

A General Framework for Decentralized Safe Optimal Control of Connected and Automated Vehicles in Multi-Lane Signal-Free Intersections

Huile Xu¹, Wei Xiao², *Member, IEEE*, Christos G. Cassandras³, *Life Fellow, IEEE*,
Yi Zhang⁴, *Member, IEEE*, and Li Li⁵, *Fellow, IEEE*

Abstract—We address the problem of optimally controlling Connected and Automated Vehicles (CAVs) arriving from four multi-lane roads at a signal-free intersection where they conflict in terms of safely crossing (including turns) with no collision. The objective is to jointly minimize the travel time and energy consumption of each CAV while ensuring safety. This problem was solved in prior work for single-lane roads. A direct extension to multiple lanes on each road is limited by the computational complexity required to obtain an explicit optimal control solution. Instead, we propose a general framework that first converts a multi-lane intersection problem into a decentralized optimal control problem for each CAV with less conservative safety constraints than prior work. We then employ a method combining optimal control and control barrier functions, which has been shown to efficiently track tractable unconstrained optimal CAV trajectories while also guaranteeing the satisfaction of all constraints. Simulation examples are included to show the effectiveness of the proposed framework under symmetric and asymmetric intersection geometries and different CAV sequencing policies.

Index Terms—Connected and automated vehicles (CAVs), optimal control, control barrier function (CBF).

Manuscript received October 6, 2020; revised March 27, 2021, June 23, 2021, October 12, 2021, and December 20, 2021; accepted February 9, 2022. The work of Huile Xu, Wei Xiao, and Christos G. Cassandras was supported in part by the NSF under Grant ECCS-1509084, Grant DMS-1664644, Grant CNS-1645681, Grant IIS-1723995, and Grant CPS-1446151; in part by the Air Force Office of Scientific Research (AFOSR) under Grant FA9550-19-1-0158; in part by the Advanced Research Projects Agency-Energy (ARPA-E) under Grant DE-AR0001282; and in part by the MathWorks. The work of Huile Xu, Yi Zhang, and Li Li was supported in part by the National Key Research and Development Program of China under Grant 2018YFB1600600. The Associate Editor for this article was H. L. Vu. (*Corresponding authors: Yi Zhang; Li Li.*)

Huile Xu is with the Department of Automation, BNRist, Tsinghua University, Beijing 100084, China, also with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with the Division of Systems Engineering and Center for Information and Systems Engineering, Boston University, Brookline, MA 02446 USA (e-mail: huile.xu@ia.ac.cn).

Wei Xiao and Christos G. Cassandras are with the Division of Systems Engineering and Center for Information and Systems Engineering, Boston University, Brookline, MA 02446 USA (e-mail: xiaowei@bu.edu; cgc@bu.edu).

Yi Zhang is with the Department of Automation, BNRist, Tsinghua University, Beijing 100084, China, and also with the Tsinghua-Berkeley Shenzhen Institute (TBSI), Nanshan, Shenzhen 518055, China (e-mail: zhyi@tsinghua.edu.cn).

Li Li is with the Department of Automation, BNRist, Tsinghua University, Beijing 100084, China (e-mail: li-li@tsinghua.edu.cn).

Digital Object Identifier 10.1109/TITS.2022.3151080

I. INTRODUCTION

INTERSECTIONS are the main bottlenecks for urban traffic. As reported in [1], congestion in these areas causes US commuters to spend 6.9 billion hours more on the road and to purchase an extra 3.1 billion gallons of fuel, resulting in a substantial economic loss to society. The coordination and control problems at intersections are challenging in terms of safety, traffic efficiency, and energy consumption [1], [2].

The emergence of Connected and Automated Vehicles (CAVs) provides a promising way for better planning and controlling trajectories to reduce congestion and ultimately improve safety as well as efficiency. Enabled by vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication, CAVs can exchange real-time operational data with vehicles in their vicinity and communicate with the infrastructure [3]. Based on these technologies, researchers have made significant progress in the optimal control of CAVs at intersections. One of the prevailing ideas is to formulate an optimization problem whose decision variables are both crossing sequences and control inputs (e.g., desired velocity and desired acceleration). Hult *et al.* [4] formulated the traffic coordination problem at a three-way intersection as an optimal control problem where it is required that the trajectories of any two vehicles do not intersect in order to guarantee safety. However, it is not explicitly explained how to realize these safety constraints. Some recent studies show that we can realize safety constraints by introducing binary variables to represent crossing sequences, leading to the formulation of Mixed-Integer Linear Programming (MILP) problems [5]. Though some techniques, such as the grouping scheme [6], have been proposed to accelerate the computation process, it is still difficult to extend this method to *multi-lane* intersections with a large number of vehicles which may also change lanes along the way. In terms of optimizing control inputs, Model Predictive Control (MPC) is effective for problems with simple (usually linear or linearized) vehicle dynamics and constraints. However, when the vehicle dynamics are highly nonlinear and complex, the problem becomes a non-linear MPC whose computation time is still prohibitive for practical applications [7].

Another idea is to decompose the whole optimization problem into two separate problems, i.e., first determining

the crossing sequence and then solving for CAV control inputs according to this sequence [8]. The most straightforward crossing sequence mechanism follows the First-In-First-Out (FIFO) rule. For example, Dresner and Stone [9] proposed an autonomous intersection management cooperative driving strategy which divides the intersection into grids (resources) and assigns these grids to CAVs in a FIFO manner. In [10] and [11], a decentralized optimal control framework is designed for CAVs to jointly minimize energy and time by deriving the desired CAV arrival times at an intersection based on the FIFO crossing sequence. However, some recent studies have shown that the performance of the FIFO mechanism can lead to poor performance in some cases [12]. Thus, several studies have been conducted to search for better crossing sequences. For example, Guler *et al.* proposed to search all possible combinations of crossing sequences to find the optimal one [13]. However, this number increases exponentially with the number of vehicles. To reduce the computational time, Li *et al.* used a Genetic Algorithm (GA) to make decisions on the crossing sequence [14]. In [15], a Dynamic Resequencing (DR) scheme is designed to adjust the crossing sequence whenever a new CAV enters the intersection control zone and showed that this scheme is computationally efficient and improves traffic efficiency. Xu *et al.* [16] proposed a Monte Carlo Tree Search (MCTS)-based cooperative strategy to find a promising crossing sequence for all CAVs and demonstrated that this strategy may determine a good enough sequence even for complicated multi-lane intersections where the search space is enormous. It is worth noting that both of the above studies and our paper consider a 100% CAV environment. Such an environment does not usually need traffic lights, so that related work describes it as “signal-free” or “unsignalized”. However, [17] defines unsignalized intersections as those where at least one of the movements is controlled by a STOP or a YIELD sign. Thus, we refer to the scenarios considered here as signal-free intersections to avoid ambiguity. This paper will not discuss the optimization of crossing sequences or trajectories in mixed traffic environments consisting of both CAVs and conventional vehicles. Interested readers are referred to [18], [19].

After determining the crossing sequence, some studies [15], [16] use the analytical solutions proposed in [10] and [11] to solve for the optimal control inputs. Although [20] has extended this solution to consider speed-dependent rear-end safety constraints, the resulting computational cost significantly increases. Chalaki and Malikopoulos have also proposed a two-layer framework that considers lane-changing in the optimal control of CAVs crossing adjacent multi-lane intersections [21]. However, it is difficult to generalize optimal control methods for complex and nonlinear vehicle dynamics without incurring computational costs which make its real-time applicability prohibitive.

To address the above limitations, we propose an approach which combines the use of Control Barrier Functions (CBFs) with the conventional optimal control method to bridge the gap between optimal control solutions (which represent a lower bound for the optimal achievable cost) and controllers which can provably guarantee on-line safe execution [22].

The key idea is to design CAV trajectories which optimally track analytically tractable solutions of the basic intersection-crossing optimization problem while also provably guaranteeing that all safety constraints are satisfied. Through CBFs, we can map continuously differentiable state constraints into new control-based constraints. Due to the forward invariance of the associated safe set [23]–[25], a control input that satisfies these new constraints is also guaranteed to satisfy the original state constraints. This property makes the CBF method effective even when the vehicle dynamics and constraints become complicated and include noise.

Along these lines, the main contribution of this paper is a novel decentralized optimal control framework combining the optimal control and CBF methods for a *multi-lane* signal-free intersection. Specifically, we first formulate the multi-lane intersection problem as an optimal control problem whose objective is to jointly minimize the traffic delay and energy consumption while guaranteeing that all CAVs safely cross a four-way intersection that includes left and right turns. Unlike prior work, we replace roadway segments referred to as “merging zones (MZs)” or “conflict zones” by *Merging Points (MPs)* which are much less conservative while still guaranteeing collision avoidance. When we state that one method is more conservative than another, we mean that vehicles controlled by this method pass through the intersection with larger headways. Allowing lane-changing behavior, we design a strategy to determine the desirable locations of lane-changing MPs for all CAVs. Then, we develop a search algorithm to determine rear-end safety constraints and lateral safety constraints that every CAV has to meet. Once these constraints are specified for any CAV, we design an *Optimal Control and Barrier Function (OCBF)* controller for solving the problem efficiently, as verified through multiple simulation experiments. Our framework can accommodate a variety of resequencing methods for finding a near-optimal crossing sequence, including the aforementioned DR and MCTS schemes, leading to improved performance compared to the FIFO rule, especially when the intersection is geometrically asymmetrical.

The paper is organized as follows. *Section II* formulates the multi-lane intersection problem as an optimal control problem with safety constraints applied to a sequence of MPs. *Section III* introduces our search algorithm for determining all safety constraints pertaining to a given CAV, and *Section IV* presents our joint OCBF controller. *Section V* validates the effectiveness of the proposed method through simulation experiments. Finally, *Section VI* gives concluding remarks.

II. PROBLEM FORMULATION

Figure 1 shows a typical intersection with multiple lanes. The area within the outer red circle is called the *Control Zone (CZ)*, and the length of each CZ segment is L_1 which is initially assumed to be the same for all entry points to the intersection; extensions to asymmetrical intersections are straightforward and discussed in Section V.E. The significance of the CZ is that it allows all vehicles to share information and be automatically controlled while in it, as long as all

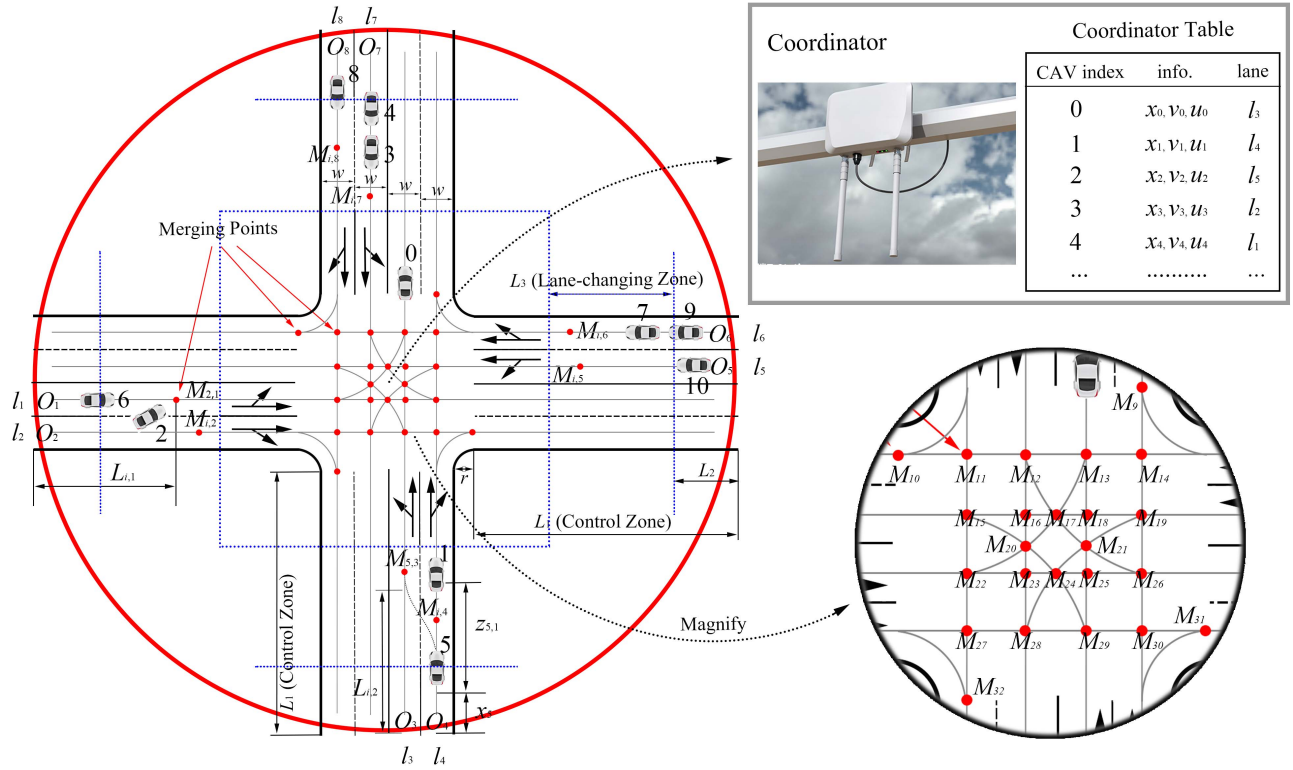


Fig. 1. The multi-lane intersection problem. Collisions may happen at the MPs (red dots shown in above figure).

vehicles in the CZ are assumed to be CAVs. Red dots show all MPs where potential collisions may occur. We assume that the motion trajectory of each CAV in the intersection is determined upon its entrance to the CZ (see grey lines in Fig. 1). Based on these trajectories, all MPs in the intersection are fixed and can be easily determined. However, unlike most prior similar studies, we also allow possible lane-changing behaviors in the CZ, which adds generality to our method. In order to avoid potential collisions with newly arriving vehicles and to conform with the driving rules that vehicles are prohibited from changing lanes when they are very close to the intersection, these lane changes are only allowed in a “lane-changing zone”, i.e., an area between the two blue lines shown in Fig. 1. We use dotted lines to differentiate these zones from the rest of the area where solid lines are shown. The distance from the entry of the CZ to the lane-changing zone is L_2 and the length of the lane-changing zone is L_3 . Since we initially consider a symmetrical intersection, L_2 and L_3 are the same for all lanes. However, it is easy to extend our method to asymmetrical intersections and set different parameters for each lane, as shown in Section V.E. Due to lane-changing, apart from the fixed MPs in the intersection, some “floating” MPs may also appear in lane-changing zones which are not fixed in space. Thus, there are two kinds of MPs: (i) the fixed MPs in the intersection and (ii) the floating MPs in the lane-changing zones.

We label the lanes from l_1 to l_8 in a counterclockwise direction with corresponding origins O_1 to O_8 . The rightmost lanes in each direction only allow turning right or going

straight, while the leftmost lanes only allow turning left or going straight. However, all CAVs have three possible movements: going straight, turning left, and turning right. Thus, some CAVs must change their lanes so as to execute a movement, e.g., left-turning CAV 2 in l_2 in Fig. 1. Due to such lane-changing behavior, a new MP $M_{2,1}$ is generated since a conflict of CAV 2 with a CAV in l_1 may arise. Similarly, possible MPs may also appear in other lanes when vehicles perform lane-changing maneuvers, as the red dots ($M_{i,2}, M_{i,3}, \dots$, and $M_{i,8}$) indicate in Fig. 1. Moreover, it is worth noting that if a CAV needs to change lanes, then it has to travel an additional (lateral) distance; we assume that this extra distance is a constant $l > 0$. In what follows, we consider an intersection that has two lanes in each direction, which, in our view, represents one of the most common intersection configurations worldwide, and observe that this model is easy to generalize to intersections with more than two lanes.

The intersection has a coordinator (typically a Road Side Unit (RSU)) whose function is to maintain the crossing sequence and all individual CAV information. The most common crossing sequence is based on the FIFO queue formed by all CAVs using their arrival time at the CZ, regardless of the lane each CAV belongs to. The FIFO queue is fair and simple to implement, however, its performance can occasionally be poor. Thus, various cooperative driving strategies have been proposed to generate a more promising crossing sequence, as in [15], [16], [26]. Our approach for controlling CAVs does not depend on the specific crossing sequence selected. Therefore, we first use the FIFO queue so as to enable accurate

comparisons with related work which also uses this scheme and then generalize it to include other resequencing methods which adjust the crossing sequence whenever there is a new arriving vehicle, e.g., the DR method in [15]. This allows CAVs to overtake other CAVs in the CZ from different roads, which better captures actual intersection traffic behaviors.

So far, we have introduced several new elements to prior intersection models, including: (i) replacing MZs with MPs which are less conservative since they reduce the spacing between CAVs; (ii) allowing lane-changing; (iii) allowing resequencing to improve the intersection throughput. In subsequent sections, the following additional features will be introduced: (iv) we use an extended coordinator table to identify pairs of conflicting CAVs at MPs, necessary to enforce safe merging constraints; (v) we provide safety guarantees through the use of CBFs; (vi) we allow noise and more complicated vehicle dynamics through the use of the OCBF controller. Specifically, the details of the solution framework is provided in Sections III (addressing (i)-(iv) above) and IV (addressing (v)-(vi) above).

A. Optimization Problem

We begin by reviewing the same basic model as in previous work [11], which will allow us to make accurate comparisons. When a CAV enters the CZ, the coordinator will assign it a unique index. Let $S(t)$ be the set of FIFO-ordered CAV indices and $N(t)$ be the cardinality of $S(t)$. Based on $S(t)$, the coordinator stores and maintains an information table, as shown in Fig. 1. For example, the current lane of CAV 2 changes from l_2 to l_1 after it completes a lane-changing maneuver. In addition, after CAV 0 passes the intersection, its index will be dropped from the table and the indices of all other CAVs decrease by one. This table enables each CAV to quickly identify other CAVs that have potential collisions with it and to optimize its trajectory to maximize some specific objectives. The search algorithm for identifying conflicting vehicles will be introduced in detail in the next section.

The vehicle dynamics for CAV i take the form

$$\begin{bmatrix} \dot{x}_i(t) \\ \dot{v}_i(t) \end{bmatrix} = \begin{bmatrix} v_i(t) + w_{i,1}(t) \\ u_i(t) + w_{i,2}(t) \end{bmatrix}, \quad (1)$$

where $x_i(t)$ is the distance to its origin along the lane that CAV i is located in when it enters the CZ, $v_i(t)$ denotes the velocity, and $u_i(t)$ denotes the control input (acceleration). Moreover, to compensate for possible measurement noise and modeling inaccuracy, we use $w_{i,1}(t)$ and $w_{i,2}(t)$ to denote two random processes defined in an appropriate probability space.

Based on the notation established above, we can now view trajectory planning of vehicles as an optimization problem where we consider two objectives for each CAV subject to three constraints, including the rear-end safety constraint with the preceding vehicle in the same lane, the lateral safety constraints with vehicles in the other lanes, and the vehicle physical constraints, as detailed next.

Objective 1 (Minimize Travel Time): Let t_i^0 and t_i^m denote the time that CAV i arrives at the origin and the time that it enters the intersection, respectively. To improve traffic efficiency, we wish to minimize the travel time $t_i^m - t_i^0$ for CAV i .

Objective 2 (Minimize Energy Consumption): Apart from traffic efficiency, another objective is energy efficiency. Ignoring any noise terms in (1) for the time being, since the energy consumption rate is a monotonic function of the acceleration control input, an energy consumption function we use is defined as

$$J_i = \int_{t_i^0}^{t_i^m} \mathcal{C}(u_i(t)) dt, \quad (2)$$

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

Constraint 1 (Rear-End Safety Constraint): Let i_p denote the index of the CAV which physically immediately precedes i in the CZ (if one is present). To avoid rear-end collisions, we require that the spacing $z_{i,i_p}(t) \equiv x_{i_p}(t) - x_i(t)$ be constrained by:

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

where δ is the minimum safety distance, and φ denotes the reaction time (as a rule, $\varphi = 1.8s$ is suggested, e.g., [27]). 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 is a constant determined by the length of these two CAVs (thus, δ is generally dependent on CAVs i and i_p but taken to be a constant over all CAVs in the sequel, only for simplicity). Note that i_p may change when a lane change event or an overtaking event (discussed in Section III.B) occurs.

Constraint 2 (Lateral Safety Constraint): Let t_i^k denote the time that CAV i arrives at the MP M_k , $k \in \{1, 2, \dots, 32\}$. CAV i may collide with other vehicles that travel through the same MP. For all MPs, including the floating MPs $M_{i,l}$ due to lane-changing, there must be enough safe space when CAV i is passing through, that is,

$$z_{i,j}(t_i^k) \geq \varphi v_i(t_i^k) + \delta, \quad (4)$$

where $j \neq i$ is a CAV that may collide with i (note that j may not exist and that there may also be multiple CAVs indexed by j for which this constraint applies at different t_i^k). The determination of j is challenging, and will be addressed in the following section. Compared with related work that requires no more than one CAV within a conflict (or merging) zone at any time, we use (4) to replace this conservative constraint. Instead of such a fixed zone, the space defining collision avoidance around each MP now depends on the CAV's speed (and possibly size if we allow δ to be CAV-dependent), hence it is much more flexible.

Constraint 3 (Vehicle Physical Limitations): Due to the physical limitations on motors and actuators, there are physical constraints on the velocity and control inputs for each CAV i :

$$\begin{aligned} v_{\min} &\leq v_i(t) \leq v_{\max}, \quad \forall t \in [t_i^0, t_i^m], \\ u_{i,\min} &\leq u_i(t) \leq u_{i,\max}, \quad \forall t \in [t_i^0, t_i^m], \end{aligned} \quad (5)$$

where $v_{\max} > 0$ and $v_{\min} \geq 0$ denote the maximum and minimum velocity allowed in the CZ, while $u_{i,\min} < 0$ and $u_{i,\max} > 0$ denote the minimum and maximum control input for each CAV i , respectively. We assume that all vehicles are homogeneous passenger cars and their minimum and maximum control inputs are the same. Thus, in what follows,

we use $u_{\min}$ and $u_{\max}$ instead of $u_{i,\min}$ and $u_{i,\max}$; this is done only for simplicity and does not limit our analysis.

Similar to prior work, we use a quadratic function for $\mathcal{C}(u_i(t))$ in (2) thus minimizing energy consumption by minimizing the control input effort $\frac{1}{2}u_i^2(t)$ [10], [28]. By normalizing travel time and $\frac{1}{2}u_i^2(t)$, and using $\alpha \in [0, 1)$, we construct a convex combination as follows:

$$\min_{u_i(t), t_i^m} \int_{t_i^0}^{t_i^m} \left(\alpha + \frac{(1-\alpha)\frac{1}{2}u_i^2(t)}{\frac{1}{2}\max\{u_{\max}^2, u_{\min}^2\}} \right) dt. \quad (6)$$

If $\alpha = 1$, problem (6) is equivalent to a minimum travel time problem; if $\alpha = 0$, it becomes a minimum energy consumption problem.

By defining $\beta \equiv \frac{\alpha \max\{u_{\max}^2, u_{\min}^2\}}{2(1-\alpha)}$ (assuming $\alpha < 1$) and multiplying (6) by the constant $\frac{\beta}{\alpha}$, we have:

$$\min_{u_i(t), t_i^m} \beta(t_i^m - t_i^0) + \int_{t_i^0}^{t_i^m} \frac{1}{2}u_i^2(t)dt, \quad (7)$$

where $\beta \geq 0$ is a weight factor that can be adjusted through $\alpha \in [0, 1)$ to penalize travel time relative to the energy cost. Then, the optimization problem can be stated as:

Problem 1: For each CAV i governed by dynamics (1) ignoring noise terms, determine a control law such that (7) is minimized subject to (1), (3), (4), (5), given t_i^0 and the initial and final conditions $x_i(t_i^0) = 0$, $v_i(t_i^0)$, $x_i(t_i^m)$.

III. MULTI-LANE INTERSECTION PROBLEM SOLUTION

The solution of *Problem 1* can be obtained as described in [20] where a single MP is involved in a two-road single-lane merging configuration where the value of j in (4) is immediately known. The difficulty here is that there may be more than one CAV j defining lateral safety constraints for any $i \in S(t)$ and determining the value(s) of j is challenging since there are eight lanes and three possible movements at intersections as shown in Fig. 1. Obviously, this can become even harder as more lanes are added or asymmetrical intersections are considered. Therefore, we propose a general MP-based approach which involves two steps. The first step addresses the following two issues: (i) a strategy for determining “floating” MPs due to CAVs possibly changing lanes within the CZ, and (ii) a strategy for determining all lateral safety constraints, hence the values of j in (4). Once these issues are addressed in the remainder of this section, *Problem 1* is well-defined. The second step consists of solving *Problem 1* and developing the proposed OCBF controller in the next section. The overall process is outlined in **Algorithm 1** which is implemented in time-driven manner by replanning the control inputs of all CAVs every T seconds.

A. Determination of Lane-Changing MPs

When a new CAV $i \in S(t)$ arrives at the origins O_2, O_4, O_6, O_8 (or O_1, O_3, O_5, O_7) and must turn left (or right), it has to change lanes before getting close to the intersection. Therefore, CAV i must determine the location of the variable (floating) MP $M_{i,k}$, $k \in \{1, 2, \dots, 8\}$.

Algorithm 1 MP-Based Algorithm for Multi-Lane Intersection Problems

- 1: Initialize an empty queue table $S(t)$.
- 2: **for** every T seconds **do**
- 3: **if** a new vehicle enters the CZ **then**
- 4: Determine a passing order for all CAVs according to the FIFO rule or other resequencing methods, e.g., the DR scheme.
- 5: Plan an *unconstrained* optimal control trajectory for the new CAV.
- 6: **if** the new CAV needs to change lanes **then**
- 7: Use the lane-changing MP determination strategy (Section III.A) to determine the lane-changing location and time for the new CAV
- 8: **end if**
- 9: Add the information of the new CAV into $S(t)$.
- 10: **end if**
- 11: **for** each CAV in $S(t)$ **do**
- 12: Use the lateral safety constraint determination strategy (Section III.B) to determine which constraints it needs to meet.
- 13: Use the OCBF controller (Section IV) to obtain control inputs for it.
- 14: **if** this CAV has left the intersection **then**
- 15: Remove the information of this CAV from $S(t)$.
- 16: **end if**
- 17: **end for**
- 18: **end for**

There are three important observations to make:

(i) The *unconstrained* optimal control for such i is independent of the location of $M_{i,k}$, $k \in \{1, 2, \dots, 8\}$ since we have assumed that lane-changing will only induce a fixed extra length l regardless of where it occurs.

(ii) The optimal control solution under the lateral safety constraint is better (i.e., lower cost in (7)) than one which includes an active rear-end safety constrained arc in its optimal trajectory. This is because the former applies only to a single time instant t_i^k whereas the latter requires the constraint (3) to be satisfied over all $t \in [t_i^0, t_i^k]$. It follows that any MP $M_{i,k}$ should be as close as possible to the intersection (i.e., $L_{i,k}$ should be as large as possible, and its maximum value is $L_2 + L_3$), since the lateral safety constraint after $M_{i,k}$ will become a rear-end safety constraint with respect to some j in the adjacent lane. For instance, suppose that CAV 10 in Fig. 1 is a right-turn vehicle. After changing lanes to the destination lane l_6 , it will be constrained by the rear-end safety constraint with its preceding vehicle (e.g., CAV 9), which will influence its optimal trajectory. Thus, to reduce the impact time, CAV 10 should change lanes as late as possible.

(iii) In addition, CAV i may also be constrained by its physically preceding CAV i_p (if one exists) in the same lane as i . In this case, CAV i needs to consider both the rear-end safety constraint with i_p and the lateral safety constraint with some $j \neq i$. Thus, the solution is more constrained (hence, more sub-optimal) if i stays in the current lane after the rear-

end safety constraint due to i_p becomes active. We conclude that in this case CAV i should change its lane to the left (right) lane as late as possible, i.e., just as the rear-end safety constraint with i_p first becomes active, i.e., $L_{i,k}$ is determined by

$$L_{i,k} = x_i^*(t_i^a) \quad (8)$$

where $x_i^*(t)$ denotes the unconstrained optimal trajectory of CAV i (as determined in Sec. IV), and $t_i^a \geq t_i^0$ is the time instant when the rear-end safety constraint between i and i_p first becomes active; if this constraint never becomes active, then $L_{i,k} = L_2 + L_3$. The value of t_i^a is determined from (3) by

$$x_{i_p}^*(t_i^a) - x_i^*(t_i^a) = \phi v_i^*(t_i^a) + \delta, \quad (9)$$

where $x_{i_p}^*(t)$ is the *unconstrained* optimal position of CAV i_p and $v_i^*(t)$ is the *unconstrained* optimal speed of CAV i . If, however, CAV i_p 's optimal trajectory itself happened to include a constrained arc, then (9) is only an upper bound of t_i^a .

In summary, it follows from (i) – (iii) above that if CAV i never encounters a point in its current lane where its rear-end safety constraint becomes active, we set $L_{i,k} = L_2 + L_3$, otherwise, $L_{i,k}$ is determined through (8)-(9).

We note that the distances from the origins $O_1, \dots, O_8$ to MPs are not all the same, and the CAVs that make a lane change will induce an extra l distance. Therefore, we need to perform a coordinate transformation for those CAVs that are in different lanes and will meet at the same MP M_k , $k \in \{1, \dots, 32\}$. In other words, when $i \in S(t)$ obtains information for $j \in S(t)$ from the FIFO queue table to account for the lateral safety constraint at an MP M_k , the position information $x_j(t)$ is transformed into $x'_j(t)$ through

$$x'_j(t) := \begin{cases} [c]rclx_j(t) + L_{i,k} - L_{j,k} + l, & \text{if only } i \text{ changes lane} \\ x_j(t) + L_{i,k} - L_{j,k} - l, & \text{if only } j \text{ changes lane,} \\ x_j(t) + L_{i,k} - L_{j,k}, & \text{otherwise.} \end{cases} \quad (10)$$

where $L_{i,k}$ and $L_{j,k}$ denote the distances of the MPs M_k from the origins of CAVs i and j , respectively. Note that the coordinate transformation (10) only applies to CAV i obtaining information on j from $S(t)$, and does not involve any action by the coordinator.

B. Determination of Lateral Safety Constraints

We begin with the observation (by simple inspection of Fig. 1) that CAVs can be classified into two categories, depending on the lane that a CAV arrives at, as follows:

- 1) The CAVs arriving at lanes l_1, l_3, l_5, l_7 will pass
 - two MPs if they choose to turn right (including the floating MP $M_{i,k}$, $k \in \{2, 4, 6, 8\}$);
 - four MPs if they turn left;
 - five MPs if they go straight.
- 2) The CAVs arriving at lanes l_2, l_4, l_6, l_8 will pass
 - only one MP if they choose to turn right;

$S(t)$		cur lane		MP							
index	info.	cur lane	ori lane	1 st MP	2 nd MP	3 rd MP	4 th MP	5 th MP	6 th MP	7 th MP	8 th MP
0	x_0, v_0, u_0	l_3	l_3	M_{29}	M_{23}	M_{21}	M_{18}	M_{13}			
1	x_1, v_1, u_1	l_4	l_4	M_{30}	M_{26}	M_{19}	M_{14}	M_9			
2	x_2, v_2, u_2	l_2	l_2	M_{21}	M_{22}	M_{20}	M_{17}	M_{13}			
3	x_3, v_3, u_3	l_7	l_7	M_{12}	M_{16}	M_{20}	M_{23}	M_{28}			
4	x_4, v_4, u_4	l_7	l_7	M_{12}	M_{16}	M_{20}	M_{23}	M_{28}			
5	x_5, v_5, u_5	l_4	l_4	M_{30}	M_{26}	M_{24}	M_{20}	M_{13}			
6	x_6, v_6, u_6	l_1	l_1	M_{22}	M_{20}	M_{17}	M_{13}				
7	x_7, v_7, u_7	l_6	l_6	M_9							
8	x_8, v_8, u_8	l_6	l_6	M_{10}							
9	x_9, v_9, u_9	l_6	l_6	M_{14}	M_{13}	M_{12}	M_{11}	M_{10}			
10	x_{10}, v_{10}, u_{10}	l_3	l_3	M_{19}	M_{18}	M_{17}	M_{16}	M_{15}			

Fig. 2. The extended coordinator queue table.

- five MPs if they turn left (including the floating MP $M_{i,k}$, $k \in \{1, 3, 5, 7\}$) or if they go straight.

Clearly, the maximum number of MPs a CAV may pass is 5. Since all such MPs are determined upon arrival at the CZ, we augment the queue table in Fig. 1 by adding the original lane and the MP information for each CAV as shown in Fig. 2 for a snapshot of Fig. 1. The current and original lanes are shown in the third and fourth column, respectively. The original lane is fixed, while the current lane may vary dynamically: in **Algorithm 1**, the state of all CAVs in the queue is updated if any of them has changed lanes. The remaining five columns show all MPs a CAV will pass through in order. For example, left-turning CAV 2 in Fig. 1 passes through five MPs $M_{2,1}$, M_{22} , M_{20} , M_{17} , and M_{13} sequentially, where we label the 1st MP as $M_{2,1}$ and so forth.

When a new CAV enters the CZ, it first determines whether it will change lanes (as described in Section III.A) and identifies all MPs that it must pass. At this point, it looks at the extended queue table $S(t)$ (an example is shown in Fig. 2) which already contains all prior CAV states and MP information. First, from the current lane column, CAV i can determine its current physically immediately preceding CAV i_p if one exists. Next, since the passing priority has been determined by the sequencing method selected (FIFO or otherwise), CAVs need to yield to other CAVs that rank higher in the queue $S(t)$. In addition, for any MP that CAV i will pass through, it only needs to yield to the closest CAV that has higher priority than it, and this priority is determined by the order of $S(t)$. For instance, CAVs 3, 4, and 5 will all pass through M_{20} , as shown in Fig. 2. For the MP M_{20} , CAV 5 only needs to meet the lateral safety constraint with CAV 4, while the constraint with CAV 3 will be automatically met since CAV 4 yields to CAV 3. Similarly, we can find indices of CAVs for other MPs crossed by CAV 5. Since CAV 5 passes through five MPs, we define an index set Ω_5 for CAV 5 which has at most 5 elements. In this example, CAV 5 only conflicts with CAV 0 at M_{29} besides M_{20} , so that $\Omega_5 = \{0, 4\}$.

Therefore, it remains to use the information in $S(t)$ in a systematic way so as to determine all the indices of those CAVs that CAV i needs to yield to; these define each index j in (4) constituting all lateral safety constraints that CAV i needs to satisfy. This is accomplished by a search

Algorithm 2 Search Algorithm for Conflict CAVs**Require:** The extended coordinator queue table $S(t)$, CAV i **Ensure:** The index of conflict CAVs for CAV i

```

1: Initialize an empty index set  $\Omega_i$ .
2: Initialize a set  $\Theta_i$  including all MPs CAV  $i$  will pass through
3: Find the position  $k$  of CAV  $i$  in the  $S(t)$ .
4: for  $j = k - 1 : -1 : 1$  do
5:   if the  $j$ th CAV in  $S(t)$  passes at least one MP in  $\Theta_i$  then
6:     Add the index of this CAV into the set  $\Omega_i$ .
7:     Remove the same MPs from  $\Theta_i$ .
8:   end if
9:   if  $\Theta_i$  is empty then
10:    break;
11:   end if
12: end for
13: return  $\Omega_i$ 

```

algorithm (**Algorithm 2**) based on the following process. CAV i compares its original lane and MP information to that of every CAV in the table starting with the last row of itself and moving up. The process terminates the first time that any one of the following three conditions is satisfied at some row $j < i$:

- 1) All MP information of CAV i matches row j and Ω_i is empty.
- 2) Every MP for CAV i has been matched to some row j .
- 3) All prior rows $j < i$ have been looked up.

These three conditions are examined in order:

Condition (1): If this is satisfied, there are no conflicting MPs for CAV i and this implies that CAV i_p is the physically immediately preceding CAV all the way through the CZ. Thus, CAV i only has to satisfy the safety constraint (3) with respect to i_p , i.e., it just follows CAV i_p . For example, $i = 4$ and $i_p = 3$ in Fig. 1 (and Fig. 2).

Condition (2): If this holds, then CAV i has to satisfy several lateral safety constraints with one or more CAV $j \in \Omega_i$. Moreover, it also has to satisfy the rear-end safety constraint (3) with CAV i_p , where i_p is determined by the first matched row in the current lane column of Fig. 2. For example, $i = 10$, $j = 0, 1, 4, 5$, and 6 in Fig. 1 (and Fig. 2).

Condition (3): There are two cases. First, if the index set Ω_i is empty, then CAV i does not have to satisfy any lateral safety constraint; for example, $i = 7$ in Fig. 1 (and Fig. 2). Otherwise, it needs to yield to all CAVs in Ω_i ; for example, $i = 2$, $j = 0$ in Fig. 1 (and Fig. 2).

We observe that **Algorithm 2** can be implemented for all CAVs in an event-driven way (since $S(t)$ needs to be updated only when an event that changes its state occurs). The triggering events are: (i) a CAV entering the CZ, (ii) a CAV departing the CZ, (iii) a CAV completing a lane change at a floating MP, and (iv) a CAV overtaking event (a lane change event at a fixed MP). This last event may occur when a CAV merges into another lane at an MP through which it leaves the CZ. In particular, consider three CAVs i, j , and k such that $k > j > i$, and CAV j merges into the same

lane as i and k . Then, CAV k looks at the first row above it where there is a CAV with the same lane; that's now CAV j . However, i is physically ahead of k . Thus, we need to re-order the queue according to the incremental position order, so that a CAV following i (CAV k) can properly identify its physically preceding CAV. For example, consider $i = 7$, $j = 8$, and $k = 9$ in Fig. 1, and suppose that CAV 7 turns right, CAV 8 turns right, and CAV 9 goes straight. Their order in $S(t)$ is 7, 8, 9. CAV 8 can overtake CAV 7, and its current lane will become l_6 when it passes all MPs. Since CAV 7 and CAV 9 also are in l_6 , CAV 9 will mistake CAV 8 as its new preceding CAV after the current lane of CAV 8 is updated. However, in reality CAV 7 is still the preceding CAV of CAV 9, hence CAV 9 may accidentally collide with CAV 7. To avoid this problem, we need to re-order the queue according to the position information when this event occurs, i.e., making CAV 8 have higher priority than CAV 7 in the queue. Alternatively, this problem may be resolved by simply neglecting CAVs that have passed all MPs when searching for the correct identity of the CAV that precedes i .

IV. JOINT OPTIMAL AND CONTROL BARRIER FUNCTION CONTROLLER

Once a newly arriving CAV $i \in S(t)$ has determined all the lateral safety constraints (4) it has to satisfy, it can solve problem (7) subject to these constraints along with the rear-end safety constraint (3) and the state limitations (5). Obtaining a solution to this constrained optimal control problem is computationally intensive, as shown in the single-lane merging problem [20], and this complexity is obviously higher for our multi-lane intersection problem since there are more lateral safety constraints. Therefore, in this section, we proceed in two steps: (i) We solve the *unconstrained* version (only includes initial and final conditions, and terminal time is free) of problem (7); this can be done with minimal computational effort, and (ii) We optimally track the unconstrained problem solution while using CBFs to account for all constraints and guarantee that they are never violated (as well as Control Lyapunov Functions (CLFs) to better track the unconstrained optimal states). Since this controller $u_i(t)$ for CAV i combines an optimal control solution with CBFs, we refer to it as the OCBF control. Note that each CAV can solve *Problem 1* in a decentralized way.

A. Unconstrained Decentralized Optimal Control Solution

As mentioned above, we use the CAV trajectory obtained from the unconstrained optimal solution *Problem 1* as a *reference* trajectory and deal with all constraints through our OCBF controller. When all state and safety constraints are inactive, we can obtain an analytical solution of *Problem 1* which, as shown in [15], provides optimal control, speed, and position trajectories given by

$$u_i^*(t) = a_i t + b_i \quad (11)$$

$$v_i^*(t) = \frac{1}{2} a_i t^2 + b_i t + c_i \quad (12)$$

$$x_i^*(t) = \frac{1}{6} a_i t^3 + \frac{1}{2} b_i t^2 + c_i t + d_i \quad (13)$$

where a_i , b_i , c_i and d_i are integration constants that can be solved along with t_i^m by the following five algebraic equations (details given in [15]):

$$\begin{aligned} \frac{1}{2}a_i \cdot (t_i^0)^2 + b_i t_i^0 + c_i &= v_i^0, \\ \frac{1}{6}a_i \cdot (t_i^0)^3 + \frac{1}{2}b_i \cdot (t_i^0)^2 + c_i t_i^0 + d_i &= 0, \\ \frac{1}{6}a_i \cdot (t_i^m)^3 + \frac{1}{2}b_i \cdot (t_i^m)^2 + c_i t_i^m + d_i &= L_k, \\ a_i t_i^m + b_i &= 0, \\ \beta - \frac{1}{2}b_i^2 + a_i c_i &= 0. \end{aligned} \quad (14)$$

where the third equation is the terminal condition for the total distance traveled on a lane. This solution is computationally very efficient to obtain. We then use this unconstrained optimal control solution as a reference to be tracked by a controller which uses CBFs to account for all the constraints (3), (4), and (5) and guarantee they are not violated. We review next how to use CBFs to map all these constraints from the state $x_i(t)$ to the control input $u_i(t)$.

B. OCBF Controller

The OCBF controller aims to track the unconstrained optimal control solution (11)-(13) while satisfying all constraints (3), (4) and (5). To accomplish this, first let $\mathbf{x}_i(t) \equiv (x_i(t), v_i(t))$. Referring to the vehicle dynamics (1), let $f(\mathbf{x}_i(t)) = [x_i(t), 0]^T$ and $g(\mathbf{x}_i(t)) = [0, 1]^T$. Each of the constraints in (3), (4) and (5) can be expressed as $b_k(\mathbf{x}_i(t)) \geq 0$, $k \in \{1, \dots, n\}$ where n is the number of constraints and each $b_k(\mathbf{x}_i(t))$ is a CBF. For example, $b_1(\mathbf{x}_i(t)) = z_{i,i_p}(t) - \varphi v_i(t) - \delta$ for the rear-end safety constraint (3). In the CBF approach, each of the continuously differentiable state constraints $b_k(\mathbf{x}_i(t)) \geq 0$ is mapped onto another constraint on the control input such that the satisfaction of this new constraint implies the satisfaction of the original constraint $b_k(\mathbf{x}_i(t)) \geq 0$. The forward invariance property of this method [24], [25] ensures that a control input that satisfies the new constraint is guaranteed to also satisfy the original one. In particular, each of these new constraints takes the form

$$L_f b_k(\mathbf{x}_i(t)) + L_g b_k(\mathbf{x}_i(t))u_i(t) + \gamma(b_k(\mathbf{x}_i(t))) \geq 0, \quad (15)$$

where L_f and L_g denote the Lie derivatives of $b_k(\mathbf{x}_i(t))$ along f and g (defined above from the vehicle dynamics) respectively and $\gamma(\cdot)$ denotes a class of $\mathcal{K}$ functions [29] (typically, linear or quadratic functions). As an alternative, a CLF [24] $V(\mathbf{x}_i(t))$, instead of $b_k(\mathbf{x}_i(t))$, can also be used to track (stabilize) the optimal speed trajectory (13) through a CLF constraint of the form

$$L_f V(\mathbf{x}_i(t)) + L_g V(\mathbf{x}_i(t))u_i(t) + \epsilon V(\mathbf{x}_i(t)) \leq e_i(t), \quad (16)$$

where $\epsilon > 0$ and $e_i(t)$ is a relaxation variable that makes this constraint soft. As is usually the case, we select $V(\mathbf{x}_i(t)) = (v_i(t) - v_{ref}(t))^2$ where $v_{ref}(t)$ is the reference speed to be tracked (specified below). Therefore, the OCBF controller

solves the following problem:

$$\begin{aligned} \min_{u_i(t), e_i(t)} & J_i(u_i(t), e_i(t)) \\ &= \int_{t_i^0}^{t_i^m} \left(\beta e_i^2(t) + \frac{1}{2}(u_i(t) - u_{ref}(t))^2 \right) dt, \end{aligned} \quad (17)$$

subject to the vehicle dynamics (1), the CBF constraints (15) and the CLF constraint (16). The obvious selection for speed and acceleration reference signals is $v_{ref}(t) = v_i^*(t)$, $u_{ref}(t) = u_i^*(t)$ with $v_i^*(t)$, $u_i^*(t)$ given by (13) and (11) respectively. In [30], $u_{ref}(t) = \frac{x_i^*(t)}{x_i(t)} u_i^*(t)$ is used to provide the benefit of feedback obtained by observing the actual CAV trajectory $x_i(t)$ and automatically reducing the tracking position error; we use only $u_{ref}(t) = u_i^*(t)$ in the sequel for simplicity.

The CBF conversions from the original constraints to the form (15) are straightforward. For example, using a linear function $\gamma(\cdot)$ in (15), we can directly map **Constraint 1** onto the following constraint in terms of control inputs:

$$\underbrace{v_{i_p}(t) - v_i(t)}_{L_f b(x_i(t))} + \underbrace{-\varphi}_{L_g b(x_i(t))} u_i(t) + z_{i,i_p}(t) - \varphi v_i(t) - \delta \geq 0. \quad (18)$$

However, there are some points that deserve some further clarification as follows.

Constraint 2 (Lateral Safety Constraint): The lateral safety constraints in (4) are specified only at time instants t_i^k . However, to use CBFs as in (15), they have to be converted to *continuously differentiable forms*. Thus, we use the same technique as in [31] to convert (4) into:

$$z_{i,j}(t) \geq \Phi(x_i(t))v_i(t) + \delta, \quad i \in S(t), \quad t \in [t_i^0, t_i^k], \quad (19)$$

where $j \in \Omega_i$ is determined through the lateral safety constraint determination strategy (**Algorithm 2**). Recall that CAV j depends on some MP M_k and we may have several $j \in \Omega_i$ since CAV i may conflict with several CAVs j at different MPs. The selection of function $\Phi: \mathbb{R} \rightarrow \mathbb{R}$ is flexible as long as it is a strictly increasing function that satisfies $\Phi(x_i(t_i^0)) = 0$ and $\Phi(x_i(t_i^k)) = \varphi$ where t_i^k is the arrival time at MP M_k corresponding to the constraint and $x_i(t_i^k)$ is the location of MP M_k . Thus, we see that at $t = t_i^k$ all constraints in (19) match the safe-merging constraints (4), and that at $t = t_i^0$ we have $z_{i,i_p}(t_i^0) = \delta$. Since the selection of $\Phi(\cdot)$ is flexible, for simplicity, we define it to have the linear form $\Phi(x_i(t)) = \frac{\varphi}{x_i(t_i^k)} x_i(t)$ which we can immediately see satisfies the properties above.

Improving the Feasibility of Constraints 1 and 2: In order to ensure that a feasible solution always exists for these constraints, we need to take the braking distance into consideration. CAV i should stop within a minimal safe distance when its speed $v_i(t)$ approaches the speed $v_j(t)$ for any j such that CAV j is the preceding vehicle of CAV i or any vehicles that may laterally collide with CAV i . Thus, we use the following more strict constraint when $v_i(t) \geq v_j(t)$:

$$z_{i,j}(t) \geq \frac{\varphi(x_i(t) + \frac{1}{2} \frac{v_i^2(t) - v_j^2(t)}{u_{\min}})v_j(t)}{L} + \frac{1}{2} \frac{(v_j(t) - v_i(t))^2}{u_{\min}} + \delta, \quad (20)$$

A detailed analysis of this constraint is given in [31].

Observing that **Constraint 3** (vehicle limitations) can be directly converted using the standard CBF method, we are now in a position where all constraints are mapped onto constraints expressed in terms of control inputs. We refer to the resulting $u_i(t)$ in (17) as the OCBF control. The solution to (17) is obtained by discretizing the time interval $[t_i^0, t_i^m]$ with time steps of length Δ and solving (17) over $[t_i^0 + k\Delta, t_i^0 + (k+1)\Delta]$, $k = 0, 1, \dots$, with $u_i(t)$, $e_i(t)$ as decision variables held constant over each such interval (see also [31]). Consequently, each such problem is a Quadratic Program (QP) since we have a quadratic cost and a number of linear constraints on the decision variables at the beginning of each time interval. The solution of each such problem gives an optimal control $u_i^*(t_i^0 + k\Delta)$, $k = 0, 1, \dots$, allowing us to update (1) in the k^{th} time interval. This process is repeated until CAV i leaves the CZ.

C. The Influence of Noise and Complicated Vehicle Dynamics

Aside from the potentially long computation time, other limitations of the OC controller include: (i) It only plans the optimal trajectory once. However, the trajectory may violate safety constraints due to noise in the vehicle dynamics and control accuracy; (ii) The OC analytical solution is limited to simple vehicle dynamics as in (1) and becomes difficult to obtain when more complicated vehicle dynamics are considered to better match realistic operating conditions. For instance, in practice, we usually need to control the input driving force of an engine instead of directly controlling acceleration. Compared with the OC method, our OCBF approach can effectively deal with the above problems with only slight modifications as described next.

First, due to the presence of noise, constraints may be temporarily violated, which prevents the CBF method from satisfying the forward invariance property. Thus, when a constraint is violated at time t_1 , i.e., $b_k(x_i(t_1)) < 0$, we add a threshold to the original constraint as follows:

$$L_f b_k(x_i(t)) + L_g b_k(x_i(t))u_i(t) \geq c_k(t), \quad (21)$$

where $c_k(t) \geq 0$ is a large enough value so that $b_k(x_i(t))$ is strictly increasing even if the system is under the worst possible noise case. Since it is hard to directly determine the value of $c_k(t)$, we add it to the objective function and have

$$\min_{u_i(t), e_i(t), c_k(t)} \int_{t_i^0}^{t_i^m} \left(\beta e_i^2(t) + \frac{1}{2}(u_i(t) - u_{ref}(t))^2 - \eta c_k(t) \right) dt, \quad (22)$$

where η is a weight parameter. If there are multiple constraints that are violated at one time, we rewrite them all as (21) and add all thresholds into the optimization objective. Starting from t_1 , we use the constraint (21) and objective function (22) to replace the original CBF constraint and objective function, and $b_k(x_i(t))$ will be positive again in finite time since it is increasing. When $b_k(x_i(t))$ becomes positive again, we revert to the original CBF constraint.

Next, considering vehicle dynamics, there are numerous models which achieve greater accuracy than the simple model (1) depending on the situation of interest. As an

example, we consider the following frequently used nonlinear model:

$$\begin{bmatrix} \dot{x}_i(t) \\ \dot{v}_i(t) \end{bmatrix} = \begin{bmatrix} v_i(t) \\ -\frac{1}{m_i} F_r(v_i(t)) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m_i} \end{bmatrix} u_i(t), \quad (23)$$

where m_i denotes the vehicle mass and $F_r(v_i(t))$ is the resistance force that is normally expressed as

$$F_r(v_i(t)) = \alpha_0 \text{sgn}(v_i(t)) + \alpha_1 v_i(t) + \alpha_2 v_i^2(t), \quad (24)$$

where $\alpha_0 > 0$, $\alpha_1 > 0$, and $\alpha_2 > 0$ are parameters determined empirically, and $\text{sgn}(\cdot)$ is the signum function. It is clear that due to the nonlinearity in these vehicle dynamics, it is unrealistic to expect an analytical solution for it. However, in our proposed OCBF method, we only need to derive the Lie derivative along these new dynamics and solve the corresponding QP based on these new CBF constraints. For instance, it is easy to see that for these new dynamics, the CBF constraint (18) becomes

$$\underbrace{v_{i,p}(t) - v_i(t) + \frac{\phi F_r(v_i(t))}{m_i}}_{L_f b(x_i(t))} + \underbrace{-\frac{\phi}{m_i} u_i(t)}_{L_g b(x_i(t))} + z_{i,i_p}(t) - \phi v_i(t) - \delta \geq 0. \quad (25)$$

Thus, our method can be easily extended to more complicated vehicle dynamics dictated by any application of interest.

V. SIMULATION RESULTS

To validate the effectiveness of the proposed OCBF method, we compare it to a state-of-the-art method in [11] where CAVs calculate the fastest arrival time to the conflict zone first when they enter the CZ and then derive an energy-time-optimal trajectory. This uses the same objective function (7) as our method. The main differences are: 1) it considers the merging (conflict) zone as a whole and imposes the conservative requirement that any two vehicles that have potential conflict cannot be in the conflict zone at the same time; 2) when it plans an energy-time-optimal trajectory for a new incoming vehicle, it takes all safety constraints into account, which makes it time-consuming; 3) the rear-end safety constraints used in [11] only depend on distance, i.e., $\phi = 0$ and $\delta > 0$ in (3). Thus, in order to carry out a fair comparison with this method, we adopt the same form of rear-end safety constraints, that is,

$$z_{i,i_p}(t) \geq \delta, \quad \forall t \in [t_i^0, t_i^m]. \quad (26)$$

A complication caused by this choice is that after using the standard CBF method (simply substituting $\phi = 0$ into (18)), the control input should satisfy

$$\underbrace{v_{i,p}(t) - v_i(t)}_{L_f b(x_i(t))} + \underbrace{0}_{L_g b(x_i(t))} u_i(t) + z_{i,i_p}(t) - \delta \geq 0, \quad (27)$$

which violates the condition $L_g b(x_i(t)) \neq 0$. This is because we cannot obtain a relationship involving the control input $u_i(t)$ from the first-order derivative of the constraint (26). This problem was overcome in [25] by using a high order CBF

(HOCBF) of relative degree 2 for system (1). In particular, letting $b_k(x_i(t)) = z_{i,i_p}(t) - \delta$ and considering all class $\mathcal{K}$ functions to be linear functions, we define

$$\begin{aligned}\psi_1(x_i(t)) &= \dot{b}(x_i(t)) + pb(x_i(t)), \\ \psi_2(x_i(t)) &= \dot{\psi}_1(x_i(t)) + p\psi_1(x_i(t)).\end{aligned}\quad (28)$$

where p is a (tunable) penalty coefficient. Combining the vehicle dynamics (1) with (28), any control input should satisfy

$$\underbrace{0}_{L_f^2 b(x_i(t))} + \underbrace{-1}_{L_g L_f b(x_i(t))} u_i(t) + 2p\dot{b}(x_i(t)) + p^2 b(x_i(t)) \geq 0. \quad (29)$$

Thus, in the following simulation experiments, we set φ for **Constraint 1** and **Constraint 2** to be $\varphi = 0$ and $\varphi = 1.8s$, respectively.

Our simulation experiments are organized as follows. First, in Section V.A we consider an intersection with a single lane in each direction which only allows a CAV to go straight. Our purpose here is to show that using MPs instead of an entire arbitrarily defined conflict zone can effectively reduce the conservatism of the latter. Then, in Section V.B, we allow turns so as to analyze the influence of different behaviors (going straight, turning left, and turning right) on the performance of the methods compared. Next, Section V.C is intended to validate the effectiveness of our OCBF method for intersections with two lanes and include possible lane-changing behaviors. Section V.D shows simulations that quantify how our proposed method outperforms the signal-based method in all metrics. In Section V.E, we extend our method to combine it with the DR method and show that the performance of the OCBF+DR method is better than the OCBF+FIFO method for asymmetrical intersections. Finally, Section V.F demonstrates that our method can effectively deal with complicated vehicle dynamics and noise.

The baseline for our simulation results uses SUMO, a microscopic traffic simulation software package. In SUMO simulations, we set most models and their parameters at their default values. For example, the car-following model used is the default Krauss car-following model, and its parameters are also set at their default values. Interested readers can refer to [32] for more details. Then, we use our OCBF controller and the controller proposed in [11] to control CAVs for intersection scenarios with the same vehicle arrival patterns as SUMO. The parameter settings (see Fig. 1) are as follows: $L_1 = 300m$, $L_2 = 50m$, $L_3 = 200m$, $l = 0.9378m$, $w = 3.5m$, $r = 4m$, $\delta = 10m$, $v_{\max} = 15m/s$, $v_{\min} = 0m/s$, $u_{\max} = 3m/s^2$, and $u_{\min} = -3m/s^2$.

The energy model we use in the objective function is an approximate one. The $\frac{1}{2}u^2$ metric treats acceleration and deceleration the same and does not account for speed as contributing to energy consumption. This metric is viewed as a simple surrogate function for energy or simply as a measure of how much the solution deviates from the ideal constant-speed trajectory. In contrast, the following energy model [33] captures fuel consumption in detail and provides

TABLE I
THE COMPARISON RESULTS FOR A SINGLE-LANE
INTERSECTION DISALLOWING TURNS

β	Methods	Energy	Travel time (s)	Fuel (mL)	Ave. obj. ¹
0.1	SUMO	23.1788	28.3810	30.3597	26.0169
	OC	0.1498	28.3884	14.6266	2.9886
	OCBF	0.9501	25.0863	18.3088	3.4587
0.5	SUMO	23.1788	28.3810	30.3597	37.3693
	OC	0.6515	26.0315	17.1585	13.6673
	OCBF	2.1708	22.6623	18.9396	13.5020
1	SUMO	23.1788	28.3810	30.3597	51.5598
	OC	0.8782	25.6961	17.1555	26.5743
	OCBF	2.9106	22.2617	18.9589	25.1723
2	SUMO	23.1788	28.3810	30.3597	79.9408
	OC	1.1869	25.4834	17.1658	52.1537
	OCBF	3.9157	21.9139	18.9852	47.7435

¹ Ave. obj. = $\beta \times \text{Travel time} + \text{Energy}$.

another measure of performance:

$$\begin{aligned}F_i &= \int_0^{a_i} f_{V,i}(t) dt, \\ f_{V,i}(t) &= f_{cruise,i}(t) + f_{accel,i}(t), \\ f_{cruise,i}(t) &= b_0 + b_1 v_i(t) + b_2 v_i^2(t) + b_3 v_i^3(t), \\ f_{accel,i}(t) &= u(t)(c_0 + c_1 v_i(t) + c_2 v_i^2(t)),\end{aligned}\quad (30)$$

where $f_{cruise,i}(t)$ denotes the fuel consumed per second when CAV i drives at a steady velocity $v_i(t)$, and $f_{accel,i}(t)$ is the additional fuel consumed due to the presence of positive acceleration. If $u(t) \leq 0$, then $f_{accel,i}(t)$ will be 0 since, in this case, the engine is rotated by the kinetic energy of the CAV. $b_0, b_1, b_2, b_3, c_0, c_1$, and c_2 are seven model parameters; here we use the same parameters as in [33], which are obtained through curve-fitting for data from a typical vehicle.

A. MPs Versus Conflict Zone

In this experiment, we only allow CAVs to go straight in order to investigate the relative performance of the MP-based method (our OCBF controller) and the merging (conflict) zone-based method (OC controller) [11]. We set the arrival rates at all lanes to be the same, i.e., $270veh/h/lane$. The comparison results are shown in Table I.

It is clear that both controllers significantly outperform the results obtained from the SUMO car-following controller. The OC controller is energy-optimal since it has considered all safety constraints for each CAV upon its arrival at the CZ. Subsequently, CAVs strictly follow the planned trajectory assuming the absence of noise. However, our OCBF controller only uses an *unconstrained* reference trajectory and employs CBFs to account for the fact that this reference trajectory may violate the safety constraints: for each CAV, this controller continuously updates its control inputs according to the latest states of other CAVs. As a result, its energy consumption is larger than that of the OC controller, although still small and much lower than the one evaluated under the SUMO car-following controller.

In terms of travel time, we find that the travel time of the OCBF controller is better than that of the OC controller.

TABLE II
THE INFLUENCE OF TURNS ON DIFFERENT CONTROLLERS

Groups	Methods	Energy	Travel time	Fuel	Ave. obj.
1	SUMO	23.7011	28.2503	28.3357	51.9514
	OC	1.2957	22.4667	18.8433	23.7624
	OCBF	2.6193	21.8514	18.9208	24.4707
2	SUMO	26.5612	31.3822	29.0524	57.9434
	OC	1.0666	24.2179	18.0072	25.2845
	OCBF	2.4537	21.8707	18.8704	24.3244
3	SUMO	19.9066	24.2937	25.6778	44.2003
	OC	1.3775	22.0706	18.8803	23.4481
	OCBF	2.3623	21.4874	18.8306	23.8497
4	SUMO	21.7450	26.6148	29.4484	48.3598
	OC	1.2602	22.7417	18.7339	24.0019
	OCBF	2.5884	22.0114	18.9230	24.5998

This is because the safety requirements in the OC controller are too strict because CAV i must wait until a CAV $j \neq i$ that conflicts with it leaves the conflict zone. Instead, the OCBF controller using MPs allows us to relax a merging constraint and still ensure safety by requiring that only one CAV can arrive at the same MP ϕ a short time after the other vehicle leaves. Since our method reduces conservatism, it shows significant improvement in travel time when compared with the OC controller in [11].

In addition, we can adjust the parameter β to emphasize the relative importance of one objective (energy or time) relative to the other. If we are more concerned about energy consumption, we can use a smaller value of β ; otherwise, a larger β value emphasizes travel time reduction. Thus, when β is relatively large, the average objective under the OCBF controller is better than that of the OC controller since it is more efficient with respect to travel time.

Another interesting observation is that even though the relationship between the accurate fuel consumption model and the estimated energy is complicated, we see that a larger estimated energy consumption usually corresponds to larger fuel consumption. Thus, it is reasonable to optimize energy consumption through a simple model, e.g., $\frac{1}{2}u^2$, which also significantly reduces the computational complexity caused by the accurate energy model.

B. The Influence of Turns

In this experiment, we allow turns at the intersection assuming that the intention (i.e., going straight, turning left, and turning right) of a CAV when it enters the CZ is known. We have conducted four groups of simulations as shown in Table II. In the first group, all CAVs choose their behavior with the same probability, i.e., $\frac{1}{3}$ going straight, $\frac{1}{3}$ turning left, and $\frac{1}{3}$ turning right. In the second group, 80% of CAVs turn left while 10% CAVs go straight and 10% CAVs turn right. In the third group, 80% CAVs turn right while 10% CAVs go straight and 10% CAVs turn left. In the fourth group, 80% CAVs go straight while 10% CAVs turn left and 10% CAVs turn right. We set $\beta = 1$ in all results shown in Table II.

First, we can draw the same conclusion as in Table I that the OC controller is energy-optimal and the OCBF

TABLE III
COMPARISON RESULTS FOR A TWO-LANE INTERSECTION

β	Methods	Energy	Travel time (s)	Fuel (mL)	Ave. obj.
0.1	SUMO	24.0124	29.5955	30.5588	26.9720
	OC	0.1514	28.5711	14.6685	3.0085
	OCBF	1.0350	25.0804	18.5086	3.5430
0.5	SUMO	24.0124	29.5955	30.5588	38.8102
	OC	0.6722	26.1284	17.2866	13.7364
	OCBF	2.2244	22.6351	19.0753	13.5420
1	SUMO	24.0124	29.5955	30.5588	53.6079
	OC	0.9063	25.8000	17.2933	26.7063
	OCBF	2.9955	22.2347	19.1126	25.2302
2	SUMO	24.0124	29.5955	30.5588	83.2034
	OC	1.2294	25.5773	17.3042	52.3840
	OCBF	4.2353	22.1167	19.1500	48.4687

controller achieves the lowest travel time since it reduces conservatism. Next, we also observe that when we increase the ratio of left-turning vehicles, the average travel times under all controllers increase; when we increase the ratio of right-turning vehicles, the average travel times all decrease. This demonstrates that the left-turning behavior usually has the largest impact on traffic coordination since left-turning CAVs cross the conflict zone diagonally and are more likely to conflict with other CAVs. In addition, it is worth noting that going straight produces the largest travel time since this involves the largest number of MPs. However, when we use the OCBF controller, the travel times under all situations are similar, which shows that this controller can utilize the space resources of the conflict zone and handle the influence of turns more effectively.

C. Comparison Results for More Complicated Intersections

In this experiment, we consider more complicated intersections with two lanes in each direction as shown in Fig. 1. The left lane in each direction only allows going straight and turning left, while the right lane only allows going straight and turning right. We set the arrival rate at all lanes to be the same, i.e., $180veh/h/lane$ and disallow lane-changing. Each new incoming CAV chooses its behavior from the allowable behaviors with the same probability, e.g., the CAV arriving at the entry of the left lane can go straight or turn left with probability 0.5. The comparison results are shown in Table III.

The results here are similar to those in the single-lane intersections. Although the number of MPs increases with the number of lanes, our method can still effectively ensure safety and reduce travel time. It is worth noting that the values of safety time headway for the OC controller are difficult to determine. The OC controller requires that no CAV can enter the conflict zone until the conflict CAV leaves it. However, the time spent for passing through the conflict zone differs from vehicle to vehicle. If we choose a larger value, then this significantly increases travel time and amplifies conservatism. In contrast, if we choose a smaller value, the potential of collision increases. Therefore, the MP-based method is significantly better since it not only ensures safety but also reduces conservatism.

TABLE IV
THE INFLUENCE OF LANE-CHANGING BEHAVIOUR
ON THE PROPOSED METHOD

β	Methods	Energy	Travel time	Fuel	Ave. obj.
0.1	SUMO with LC	23.9988	30.0337	30.0000	27.0022
	OCBF w/o LC	1.0350	25.0804	18.5086	3.5430
	OCBF with LC	1.0738	25.1200	18.5474	3.5858
0.5	SUMO with LC	23.9988	30.0337	30.0000	39.0157
	OCBF w/o LC	2.2244	22.6351	19.0753	13.5420
	OCBF with LC	2.2584	22.6689	19.1148	13.5929
1	SUMO with LC	23.9988	30.0337	30.0000	54.0325
	OCBF w/o LC	2.9955	22.2347	19.1126	25.2302
	OCBF with LC	3.0282	22.2684	19.1575	25.2966
2	SUMO with LC	23.9988	30.0337	30.0000	84.0662
	OCBF w/o LC	4.2353	22.1167	19.1500	48.4687
	OCBF with LC	4.2887	22.1536	19.2457	48.5959

In what follows, we briefly discuss the computation time involved in our OCBF method. The method is driven by solving a QP problem using the latest information from all CAVs over each control period. The average computation time for solving such a QP is 3.5ms (Intel(R) Core(TM) i7-6700 CPU), which is suitable for real-time implementation. In contrast, the OC-based controller plans a trajectory for each vehicle when it enters the control zone, and the average computation time for such a plan is about 1s (Intel(R) Core(TM) i7-6700 CPU). Furthermore, when multiple constraints become active, the computation time ranges from 3s to 30s (Intel(R) Core(TM) i7-8700 CPU) [20]. Thus, the OCBF method is more suitable for complicated vehicle dynamics and constraints since its computation time is not affected by these factors.

Next, we consider the impact of lane-changing on our OCBF method. For the same two-lane intersection, we allow lane-changing and CAVs can choose any movement (going straight, turning left and right). Since the left lane only allows going straight and turning left, the right-turning CAV in this lane must change its lane. The situation is similar for the left-turning CAV in the right lane. To make a better comparison with the scenario without lane-changing, we use the same arrival data (including the times all CAVs enter the CZ and initial velocities) as the last experiment and only change the lane that the turning CAV arrives at. For example, the left-turning CAVs must arrive at the left lane in the last experiment, but, in this experiment, the lane they enter can be random. The results are shown in Table IV.

We can see that the lane-changing behavior slightly increases all performance measures compared with the results in scenarios disallowing lane-changing. This is expected since a new (floating) MP is added and more control is required to ensure safety. Nevertheless, the changes are minor, fully demonstrating the effectiveness of our method in handling lane changing. Although we have assumed that lane changing only induces a fixed length, we can extend our OCBF method to more complicated lane-changing trajectories, e.g., trajectories fitted by polynomial functions. Note that in the SUMO simulation, it is assumed that a vehicle can jump directly from one lane to another. However, our method still outperforms it

TABLE V
THE INFLUENCE OF ASYMMETRICAL AND HEAVY TRAFFIC FLOWS

Sc.	Methods	Energy	Travel time	Fuel	Ave. obj.
1	SUMO	43.1656	68.6488	40.4731	111.8144
	OCBF	2.6312	22.1812	19.1493	24.8124
2	SUMO	44.4815	96.0517	44.6265	140.5332
	OCBF	3.0999	22.5594	19.5380	25.6593

¹ Scenario 1: the arrival rates at l_1 and l_2 are 540veh/h/lane; while at l_3 to l_8 are 180veh/h/lane.

² Scenario 2: the arrival rates at l_1 , l_2 , l_5 , and l_6 are 540veh/h/lane; while at l_3 , l_4 , l_7 , and l_8 are 180veh/h/lane.

in all metrics, further supporting the advantages of the OCBF controller.

Next, we explore the effect of asymmetrical arrival rates through two scenarios, in order to confirm that our OCBF method is effective even when traffic flows are heavy. In the first scenario, we set the arrival rates in Lanes 1, 2 to be three times as large as Lanes 3-8; while in the second scenario, the arrival rates in Lanes 1, 2, 5, 6 are three times as large as the remaining lanes. The comparison results are shown in Table V.

We can see in the SUMO simulation that traffic flows in lanes with high arrival rates are highly congested with CAVs forming long queues in these lanes. All metrics obtained from SUMO significantly increase compared with the results obtained from medium traffic shown in Table III. However, since the coordination performance under our OCBF controller is much better than SUMO, all metrics remain at low levels, indicating the effectiveness of the OCBF approach in congested situations.

D. Comparison Results for an Actuated Signal-Based Method

To further demonstrate that our method can achieve more efficient use of road resources, we carry out a comparison with a baseline scenario under the control of adaptive four-phase traffic lights using SUMO. In this baseline scenario, an intersection has two lanes in each direction as shown in Fig. 1. It is also worth noting that SUMO employs a gap-based actuated traffic control method whose main idea is to prolong a traffic phase when detecting a continuous stream of traffic. In this experiment, all model parameters used are set at their default values. Then, to investigate the influence of different traffic volumes, we vary the arrival rates to generate different traffic demands. The comparison results are shown in Fig. 3.

The simulation results show that when the traffic volume is low, the actuated traffic control method tends to shorten each phase's duration to reduce the waiting time of vehicles before the stop line. However, we can still observe the undesirable phenomenon that a vehicle is waiting for a green light in order to pass, while the intersection is empty. In contrast, when the traffic volume is high, the actuated traffic control method makes each phase's duration as long as possible to avoid interrupting the traffic flow in that direction. However, vehicles need to wait for an excessive amount of time before the stop line in this situation. Compared with the signal-based method,

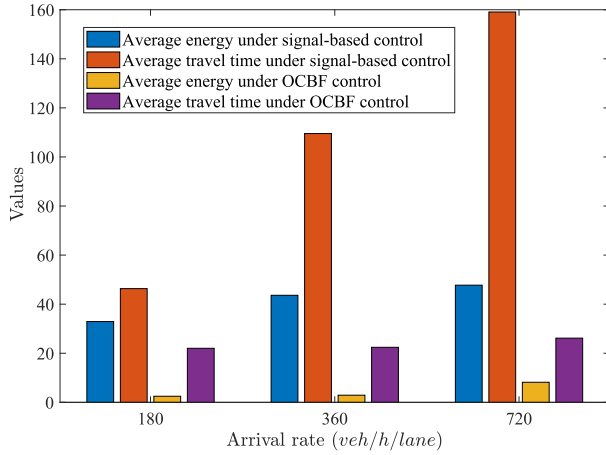


Fig. 3. Comparison results between the OCBF method and the actuated signal-based method.

our optimized signal-free-based method is significantly better in all metrics. Although its performance deteriorates when traffic congestion builds up, the changes in metric values are relatively small. Based on the SUMO simulations, the maximum traffic volume the intersection can support under the signal-based method is approximately 3000veh/h (or 375veh/h/lane). When we continue to increase the arrival rates, almost all roads are jammed so that new vehicles cannot enter the control zone. Thus, the throughput values when arrival rates are 360veh/h/lane and 720veh/h/lane do not differ much. Conversely, traffic flows in all directions are still smooth under the control of the proposed method when the arrival rate is 360veh/h/lane , hence effectively increasing the intersection throughput.

E. The Inclusion of the DR Method

Thus far in our experiments, the FIFO-based queue is used to determine the passing priority when potential conflicts occur. This experiment combines the OCBF controller with a typical resequencing policy, in particular the DR method [15]. When a new CAV enters the CZ, the DR policy inserts it into the optimal position of the original crossing sequence. Note that when combining the OC method with a resequencing policy an update of the arrival times and trajectories of CAVs is required whenever we adjust the original crossing sequence. However, in the OCBF method, CAV i only needs to update the indices of the CAVs with which it conflicts according to the new DR-based queue and follow the original unconstrained optimal trajectory without replanning. In the following experiments, we set $\beta = 5$ and vary the length of some lanes to generate different scenarios. The comparison results are shown in Table VI.

The DR method helps decrease travel times and achieves a better average objective value at the expense of energy consumption, since CAVs need to take more acceleration/deceleration actions to adjust their crossing order. The benefits of the DR method relative to the FIFO policy are more evident in asymmetrical intersections. This is because the FIFO rule may require a CAV that enters the CZ later but

TABLE VI
THE EFFECT OF THE DR METHOD ON THE OCBF METHOD

Scn.	Methods	Energy	Travel time	Fuel	Ave. obj.
1	OCBF+FIFO	5.1261	21.6973	19.1518	113.6126
	OCBF+DR	5.1439	21.6404	18.9364	113.3459
2	OCBF+FIFO	5.9218	20.8093	18.6319	109.9683
	OCBF+DR	6.1080	20.6102	18.9077	109.1590
3	OCBF+FIFO	8.9344	19.3548	17.7396	105.7084
	OCBF+DR	7.0501	17.3253	17.2104	93.6766

¹ Scenario 1 is a symmetrical intersection with all lanes are 300m .

² Scenario 2 is an asymmetrical Intersection with lane 3 and 4 are 200m while lane 1, 2, 5, 6, 7, and 8 are 300m .

³ Scenario 3 is an asymmetrical Intersection with lane 3 and 4 are 200m , lane 5 and 6 are 100m , while lane 1, 2, 7, and 8 are 300m .

TABLE VII
THE INFLUENCE OF THE NONLINEAR VEHICLE DYNAMICS ON THE OCBF METHOD

β	Energy	Travel time	Fuel	Ave. obj.
0.1	0.4751	24.5680	16.6496	2.9319
0.5	1.6822	22.3889	18.2822	12.8767
1	2.4702	21.9973	18.4682	24.4675
2	3.4667	21.7871	18.5832	47.0409

is much closer to the intersection to yield to a CAV that is further away from the intersection. For example, in the above Scenario 3, a CAV enters the CZ from lane 5 that is 100m away from the intersection. It is unreasonable to force it to yield to a CAV entering earlier but located 250m away from the intersection. Our resequencing method can effectively avoid such situations by adjusting crossing sequences in an event-driven way. Note that the OCBF+DR method outperforms the OCBF+FIFO method in all metrics in Scenario 3, since nearly all CAVs arriving at lanes 5 and 6 need to decelerate and even stop due to the FIFO rule, indicating that the DR method is more effective when an intersection geometric configuration is asymmetrical. This finding is consistent with the conclusion given in [34] which provides detailed and comprehensive comparisons for different resequencing methods.

F. The Influence of Nonlinear Vehicle Dynamics and Noise

We first consider the nonlinear vehicle dynamics in (23) and reformulate all CBF constraints according to the new dynamics. For the symmetrical intersection with two lanes in each direction, we vary β from 0.1 to 2 and use the OCBF+FIFO method to coordinate the movements of CAVs. The results are shown in Table VII.

It is clear that the results conform to the results for the double integrator vehicle dynamics (1). When β increases, we are more concerned about the travel time, thus travel time decreases while the energy and fuel consumption rise. Note that though the nonlinear vehicle dynamics are more complicated than the double integrator vehicle dynamics, the only necessary modification is to derive the CBF constraints based on the new dynamics. The computation times for these two different dynamics are nearly the same.

TABLE VIII
THE INFLUENCE OF NOISE ON THE OCBF METHOD

Noise	Energy	Travel time	Fuel	Ave. obj.
no noise	0.4751	24.5680	16.6496	2.9319
$w_{i,u}(t) \in [-0.5, 0.5]$	0.5777	24.6067	16.8714	3.0384
$w_{i,p}(t) \in [-1, 1]$ $w_{i,v}(t) \in [-1, 1]$	5.4662	24.8587	22.3935	7.9521
$w_{i,p}(t) \in [-1, 1]$ $w_{i,v}(t) \in [-1, 1]$ $w_{i,u}(t) \in [-0.5, 0.5]$	5.5723	24.8458	22.3933	8.0569
$w_{i,p}(t) \in [-2, 2]$ $w_{i,v}(t) \in [-2, 2]$ $w_{i,u}(t) \in [-0.5, 0.5]$	31.3250	24.5667	34.1352	33.7817

Next, we have considered both noise and nonlinear dynamics. Due to the measurement errors of sensors and imperfect actuators, there exists random noise in position, velocity, and control inputs. To analyze the influence of noise to the OCBF method, we consider uniformly distributed noise processes ($w_{i,p}(t)$ for the position of CAV i , $w_{i,v}(t)$ for the velocity, and $w_{i,u}(t)$ for the control inputs) for this simulation. We set $\beta = 0.1$ and use the OCBF+FIFO method for all experiments. The results are shown in Table VIII.

The results show that the measurement errors of positions and velocities significantly increase energy consumption. This is because noise causes CAVs to misjudge their states necessitating additional control actions. For example, suppose CAV i is following CAV j and their actual distance is 10m at some time point, but, due to noise, CAV i may misjudge this distance to be 8m, therefore decelerating to enlarge their relative spacing. Then, at the next time point, it may accelerate to keep a desired inter-vehicle space. These frequent acceleration/deceleration maneuvers cause a considerable waste of energy. As uncertainty increases, more control effort is needed to ensure the safety of CAVs when the number of noise sources increases and the noise magnitudes goes up. Note that CAVs may even collide with other CAVs when we continuously increase the magnitude of noise. However, when noise is limited, our method can effectively handle it and does not add any computational burden.

VI. CONCLUSION

This paper presents a decentralized optimal control method for controlling CAVs passing through a multi-lane intersection safely while jointly minimizing the travel time and energy consumption of each CAV. First, CAVs derive a desired reference trajectory generated by unconstrained optimal control. Then, we design a search algorithm for a CAV to identify all conflicting CAVs defining lateral safety constraints that it needs to satisfy. An OCBF controller is then used which optimally tracks the desired reference trajectory while guaranteeing all safety constraints and physical vehicle limitation through appropriate CBFs. Extensive simulation experiments we have conducted show that the proposed method can handle complex objective functions, nonlinear vehicle dynamics, and the presence of noise, and that it is still effective under

the influence of lane-changing behaviors, heavy traffic flows, and asymmetrical intersections. Our ongoing work includes extensions of decentralized CAV control to traffic networks and to ensuring the feasibility of the QPs in challenging environments where this may not always be the case. Finally, when the CAV trajectories involve curves, we plan to include more complicated lateral dynamics that better model vehicle behavior in such cases, as recently shown in [35].

REFERENCES

- [1] 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. Intell. Transp. Syst.*, vol. 18, no. 5, pp. 1066–1077, May 2017.
- [2] L. Chen and C. Englund, "Cooperative intersection management: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 2, pp. 570–586, Feb. 2016.
- [3] L. Li, D. Wen, and D. Y. Yao, "A survey of traffic control with vehicular communications," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 1, pp. 425–432, Feb. 2014.
- [4] R. Hult, G. R. Campos, E. Steinmetz, L. Hammarstrand, P. Falcone, and H. Wymersch, "Coordination of cooperative autonomous vehicles: Toward safer and more efficient road transportation," *IEEE Signal Process. Mag.*, vol. 33, no. 6, pp. 74–84, Nov. 2016.
- [5] S. A. Fayazi and A. Vahidi, "Mixed-integer linear programming for optimal scheduling of autonomous vehicle intersection crossing," *IEEE Trans. Intell. Veh.*, vol. 3, no. 3, pp. 287–299, Sep. 2018.
- [6] H. Xu, S. Feng, Y. Zhang, and L. Li, "A grouping-based cooperative driving strategy for CAVs merging problems," *IEEE Trans. Veh. Technol.*, vol. 68, no. 6, pp. 6125–6136, Jun. 2019.
- [7] X. Qian, J. Gregoire, A. de La Fortelle, and F. Moutarde, "Decentralized model predictive control for smooth coordination of automated vehicles at intersection," in *Proc. Eur. Control Conf. (ECC)*, Jul. 2015, pp. 3452–3458.
- [8] H. Xu, Y. Zhang, C. G. Cassandras, L. Li, and S. Feng, "A bi-level cooperative driving strategy allowing lane changes," *Transp. Res. C, Emerg. Technol.*, vol. 120, Nov. 2020, Art. no. 102773.
- [9] K. Dresner and P. Stone, "A multiagent approach to autonomous intersection management," *J. Artif. Intell. Res.*, vol. 31, pp. 591–656, Mar. 2008.
- [10] A. A. Malikopoulos, C. G. Cassandras, and Y. Zhang, "A decentralized energy-optimal control framework for connected automated vehicles at signal-free intersections," *Automatica*, vol. 93, pp. 244–256, Jul. 2018.
- [11] Y. Zhang and C. G. Cassandras, "Decentralized optimal control of connected automated vehicles at signal-free intersections including comfort-constrained turns and safety guarantees," *Automatica*, vol. 109, pp. 1–9, Nov. 2019.
- [12] Y. Meng, L. Li, F.-Y. Wang, K. Li, and Z. Li, "Analysis of cooperative driving strategies for nonsignalized intersections," *IEEE Trans. Veh. Technol.*, vol. 67, no. 4, pp. 2900–2911, Apr. 2018.
- [13] S. I. Guler, M. Menendez, and L. Meier, "Using connected vehicle technology to improve the efficiency of intersections," *Transp. Res. C, Emerg. Technol.*, vol. 46, pp. 121–131, Sep. 2014.
- [14] Z. Li, M. Pourmehrab, L. Eleftheriadou, and S. Ranka, "Intersection control optimization for automated vehicles using genetic algorithm," *J. Transp. Eng., A, Syst.*, vol. 144, no. 12, Dec. 2018, Art. no. 04018074.
- [15] Y. Zhang and C. G. Cassandras, "A decentralized optimal control framework for connected automated vehicles at urban intersections with dynamic resequencing," in *Proc. IEEE Conf. Decis. Control (CDC)*, Dec. 2018, pp. 217–222.
- [16] H. Xu, Y. Zhang, L. Li, and W. Li, "Cooperative driving at unsignalized intersections using tree search," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 11, pp. 4563–4571, Nov. 2020.
- [17] L. Eleftheriadou, *An Introduction to Traffic Flow Theory*, vol. 84. New York, NY, USA: Springer, 2014.
- [18] M. Pourmehrab, L. Eleftheriadou, S. Ranka, and M. Martin-Gasulla, "Optimizing signalized intersections performance under conventional and automated vehicles traffic," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 7, pp. 2864–2873, Jul. 2020.
- [19] M. Pourmehrab, P. Emami, M. Martin-Gasulla, J. Wilson, L. Eleftheriadou, and S. Ranka, "Signalized intersection performance with automated and conventional vehicles: A comparative study," *J. Transp. Eng., A, Syst.*, vol. 146, no. 9, Sep. 2020, Art. no. 04020089.

- [20] W. Xiao and C. G. Cassandras, "Decentralized optimal merging control for connected and automated vehicles with safety constraint guarantees," *Automatica*, vol. 123, Jan. 2021, Art. no. 109333.
- [21] B. Chalaki and A. A. Malikopoulos, "Optimal control of connected and automated vehicles at multiple adjacent intersections," 2020, *arXiv:2008.02379*.
- [22] W. Xiao, C. G. Cassandras, and C. A. Belta, "Bridging the gap between optimal trajectory planning and safety-critical control with applications to autonomous vehicles," *Automatica*, vol. 129, Jul. 2021, Art. no. 109592.
- [23] A. D. Ames, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs with application to adaptive cruise control," in *Proc. 53rd IEEE Conf. Decis. Control*, Dec. 2014, pp. 6271–6278.
- [24] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *Proc. 18th Eur. Control Conf. (ECC)*, Jun. 2019, pp. 3420–3431.
- [25] W. Xiao and C. Belta, "Control barrier functions for systems with high relative degree," in *Proc. IEEE 58th Conf. Decis. Control (CDC)*, Nice, France, Dec. 2019, pp. 474–479.
- [26] W. Xiao and C. G. Cassandras, "Decentralized optimal merging control for connected and automated vehicles with optimal dynamic rescheduling," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2020, pp. 4090–4095.
- [27] K. Vogel, "A comparison of headway and time to collision as safety indicators," *Accident Anal. Prevention*, vol. 35, no. 3, pp. 427–433, May 2003.
- [28] A. A. Malikopoulos, "Stochastic optimal control for series hybrid electric vehicles," in *Proc. Amer. Control Conf.*, Jun. 2013, pp. 1189–1194.
- [29] H. K. Khalil and J. W. Grizzle, *Nonlinear Systems*, vol. 3. Upper Saddle River, NJ, USA: Prentice-Hall, 2002.
- [30] W. Xiao, C. G. Cassandras, and C. Belta, "Decentralized merging control in traffic networks with noisy vehicle dynamics: A joint optimal control and barrier function approach," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 3162–3167.
- [31] W. Xiao, C. Belta, and C. G. Cassandras, "Decentralized merging control in traffic networks: A control barrier function approach," in *Proc. 10th ACM/IEEE Int. Conf. Cyber-Phys. Syst.*, Montreal, QC, Canada, Apr. 2019, pp. 270–279.
- [32] P. A. Lopez *et al.*, "Microscopic traffic simulation using SUMO," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 2575–2582.
- [33] M. A. S. Kamal, M. Mukai, J. Murata, and T. Kawabe, "Model predictive control of vehicles on urban roads for improved fuel economy," *IEEE Trans. Control Syst. Technol.*, vol. 21, no. 3, pp. 831–841, Apr. 2012.
- [34] H. Xu, C. G. Cassandras, L. Li, and Y. Zhang, "Comparison of cooperative driving strategies for CAVs at signal-free intersections," *IEEE Trans. Intell. Transp. Syst.*, early access, Apr. 16, 2021, doi: 10.1109/TITS.2021.3071456.
- [35] W. Xiao and C. G. Cassandras, "Decentralized optimal merging control for connected and automated vehicles on curved roads," in *Proc. 60th IEEE Conf. Decis. Control (CDC)*, Dec. 2021, pp. 2677–2682.



Huile Xu received the B.S. degree from Chongqing University, China, in 2016, and the Ph.D. degree from Tsinghua University, China, in 2021. He was a Visiting Ph.D. Student with the Division of Systems Engineering, Boston University, USA, and the Center for Information and Systems Engineering, Boston University, from 2019 to 2020. He is currently a Post-Doctoral Researcher with the Institute of Automation, Chinese Academy of Sciences, China. His research interests include cooperative driving, intelligent vehicles, and optimal control.



Wei Xiao (Member, IEEE) received the B.Sc. degree from the University of Science and Technology Beijing, China, in 2013, and the M.Sc. degree from the Chinese Academy of Sciences (Institute of Automation), China, in 2016. He is currently pursuing the Ph.D. degree in systems engineering with Boston University, Brookline, MA, USA. His research interests include control theory, formal methods, and machine learning, with particular emphasis on robotics and traffic control. He received the Outstanding Student Paper Award at the 2020 IEEE Conference on Decision and Control.



Christos G. Cassandras (Life Fellow, IEEE) received the degrees from Yale University, Stanford University, and Harvard University. From 1982 to 1984, he was with ITP Boston, Inc., where he worked on the design of automated manufacturing systems. From 1984 to 1996, he was a Faculty Member at the Department of Electrical and Computer Engineering, University of Massachusetts at Amherst. He is currently a Distinguished Professor of engineering at Boston University. He is also the Head of the Division of Systems Engineering, a Professor of electrical and computer engineering, and the Co-Founder of the Center for Information and Systems Engineering (CISE), Boston University. He specializes 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. He has published over 450 refereed articles in these areas, and six books. In addition to his academic activities, he worked extensively with industrial organizations on various systems integration projects and the development of decision support software. He has most recently collaborated with The MathWorks, Inc., in the development of the discrete event and hybrid system simulator SimEvents. He is a member of Phi Beta Kappa and Tau Beta Pi. He is a fellow of the IFAC. He was a 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) for Discrete Event Systems: Modeling and Performance Analysis, the 2011 Prize and the 2014 Prize for the IBM/IEEE Smarter Planet Challenge Competition, the 2014 Engineering Distinguished Scholar Award at Boston University, several honorary professorships, the 1991 Lilly Fellowship, and the 2012 Kern Fellowship. He has guest-edited several technical journal issues and also serves on several journal editorial boards, including the Editor of *Automatica*. From 1998 to 2009, he was the Editor-in-Chief of the IEEE TRANSACTIONS ON AUTOMATIC CONTROL. He has also served as an Editor for *Technical Notes and Correspondence* and an Associate Editor. He was the 2012 President of the IEEE Control Systems Society (CSS). He has also served as the Vice President for Publications and on the Board of Governors for the CSS, as well as on several IEEE committees, and has chaired several conferences. He has been a Plenary/Keynote Speaker at numerous international conferences, including the 2017 IFAC World Congress, the American Control Conference in 2001, and the IEEE Conference on Decision and Control in 2002 and 2016. He has also been an IEEE Distinguished Lecturer.



Yi Zhang (Member, IEEE) received the B.S. and M.S. degrees from Tsinghua University, China, in 1986 and 1988, respectively, and the Ph.D. degree from the University of Strathclyde, U.K., in 1995. He is currently a Professor of control science and engineering with Tsinghua University, with his current research interests focusing on intelligent transportation systems. His research interests include intelligent vehicle-infrastructure cooperative systems, analysis of urban transportation systems, urban road network management, traffic data fusion and dissemination, and urban traffic control and management. His research fields also cover the advanced control theory and applications, advanced detection and measurement, and systems engineering.



Li Li (Fellow, IEEE) is currently an Associate Professor with the Department of Automation, Tsinghua University, Beijing, China, where he is involved in the fields of complex and networked systems, intelligent control and sensing, intelligent transportation systems, and intelligent vehicles. He has published over 90 SCI indexed international journal articles and over 70 international conference papers as a first/corresponding author. He serves as an Associate Editor for the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, a member of the Editorial Advisory Board for *Transportation Research Part C: Emerging Technologies*, and a member of the Editorial Board for *Transport Reviews*.