to measurement uncertainties and noise, hence making it useful for real-world implementation. To account for measurement uncertainties, the value of each bound should be greater than or equal to the measurement noise, i.e., the inequality $\mathbf{s}_i \geq \sup \mathbf{w}(t)$ should hold componentwise.

IV. TEST BED FOR COORDINATED MERGING

The event-triggered coordinated merging process was implemented in a lab environment using mobile robots as CAVs, while the coordinator was implemented on a laptop installed in the vicinity of the robots. The overall implementation can be divided into two main parts: (i) communication and (ii) control. A block-level illustration of the architecture of the robots firmware is presented in Fig. 3.

Communication. The OCBF approach only requires vehicle-to-infrastructure (V2I) communication, whereby the coordinator is responsible for exchanging information among robots. The communication was implemented using a 5G wireless LAN using ROS. The events are state-dependent, thus each robot requires its position information. Since the robots are not equipped with GPS, the position and orientation (POSE) information was obtained using the Optitrack localization system by the coordinator using ROS topics over Ethernet as shown in Fig. 3. The coordinator uses the localization information to index the robots upon arrival in the CZ for constructing the queue table based on the FIFO scheme as illustrated in Fig. 2 and transmits the indices to them. Additionally, the POSE information of each robot and its relevant robots are transmitted to every robot periodically while in the CZ.

Each robot is responsible for checking its own state to detect any violation in its state bounds in (15). When such an event occurs, the robot updates its control input by re-solving the QP in (21) and informs the coordinator with its newly obtained state (i.e., velocity and position) by publishing a ROS topic. The structure of data packets sent by any robot with index i is described using the vector $(i, \tilde{x}_{i,long}, \tilde{x}_{i,lat}, \tilde{v}_i)^T$ where $\tilde{x}_{i,long}$ and $\tilde{x}_{i,lat}$ denote the position along and orthogonal to the direction of the main road, respectively and $\tilde{v}_i$ denotes the linear speed along the direction of the road of robot i . It then becomes the responsibility of the coordinator to provide this information to the relevant robots (i.e., those that might be affected). There can be at most two robots that can be relevant to any robot i , the one which is immediately following it in the same lane and the robot that will merge behind it from the other lane. The data are published as ROS topics by the coordinator separately for each one (or two) of the robot(s). The robots subscribe to the topics published for them using the index as the identifier. As can be seen from Fig. 3, the robot firmware contains a dedicated thread to handle the ROS messages from the coordinator. Finally, the notified robots decide whether they need to re-solve their QP or maintain their control input until the next triggering event by checking for a potential bound violation. They either need to re-solve

their QP, and send their updated state to the coordinator or carry on with their current control input until the next event.

Control. The event-triggered QP in (21) for any robot i returns the acceleration $u_{i,k}^*$, which has to be controlled along with the trajectory of every robot in the CZ to prevent them from deviating from the road. The two control objectives are achieved using feedback control. The linear acceleration for the robots cannot be directly controlled, hence the desired acceleration $u_{i,k}^*$ for any robot i is mapped onto its desired linear velocity $v_i^*(t)$ using the mapping below.

$$v_i^*(t + \Delta T) = u_{i,k}^* \Delta T + v_i^*(t) \quad t \in [t_{i,k}, t_{i,k+1}), \forall k \quad (23)$$

where ΔT is the period of the control loop execution. In order to track the acceleration $u_{i,k}^*$ of robot i the value of ΔT needs to be generally chosen to be a fifth of the inter-event time, $t_{i,k+1} - t_{i,k}$. However, since events occur asynchronously, ΔT is chosen to be equal to the sampling rate of the encoders used for measuring the velocity $v_i(t)$.

It is desired that the robot follows the center of the lane while in the CZ, which is programmed as a set of way-points. A PID controller is used to adjust the lateral error for any robot i at time t is denoted by $\tilde{e}_i = \tilde{x}_{i,lat}(t) - L_c$, (where L_c is the center of the road) by adjusting the steering angle. Due to the nature of event-triggered control, two threads are dedicated in every robot to achieve the control tasks, where the first thread is responsible for controlling the motion of the robot (executed every ΔT seconds) and the other for the event-triggered control (executed asynchronously upon the occurrence of any event as shown in Fig. 3).

V. NUMERICAL RESULTS

We have considered the merging problem shown in Fig. 1 where robots arrive to the predefined CZ. Please refer to [15] to find the simulation details and implementation parameters.

A. Simulation results

We first used a MATLAB-based simulation environment using the python modules developed for actual implementation in the test bed. Note that some of the parameters for simulation and implementation are different as we had designed a lab testbed using mobile robots. The metrics used for comparing the performance of the event-triggered scheme to time-driven scheme are detailed in [15].

Table I summarizes our results from 6 separate simulations (for more extensive results see [15]) corresponding to both the event-triggered and time-driven methods under the same conditions with different values for the relative weight of energy vs time as shown in the table I. We observe that by using the event-triggered approach we can significantly reduce the number of infeasible QP cases (up to 86%) 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 50% in the event-triggered case. Note that the large majority of infeasibilities are due to holding the control input constant over the sampling period, which can invalidate the forward invariance property of CBFs over the entire time interval. These infeasible cases

were eliminated by the event-triggered approach. 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 event-triggered approach; it can, however, be dealt with by the introduction of a sufficient condition that guarantees no such conflict, as described in [9].

B. Test Bed Implementation Results

Although simulation results indicate the efficacy of the event-triggered OCBF approach, the results of the lab test bed implementation are significant in two ways: firstly, they demonstrate that the control algorithm can be implemented and executed in real-time, which is essential for safety-critical systems. Secondly, they demonstrate the robustness of the algorithm to the various noise sources present in real-world applications. To illustrate the aforementioned points, two different experiments have been considered. The first experiment involves 5 robots for comprehensive implementation and the results are depicted in Fig. 4 illustrating the satisfaction of both rear-end safety and merging constraints during the whole experiment duration. In Fig. 4, each line specified by the CAV number represents a time plot constraint for that CAV in the CZ. Note that all the CAVs are not present in the plot, as not all the CAVs are constrained. In the second experiment, the initial conditions of the 2 robots (that have to merge together safely coming from two different roads) are deliberately chosen to demonstrate that *the time-driven approach cannot guarantee safety* due to uncertainties and imperfect measurements while the proposed event-triggered scheme with a proper choice of bounds (as discussed in section III) can maintain a safe merging distance between two robots as depicted in Fig 5.

The following web link contains a video of our implementation of the event triggered scheme for two scenarios (details can be found in [15]) with different arrival sequences based on FIFO passing sequence: <https://www.youtube.com/watch?v=qwhLjEskPS8>.

VI. CONCLUSION

The problem of controlling CAVs in conflict areas of a traffic network subject to hard safety constraints by the use of CBFs can be solved through a sequence of QPs. However, the lack of feasibility guarantees for each QP, as well as control update synchronization, motivate an event-triggering scheme. In this paper, we have presented an event-triggered framework for the automated control of merging roadways (generalizable to any conflict area) and designed a lab test bed for the cooperative control of CAVs using mobile robots. In future, we will validate this algorithm in an extensive test bed for various other conflict points.

REFERENCES

- [1] T. L. D. Schrank, B. Eisele and J. Bak, "2015 urban mobility scorecard," 2015.
- [2] X. Pan, B. Chen, S. A. Evangelou, and S. Timotheou, "Optimal motion control for connected and automated electric vehicles at signal-free intersections," in *2020 59th IEEE Conference on Decision and Control (CDC)*, 2020, pp. 2831–2836.

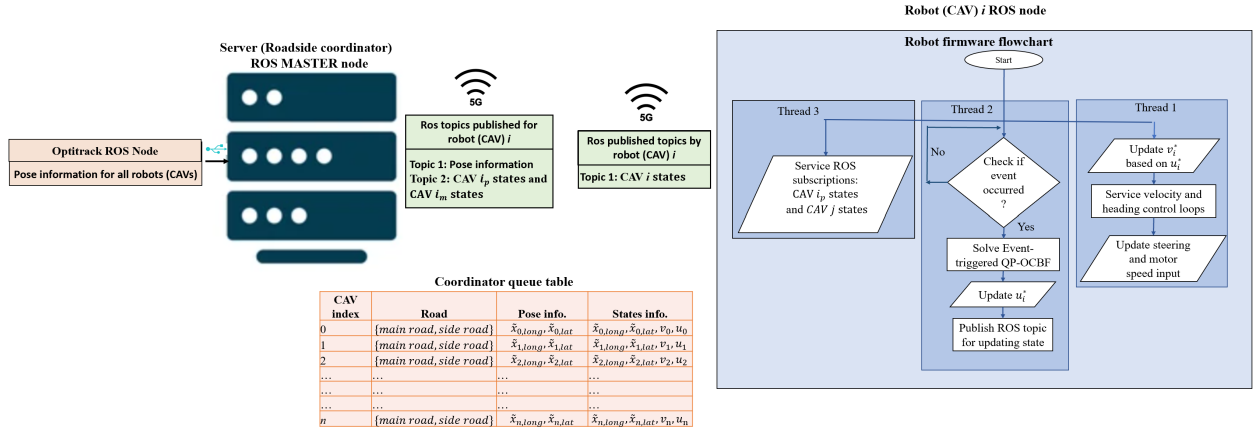


Fig. 3. Illustration of the testbed implementation.

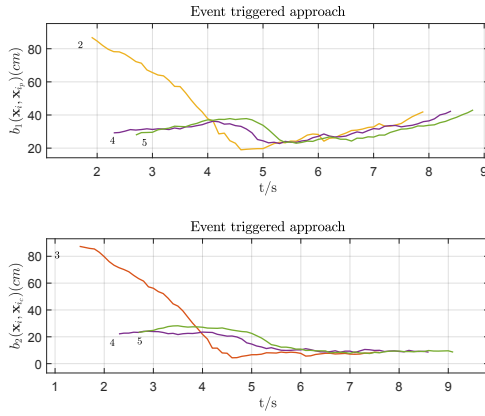


Fig. 4. Rear end and safe merging constraints of 5 robots.

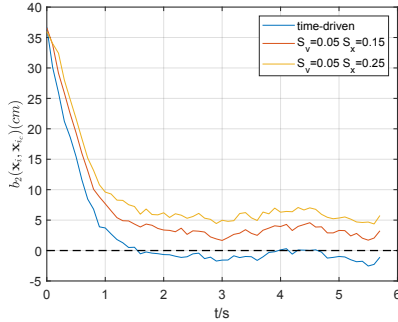


Fig. 5. Safe merging constraints of 2 robots under event-triggered scheme with 2 different event bounds and time-driven control schemes. Note the safe merging constraint violation under time-driven control.

	Perf. metrics	Event triggered	Time Triggered
$\alpha = 0.25$	Ave. Travel time	15.53	15.01
	Ave. $\frac{1}{2} u^2$	14.25	13.34
	Ave. Fuel consumption	51.42	55.81
	Computation load	48% (13707)	100% (28200)
	Num of infeasible cases	28	341
$\alpha = 0.4$	Ave. Travel time	15.53	15.01
	Ave. $\frac{1}{2} u^2$	18.22	17.67
	Ave. Fuel consumption	52.42	56.5
	Computation load	49% (13573)	100% (27412)
	Num of infeasible cases	25	321
$\alpha = 0.5$	Ave. Travel time	15.17	14.63
	Ave. $\frac{1}{2} u^2$	24.93	25.08
	Ave. Fuel consumption	53.21	56.93
	Computation load	50% (13415)	100% (26726)
	Num of infeasible cases	20	341

TABLE I

CAV METRICS UNDER EVENT-DRIVEN AND TIME-DRIVEN CONTROL.

- [3] F. Xu and T. Shen, "Decentralized optimal merging control with optimization of energy consumption for connected hybrid electric vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 6, pp. 5539–5551, 2022.
- [4] J. Rios-Torres and A. A. Malikopoulos, "A survey on the coordination of connected and automated vehicles at intersections and merging at highway on-ramps," *IEEE Trans. on Intelligent Transportation Systems*, vol. 18, no. 5, pp. 1066–1077, 2017.
- [5] J. Rios-Torres, A. Malikopoulos, and P. Pisu, "Online optimal control of connected vehicles for efficient traffic flow at merging roads," in *2015 IEEE 18th International Conf. on Intelligent Transportation Systems*. IEEE, 2015, pp. 2432–2437.
- [6] B. Li, Y. Zhang, N. Jia, and X. Peng, "Autonomous intersection management over continuous space: A microscopic and precise solution via computational optimal control," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 17071–17076, 2020, 21st IFAC World Congress.
- [7] W. Xiao, C. G. Cassandras, and C. Belta, *Safe Autonomy with Control Barrier Functions: Theory and Applications*. Springer Nature, 2023.
- [8] J. B. Rawlings, D. Q. Mayne, and M. M. Diehl, *Model Predictive Control: Theory, Computation, and Design*. Nob Hill Publishing.
- [9] W. Xiao, C. Cassandras, and C. Belta, "Bridging the gap between optimal trajectory planning and safety-critical control with applications to autonomous vehicles," *Automatica*, vol. 129, p. 109592, 2022.
- [10] W. Xiao, C. Belta, and C. G. Cassandras, "Event-triggered control for safety-critical systems with unknown dynamics," *IEEE Transactions on Automatic Control*, pp. 1–16, 2022.
- [11] K. Berntorp, T. Hoang, and S. Di Cairano, "Motion planning of autonomous road vehicles by particle filtering," *IEEE Transactions on Intelligent Vehicles*, vol. 4, no. 2, pp. 197–210, 2019.
- [12] N. Hyldmar, Y. He, and A. Prorok, "A fleet of miniature cars for experiments in cooperative driving," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 3238–3244.
- [13] M. Mukai, H. Natori, and M. Fujita, "Model predictive control with a mixed integer programming for merging path generation on motor way," in *2017 IEEE Conference on Control Technology and Applications (CCTA)*, 2017, pp. 2214–2219.
- [14] L. E. Beaver, B. Chalaki, A. M. I. Mahbub, L. Zhao, R. Zayas, and A. A. Malikopoulos, "Demonstration of a time-efficient mobility system using a scaled smart city," *Vehicle System Dynamics*, vol. 58, no. 5, pp. 787–804, feb 2020.
- [15] E. Sabouni, H. M. S. Ahmad, W. Xiao, C. G. Cassandras, and W. Li, "Optimal control of connected automated vehicles with event-triggered control barrier functions: a test bed for safe optimal merging," 2023. [Online]. Available: <https://arxiv.org/abs/2306.01871>