2020 IEEE Intelligent Transportation Systems Conference (ITSC)
Rhodes, Greece. September 20-23, 2020 (Virtual)

Decentralized Optimal Control in Multi-lane Merging for Connected and Automated Vehicles

Wei Xiao, Christos G. Cassandras and Calin Belta

Abstract— **We address the problem of optimally controlling Connected and Automated Vehicles (CAVs) arriving from two multi-lane roads and merging at multiple points where the objective is to jointly minimize the travel time and energy consumption of each CAV subject to speed-dependent safety constraints, as well as speed and acceleration constraints. This problem was solved in prior work for two single-lane roads. A direct extension to multi-lane roads is limited by the computational complexity required to obtain an explicit optimal control solution. Instead, we propose a general framework that converts a multi-lane merging problem into a decentralized optimal control problem for each CAV in a less-conservative way. To accomplish this, we employ a joint optimal control and barrier function method to efficiently get an optimal control for each CAV with guaranteed satisfaction of all constraints. Simulation examples are included to compare the performance of the proposed framework to a baseline provided by human-driven vehicles with results showing significant improvements in both time and energy metrics.**

I. INTRODUCTION

Traffic management at merging points (usually, highway on-ramps) is one of the most challenging problems within a transportation system in terms of safety, congestion, and energy consumption, in addition to being a source of stress for many drivers [1], [2], [3]. Advances in next-generation transportation technologies and the emergence of Connected and Automated Vehicles (CAVs) have the potential to drastically improve a transportation network's performance by better assisting drivers in making decisions, ultimately reducing energy consumption, air pollution, congestion and accidents.

Most research work just focuses on the single lane merging problem [4], [5], [6], with limited work done in the multi-lane merging problem. In our recent work [7], we addressed the merging problem through a decentralized optimal control (OC) formulation and derived explicit analytical solutions for each CAV when no constraints are active. We have extended the solution to include constraints, in which case the computational cost depends on the number of constraints becoming active; we have found this to become potentially prohibitive for a CAV to determine through on-board resources. In addition, our analysis has thus far assumed no noise in the vehicle dynamics and sensing measurements, and the dynamics have precluded nonlinearities.

Supported in part by NSF under grants ECCS-1509084, DMS-1664644, CNS-1645681, IIS-1723995, CPS-1446151, by AFOSR under grant FA9550-19-1-0158, by ARPA-E's NEXTCAR program under grant DE-AR0000796 and by the MathWorks.

The authors are with the Division of Systems Engineering and Center for Information and Systems Engineering, Boston University, Brookline, MA, 02446, USA {xiaowei, cgc, cbelta}@bu.edu

To address the limitations above, one can adopt Model Predictive Control (MPC) (e.g., [8], [4], [9]) or the Control Barrier Function (CBF) method [10], [11]. MPC is very effective for problems with simple (usually linear or linearized) dynamics. Unlike MPC, the CBF method does not use states as decision variables in its optimization process; instead, any continuously differentiable state constraint is mapped onto a new constraint on the control input. A control input that satisfies this new constraint is guaranteed to also satisfy the original constraint. We have adopted this approach to the single-lane merging problem in recent work [12] and shown that it provides good approximations of the analytically obtained OC solutions. To account for both optimality and computational complexity, we developed a *joint optimal control and barrier function (OCBF)* controller in [13] for a two-lane merging problem. The implementation of this approach is hard for multi-lane merging, especially in determining the safety constraints that a CAV has to satisfy. The common approach to avoid such complex safety constraint determination is to treat an entire conflict area as a point, which is conservative (e.g., for an intersection, see [14]). Alternatively, the conflict area can be partitioned according to lane intersections and a tree search approach may be used to find a feasible path for each CAV [15]; this approach is limited by the computational complexity due to the high-dimensional search space involved.

The contribution of this paper is to show how we can transform a multi-lane merging problem into a multi-point merging problem in a simpler and less-conservative way. Specifically, we first determine the merging points that a CAV must pass through and construct queueing tables maintained by a coordinator associated with the merging area. Using a simple search through these tables, we determine the safe merging and rear-end safety constraints that a CAV has to satisfy, hence transforming the multi-lane merging problem into a decentralized optimal control problem for each CAV. Finally, we use the aforementioned OCBF method to solve these optimal control problems. The main advantages of the proposed framework lie in the optimality it provides, its computational efficiency, safety guarantees, and good generalization properties for more complex traffic scenarios. Simulation results of the proposed framework have shown significantly better performance compared to human drivers.

II. PROBLEM FORMULATION

The multi-lane merging problem arises when traffic must be joined from two different roads, usually associated with a main and a merging road as shown in Fig.1. Each road has two lanes (as we will see, the same modeling method can be applied to more than two lanes). We label the lanes l_1, l_2 and l_3, l_4 for the main and merging roads respectively, with corresponding origins O_1, O_2, O_3, O_4. Only the CAVs in lane l_2 can change lanes to l_1. In addition, the CAVs in lane l_3 have the option to merge into either lane l_1 or l_2 (the main benefit being that the CAV in l_3 can surpass a group of CAVs in l_4 when l_4 is congested). Finally, the CAVs in lane l_4 can only merge to l_2.

In our original single-lane merging problem [7] only lanes l_2, l_4 are involved and the only merging point is M_3 in Fig.1. Here, CAVs from lanes l_1, l_2, l_3, l_4 may merge at the three fixed merging points M_2, M_3, M_4. In addition, a CAV from lane l_2 may merge into l_1 at an arbitrary merging point $M_{i,1}$, as long as this point is located prior to M_2. We consider the case where all traffic consists of CAVs randomly arriving at the four lanes joined at the Merging Points (MPs) $M_{i,1}, M_2, M_3, M_4$ where a collision may occur. The road segment from O_2 or O_4 to the merging point M_3 has a length L_3 and is called the *Control Zone (CZ)*. The segment from O_1 to $M_{i,1}$ for CAV i has a length $L_{i,1}$ (which is variable and depends on i). The segment from O_2 or O_3 to M_2 has a length L_2.

We assume that CAVs do not overtake each other in the CZ (unless so dictated by the CAV's controller to be developed in the sequel), that $L_{i,1} < L_2$, and that the merging point M_4 is within the CZ. Moreover, note that if the controller determines that a CAV needs to change lanes from l_2 to l_1, then it has to travel an additional distance; we assume that this extra distance is a constant $l > 0$. The same constant applies to CAVs in lane l_3 which choose to merge into l_1 at M_4 (as opposed to merging into l_2).

A coordinator (typically a Road Side Unit (RSU)) is associated with the MP M_3 whose function is to maintain First-In-First-Out (FIFO) queues of all CAVs regardless of lanes based on their arrival time at the CZ and to enable real-time communication with the CAVs that are in the CZ as well as the last one leaving the CZ (in particular, the coordinator does not make control decisions; this is done in decentralized fashion on-board each CAV). The FIFO assumption (so that CAVs cross the MP in their order of arrival) is made for simplicity and often to ensure fairness; however, it can be relaxed through dynamic resequencing schemes as described, for example, in [14], [16]. Since we have two lanes in the main road, we need two queues to manage each CAV sequence leaving the CZ via l_1 and l_2 respectively, as shown in Fig. 1. Note that the number of queues equals the number of lanes in the main road, thus this framework can be easily extended to other multi-lane road traffic configurations, such as intersections.

Let $S_1(t), S_2(t)$ be the sets of the FIFO-ordered CAV indices associated with the two possible CZ exit lanes l_1 and l_2. To maintain a single unique index for each CAV, let $n > 0$ be a large enough integer representing the road capacity over L_3 in terms of the number of CAVs that can be accommodated. Then, let the set of possible CAV indices in $S_2(t)$ be $\{0, 1, \ldots, n-1\}$ and that in $S_1(t)$ be $\{n, n+1, \ldots, 2n-1\}$. Thus, CAV $n+j$ $(j \in \mathbb{N})$ belongs to $S_1(t)$. The CAVs indexed by n or 0 are the ones that have just left the CZ from l_1, l_2 respectively. Let $N_1(t), N_2(t)$ be the cardinalities of $S_1(t), S_2(t)$, respectively. Observe that the CAVs in any one queue may have a physical conflict (i.e., collisions may happen) with the CAVs in the other queue only in lanes l_2, l_3, but not in lanes l_1, l_4. Thus, we assign a newly arriving CAV according to the following cases:

(i) If a CAV arrives at time t at lane l_1, it is assigned to $S_1(t)$ with an index $n + N_1(t)$.

(ii) If a CAV arrives at time t at lane l_2, a decision is made (as decsribed later) on whether it exits the CZ through l_2 or switches to l_1 at $L_{i,1}$. This CAV is assigned to both $S_1(t)$ and $S_2(t)$ with the index $N_2(t)$ if it chooses to stay in l_2 (e.g., CAV 2 in Fig. 1) or the index $n + N_1(t)$ if it switches to l_1 (e.g., CAV $n + 3$ in Fig. 1).

(iii) If a CAV arrives at time t at lane l_3, it is assigned to both $S_1(t)$ and $S_2(t)$ with the index $n + N_1(t)$ if the control decision is to merge to lane l_1 or the index $N_2(t)$ if it merges to lane l_2.

(iv) If a CAV arrives at time t at lane l_4, it is assigned to $S_2(t)$ with the index $N_2(t)$.

Note that in the above case (ii), the index of the CAV arriving at l_2 is dropped from $S_2(t)$ (or $S_1(t)$) after it changes its lane to l_1 at $M_{i,1}$ (or passes M_2). In the above case (iii), the index of the CAV arriving at lane l_3 is dropped from $S_1(t)$ (or $S_2(t)$) after it passes M_2 if it chooses to merge into l_2 (or l_1). In summary, the index of any CAV arriving at O_2 or O_3 will be dropped from queue $S_1(t)$ or $S_2(t)$ after it passes its first MP. All CAV indices in $S_2(t)$ decrease by one when a CAV passes MP M_3 and the CAV whose index becomes -1 is dropped (similarly for $S_1(t)$, the CAV leaving the CZ through M_4 whose index becomes $n-1$ is dropped). Observe that this scheme allows any CAV $i \in S_1(t)$ to look up only queue table $S_1(t)$ (similarly for $S_2(t)$ if $i \in S_2(t)$) in order to identify all possible collisions with other CAVs, without any need to consider the other queue.

The vehicle dynamics for each CAV $i \in S_1(t) \cup S_2(t)$ along the lane to which it belongs takes the form

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

where $x_i(t)$ denotes the distance to the origin O_1 or O_2, O_3, O_4 along the lane that 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, $w_{i,1}(t), w_{i,2}(t)$ denote two random processes defined in an appropriate probability space to capture possible noise. We consider two objectives for each CAV subject to three constraints, as detailed next.

Objective 1 (Minimize travel time): Let t_i^0 and t_i^m denote the time that CAV $i \in S_1(t) \cup S_2(t)$ arrives at the origin O_1 or O_2, O_3, O_4 and the time that CAV i leaves the CZ (through either M_3 or M_4), respectively. We wish to minimize the travel time $t_i^m - t_i^0$ for CAV i.

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

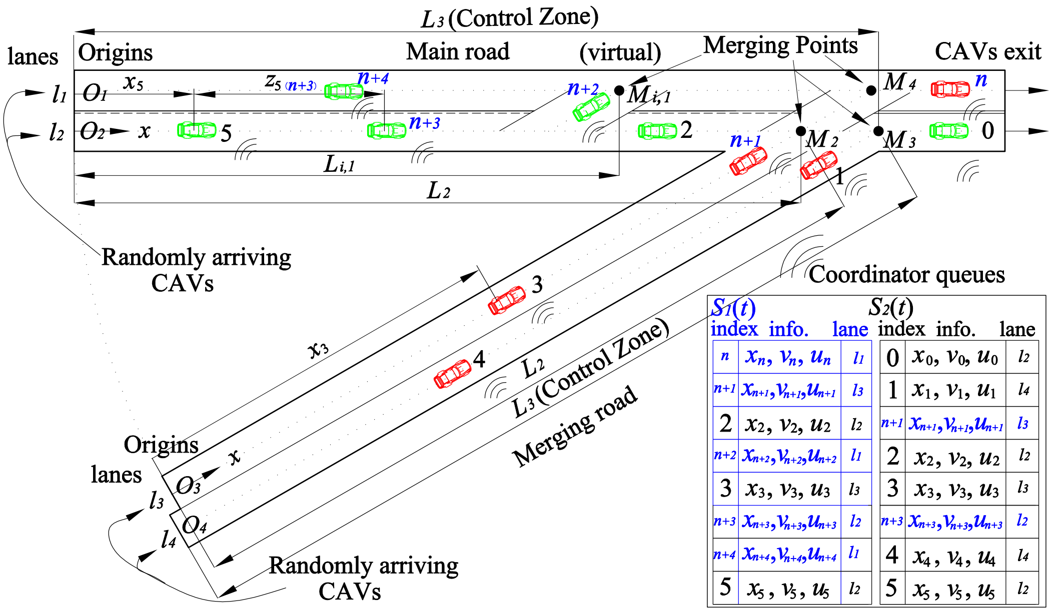

Fig. 1. The multi-lane merging problem. Collisions may happen at the merging points $M_{i,1}, M_2, M_3, M_4$.

$i \in S_1(t) \cup S_2(t)$ expressed as

$$\min_{u_i(t)} \int_{t_i^0}^{t_i^m} \mathcal{C}(u_i(t))dt, \tag{2}$$

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

Constraint 1 (Safety constraint): Let i_p denote the index of the CAV which physically immediately precedes $i \in S_1(t) \cup S_2(t)$ in the CZ (if one is present). We require that the distance $z_{i,i_p}(t) \equiv x_{i_p}(t) - x_i(t)$ be constrained by:

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

where φ denotes the reaction time (as a rule, $\varphi = 1.8$ is used, e.g., [17]). 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 (generally dependent on i and i_p but taken to be a constant over all CAVs for simplicity).

Constraint 2 (Safe merging): Let $t_i^{m_p}$, $p \in \{1,2,3,4\}$ denote the arrival time of CAV $i \in S_1(t) \cup S_2(t)$ (note that CAV i will only pass at most two of these MPs) at the merging points $M_{i,1}, M_2, M_3, M_4$, respectively. There should be enough safe space at these MPs for a merging CAV i to cut in, i.e.,

$$z_{i,j}(t_i^{m_p}) \geq \varphi v_i(t_i^{m_p}) + \delta, \qquad p \in \{1,2,3,4\}, \tag{4}$$

where $j \in S_1(t) \cup S_2(t)$ is the CAV that may collide with i (j may not exist) at the merging points $M_{i,1}, M_2, M_3, M_4$. Observe that since a CAV crosses at most two of the four MPs, CAV i only needs to satisfy the safe merging constraints above corresponding to the MPs that it will actually cross (e.g., CAV 1 in Fig. 1 only needs to satisfy the third constraint in (4)). The index j corresponding to each i is generally hard to determine; we will resolve this issue in the next section through a conflict-point-based method.

Constraint 3 (Vehicle limitations): Finally, there are constraints on the speed and control for each $i \in S_1(t) \cup S_2(t)$:

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

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

A common way to minimize energy consumption is by minimizing the control input effort $u_i^2(t)$. By normalizing travel time and $u_i^2(t)$, and using $\alpha \in [0,1)$, we construct a convex combination as follows:

$$J_i(u_i(t)) = \beta(t_i^m - t_i^0) + \int_{t_i^0}^{t_i^m} \frac{1}{2}u_i^2(t)dt, \tag{6}$$

where $\beta = \frac{\alpha \max\{u_{\max}^2, u_{\min}^2\}}{2(1-\alpha)}$ is a weight factor that can be adjusted through $\alpha \in [0,1)$ to penalize travel time relative to the energy cost. Then, we have the following problem formulation:

***Problem** 1:* For each CAV $i \in S_1(t) \cup S_2(t)$ governed by dynamics (1), determine a control law such that (6) 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 Merging Problem Solution

We now show how to decompose Problem 1 into a multi-point merging problem for each CAV and use the CBF method to account for constraints while tracking a CAV trajectory obtained through OC. We also take advantage of the robustness to noise that the CBF approach offers.

However, determining the exact merging constraints in (4) that a CAV $i \in S_1(t) \cup S_2(t)$ has to satisfy is challenging since there are four lanes and the traffic is asymmetric. This is even harder for more lanes and other scenarios, such as intersections. Using the approach introduced in [7] and considering the multi-lane merging problem in Fig. 1, there are 15 cases, making this hard to implement. Moreover, this approach does not scale well for more complicated cases. Therefore, we propose a conflict-point based approach to simplify this process, as described next.

A. Lane Merging Determination Strategy

When a new CAV $i \in S_1(t) \cup S_2(t)$ arrives at O_2 or O_3, it has the option of exiting the CZ through lane l_1 or l_2. In addition, if it arives at O_2 and decides to merge to l_1, it must also determine the location of the variable MP $M_{i,1}$.

Let us begin with the first issue. Determining the lane from which a CAV should exit the CZ may be addressed using the optimal dynamic resequencing method from [16]. This approach becomes computationally intensive; for example in the single-lane merging problem we have found this to require 3 to 30sec in MATLAB [16], and this will generally increase in the multi-lane merging problem at hand. Although this remains an option, in this paper we focus on computational efficiency by adopting the following lane-merging decision strategy: we seek to balance the expected number of CAVs in the two lanes in order to improve the cost (6) on average. In a queueing-theoretic context, this implies adopting a shortest-queue-first policy which is known to be often optimal in terms of minimizing average travel times. Thus, for any arriving CAV i at O_2 or O_3 at t_i^0:

$$i \in \begin{cases} S_1(t), & \text{if } N_1(t_i^0) < N_2(t_i^0) \\ S_2(t), & \text{otherwise} \end{cases}, \quad t \in [t_i^0, t_i^m]. \tag{7}$$

Next, we address the issue of selecting the location of the MP $M_{i,1}$ for a CAV i arriving at O_2, if its decision is $i \in S_1(t)$ above. There are three important observations to make: (i) The *unconstrained* optimal control for such i is independent of the location of $M_{i,1}$ since we have assumed that lane-changing will only induce a fixed extra length l. (ii) The OC solution under the first safe-merging constraint in (4) is better (i.e., lower cost in (6)) 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 $t_i^{m_1}$ whereas the latter requires the constraint (3) to be satisfied over all $t \in [t_i^0, t_i^{m_1}]$. It follows that the merging point $M_{i,1}$ should be as close as possible to M_2 (i.e., $L_{i,1}$ should be as large as possible), since the safe-merging constraint between i and $i-1$ will become a rear-end safety constraint after $M_{i,1}$. (iii) In addition, CAV i arriving at O_2 may also be constrained by its physically preceding CAV i_p (if one exists) in lane l_2. In this case, CAV i needs to consider both the rear-end safety constraint with i_p and the safe-merging constraint with $i-1$. Thus, the solution is more constrained (hence, more sub-optimal) if i stays in lane l_2 after the rear-end safety constraint due to i_p becomes active. We conclude that in this case CAV i should merge to lane l_1 when the rear-end safety constraint with i_p in lane l_2 first becomes active, i.e., $L_{i,1}$ is determined by

$$L_{i,1} = x_i^*(t_i^a) \tag{8}$$

where $x_i^*(t)$ denotes the unconstrained optimal trajectory of CAV i (as determined in Sec. III-C), and $t_i^a \geq t_i^0$ is the time instant when the rear-end safety constraint first becomes active between i and i_p in lane l_2; if this constraint never becomes active, then $t_i^a = t_i^{m_2}$. The value of t_i^a is determined from (3) by

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

where $x_{i_p}^*(t), v_i^*(t)$ are the *unconstrained* optimal trajectory and optimal speed respectively of CAV i_p. If, however, CAV i_p's optimal trajectory includes a constrained arc, then (9) is only an approximation (in fact, an upper bound) of t_i^a. In summary, if CAV i never encounters a point on l_2 where its rear-end safety constraint becomes active, we set $L_{i,1} = L_2$, otherwise $L_{i,1}$ is determined through (8)-(9).

B. Merging Constraint Determination Strategy

The CAVs arriving at lanes l_2, l_3 will pass two MPs. On the other hand, CAVs arriving at lane l_1 will pass either one or two MPs (depending on whether i and $i-1$ are in the same lane or not), whereas all CAVs arriving at l_4 will pass only MP M_3. Moreover, CAVs arriving at lanes l_2, l_3 may pass through different MPs, depending on which lane they choose to merge into following the strategy presented in the last subsection. Since all MPs that a CAV has to pass are now determined, we augment the FIFO queues in Fig. 1 with the original lane and the MP information for each CAV as shown in Fig. 2. The current and original lanes are shown in the third and fourth column, respectively. The last two columns indicate the first and second MPs for each CAV (note that all CAVs arriving at lane l_4 and some CAVs arriving at lane l_1 have only one MP, in which case the first MP is left blank).

$S_1(t)$

index	info.	cur. lane	ori. lane	1st MP	2nd MP
n	x_n, v_n, u_n	l_1	l_3	M_2	M_4
$n+1$	$x_{n+1}, v_{n+1}, u_{n+1}$	l_3	l_3	M_2	M_4
2	x_2, v_2, u_2	l_2	l_2	M_2	M_3
$n+2$	$x_{n+2}, v_{n+2}, u_{n+2}$	l_1	l_2	$M_{i,1}$	M_4
3	x_3, v_3, u_3	l_3	l_3	M_2	M_3
$n+3$	$x_{n+3}, v_{n+3}, u_{n+3}$	l_2	l_2	$M_{i,1}$	M_4
$n+4$	$x_{n+4}, v_{n+4}, u_{n+4}$	l_1	l_1	$M_{i,1}$	M_4
5	x_5, v_5, u_5	l_2	l_2	M_2	M_3

$S_2(t)$

index	info.	cur. lane	ori. lane	1st MP	2nd MP
0	x_0, v_0, u_0	l_2	l_2	M_2	M_3
1	x_1, v_1, u_1	l_4	l_4		M_3
$n+1$	$x_{n+1}, v_{n+1}, u_{n+1}$	l_3	l_3	M_2	M_4
2	x_2, v_2, u_2	l_2	l_2	M_2	M_3
3	x_3, v_3, u_3	l_3	l_3	M_2	M_3
$n+3$	$x_{n+3}, v_{n+3}, u_{n+3}$	l_2	l_2	$M_{i,1}$	M_4
4	x_4, v_4, u_4	l_4	l_4		M_3
5	x_5, v_5, u_5	l_2	l_2	M_2	M_3

Fig. 2. The extended coordinator queue tables.

When a new CAV i arrives at O_1 (or O_2, O_3, O_4) and has determined whether it will merge into another lane or not (based on the last subsection), it looks up the extended queue tables in Fig. 2 which already contain all prior CAV state and MP information. If $i \in S_1(t)$, it looks up the extended FIFO queue $S_1(t)$, otherwise, it looks up $S_2(t)$. From the *current lane* column in Fig. 2, CAV i can determine its current physically immediately preceding CAV i_p if one exists. Moreover, CAV i can determine the safe-merging constraints that it should satisfy (i.e., with respect to which CAV j in (4) in the queue) upon its arrival at any origin.

The precise process through which each arriving CAV i looks up each queue $S_1(t)$ and $S_2(t)$ in Fig. 2 is a follows.

CAV i compares its *original lane* and MP information to that of every CAV in each queue *starting with the last row and moving up*. Depending on which column (among the last three columns) matches first, there are four possible cases (a much smaller number than 15 if the approach in [7], [12], [13] were followed). *This process terminates the first time that any one of these four cases is satisfied at some row.* If that does not happen, this implies that CAV i does not have to satisfy any safe-merging constraint. Let $type(i) \in \{1, 2\}$ be such that $type(i) = 1$ if $i \geq n$ and $type(i) = 2$ otherwise. Then, the four cases are:

(1) All last three columns match first.

(2) [1^{st} MP column matches with $j \in S_1(t)$ (or $S_2(t)$) first] & [$type(i) = type(j)$].

(3) [1^{st} MP column matches with $j \in S_1(t)$ (or $S_2(t)$) first) & [$type(i) \neq type(j)$].

(4) The 2^{nd} MP column matches first.

If none of the four cases above is satisfied, then CAV i does not have to satisfy any safe-merging constraint. In summary, a newly arriving CAV may have to satisfy at most three safety (or safe-merging) constraints in Fig. 1. If the corresponding k or i_p is not found in the above cases, then the related safe-merging or safety constraint is skipped.

Updating $S_1(t)$ and $S_2(t)$. Observe that while the MP information in the last two columns of each queue in Fig. 2 remains unchanged, the same is not true for the *current lane* information. More precisely, the two queues need to be updated whenever one of the following four events takes place: (i) A new CAV arrives at the CZ and is added to one or both queues. (ii) A CAV $i \in S_2(t)$ (or $S_1(t)$) leaves the CZ causing the index of any CAV $j \in S_1(t) \cup S_2(t)$ with $type(j) = 2$ (or $type(j) = 1$) to decrease by 1 and the CAV whose index is -1 (or $n-1$ in $S_1(t)$) is removed from $S_2(t)$ (or $S_1(t)$). Note that CAV -1 only appears in $S_2(t)$ (CAV $n-1$ only appears in $S_1(t)$), as discussed in Sec. II. (iii) A CAV changes lanes, causing an update in the *current lane* column in Fig. 2. This event is important because the value of i_p for any CAV i already in a queue may change, since its original i_p may merge into another lane. (iv) A CAV overtake event when a CAV passes M_3 or M_4. This may occur when a CAV $i \in S_2(t)$ (or $S_1(t)$) overtakes $i-1 \in S_1(t) \cup S_2(t)$ when the two CAVs pass different MPs without conflict. Thus, if i passes M_3 or M_4 and $i-1$ is still in one of the queues, we need to re-order $S_2(t)$ (or $S_1(t)$) according to the incremental position order, so that CAV $i+1$ can properly identify its $(i+1)_p$. For example, consider $i = 4$, $i-1 = n+3$, $i+1 = 5$ in queue $S_2(t)$ of Fig. 1. CAV 4 can overtake $n+3$, and its current lane will become l_2 when it passes M_3. When this happens, CAV 5 may mistake CAV 4 as its i_p by looking at the new current lane entry for it, which is now in l_2. In reality, $i_p = n+3$ as long as CAV $n+3$ is still in lane l_2. This is avoided by re-ordering queue $S_2(t)$ according to the position information when this event occurs (i.e., swapping rows for CAVs 4 and $n+3$).

C. Joint Optimal and Barrier Function Controller

Once a newly arriving CAV $i \in S_1(t) \cup S_2(t)$ has determined all the safe merging constraints it has to satisfy as described in the last subsection, it can solve problem (6) 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 in the single-lane merging problem [7], and is obviously more computationally intensive in the multi-lane merging problem, since a CAV may have to satisfy two safe-merging constraints. Therefore, we will employ the joint optimal control and barrier function (OCBF) controller developed in [13] to account for all constraints.

We begin by noting that the distances from O_2, O_3, O_4 to M_2 or M_3 are all the same, while the distances from O_1, O_2 to $M_{i,1}$ or M_4 (or from O_1, O_3 to M_4) are different since the lane change behavior will induce an extra l distance (a CAV moving from M_2 to M_4 is equivalent to a lane change). Therefore, we need to perform a coordinate transformation for those CAVs that are in different lanes (e.g., l_2 and l_1) and will merge into the same lane (e.g., l_1). In other words, when $i \in S_1(t)$ obtains information for $j \in S_1(t)$ from queue 1, the position information $x_j(t)$ is transformed by (using the *original lane* information in Fig. 2):

$$x_j(t) := \begin{cases} x_j(t) + l, & \text{if } [i \text{ in } l_2 \text{ or } l_3] \ \& \ [i-1 \text{ in } l_1], \\ x_j(t) - l, & \text{if } [i \text{ in } l_1] \ \& \ [i-1 \text{ in } l_2 \text{ or } l_3], \\ x_j(t), & \text{Otherwise.} \end{cases} \tag{10}$$

Next, we briefly review the OCBF approach in [13] as it applies to our problem. Problem (6) was solved in [7] for the single-lane merging problem and no noise in (1) and the *unconstrained* solution gives the following optimal control, speed, and position trajectories:

$$u_i^*(t) = a_i t + b_i \tag{11}$$

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

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

where a_i, b_i, c_i and d_i are constants that can be solved along with t_i^m by the initial and terminal constraints [7].

The OCBF controller aims to track the OC solution (11)-(13) while satisfying all constraints (3), (5) and (4). To accomplish this, first let $\boldsymbol{x}_i(t) \equiv (x_i(t), v_i(t))$. Referring to the vehicle dynamics (1), let $f(\boldsymbol{x}_i(t)) = [x_i(t), 0]^T$ and $g(\boldsymbol{x}_i(t)) = [0, 1]^T$. Each of the seven constraints in (3), (5) and (4) can be expressed as $b_k(\boldsymbol{x}_i(t)) \geq 0$, $k \in \{1, \ldots, 7\}$ where each $b_k(\boldsymbol{x}_i(t))$ is a CBF. For example, we have $b_1(\boldsymbol{x}_i(t)) = z_{i,i_p}(t) - \varphi v_i(t) - \delta$ for the rear-end safety constraint (3). As an alternative, a Control Lyapunov Function (CLF) [10] $V(\boldsymbol{x}_i(t))$ can also be used to track (stabilize) the optimal speed trajectory (12) through a CLF constraint. Therefore, the OCBF controller solves the following problem:

$$\min_{u_i(t)} \int_{t_i^0}^{t_i^m} \left(\frac{1}{2} (u_i(t) - u_{ref}(t))^2 \right) dt, \tag{14}$$

subject to the vehicle dynamics (1), the CBF constraints for (3), (5) and (4) and the CLF constraint for the speed tracking.

TABLE I

COMPARISON OF OC, CBF AND OCBF (WITH NOISE)

Method	α	Noise	Ave. time(s)	Ave. $\frac{1}{2}u_i^2(t)$	Ave. obj.
CBF	N/A	no	14.7539	19.7241	N/A
Vissim	0.01	N/A	31.5351	17.0415	19.2993
OCBF		no	**22.6763**	**6.7674**	**8.4458**
		yes	22.7636	8.8133	10.4780
Vissim	0.25	N/A	31.5351	17.0415	73.4767
OCBF		no	**16.1588**	**9.6914**	**38.3694**
		yes	16.1811	11.2944	39.6146
Vissim	0.40	N/A	31.5351	17.0415	107.3404
OCBF		no	**14.4820**	**14.6545**	**53.3915**
		yes	14.4996	16.4412	54.5177

The obvious selection for speed and acceleration reference signals is $v_{ref}(t) = v_i^*(t)$, $u_{ref}(t) = u_i^*(t)$, but we select $v_{ref}(t) = \frac{x_i^*(t)}{x_i(t)} v_i^*(t), u_{ref}(t) = \frac{x_i^*(t)}{x_i(t)} u_i^*(t)$ so as to provide position feedback to automatically reduce (or eliminate) the tracking position error, since the optimal solutions in (11)-(13) depend on the position (alternative forms of $v_{ref}(t)$, $u_{ref}(t)$ are possible as shown in [13]). The OCBF control can also deal with constraint violation due to noise in the dynamics included in (1) [13].

Remark (Framework Generalization). We can generalize the framework of any traffic scenario that involves multiple lanes leading to conflict zones beyond the merging configuration of Fig. 1. For example, in a 4-way intersection with one lane in each road, the number of possible cases is 56, but a CAV can easily find all the safe merging constraints (at most 5) that it needs to satisfy by looking up the extended queue similar to Table 2.

IV. SIMULATION RESULTS

All controllers have been implemented using MATLAB and we have used the Vissim microscopic multi-model traffic flow simulation tool as a baseline for the purpose of making comparisons between our controllers and human-driven vehicles adopting standard car-following models used in Vissim. We used QUADPROG for solving QPs of the form (14) and ODE45 to integrate the vehicle dynamics.

Referring to Fig. 1, CAVs arrive according to Poisson processes with rates 2000 CAVs per hour and 1200 CAVs per hour for the main and merging roads, respectively. The initial speed $v_i(t_i^0)$ is also randomly generated with a uniform distribution over $[15m/s, 20m/s]$ at the origins O and O', respectively. The parameters for (14) and (1) are: $L_2 = 400m, L_3 = 407m, L_4 = 406.0622m, l = 0.9378m, \varphi = 1.8s, \delta = 0m, u_{max} = 3.924m/s^2, u_{min} = -5.886m/s^2, v_{max} = 30m/s, v_{min} = 0m/s$. We consider uniformly distributed noise processes (in [-2, 2] m/s for $w_{i,1}(t)$ and in [-0.05, 0.05] m/s^2 for $w_{i,2}(t)$).

We compare the simulation results between Vissim (human driver), the CBF method [12] (by setting $u_{ref}(t) = 0$ and $v_{ref}(t) = v_{max}$ in (14)) and the OCBF method, as shown in Table I. The CBF method is aggressive in travel time, and thus has larger energy consumption than both the OCBF method and human drivers. The OCBF method does better in both metrics than human drivers in Vissim, and achieves about 50% improvement in the objective function (6) under all three different trade-off parameters α (recall that α trades off travel time and energy in (6)).

V. CONCLUSIONS

We have shown how to transform a multi-lane merging problem into a decentralized optimal control problem, and combine OC with the CBF method to solve the merging problem for CAVs in order to deal with cases where the OC solution becomes difficult to obtain, as well as to handle the presence of noise in the vehicle dynamics by exploiting the ability of CBFs to add robustness to an OC controller. Remaining challenges include research on resequencing and extensions to large traffic networks.

REFERENCES

[1] B. Schrank, B. Eisele, T. Lomax, and J. Bak. (2015) The 2015 urban mobility scorecard. Texas A&M Transportation Institute. [Online]. Available: http://mobility.tamu.edu

[2] M. Tideman, M. van der Voort, B. van Arem, and F. Tillema, "A review of lateral driver support systems," in *Proc. IEEE Intelligent Transportation Systems Conference*, pp. 992–999, Seatle, 2007.

[3] D. D. Waard, C. Dijksterhuis, and K. A. Broohuis, "Merging into heavy motorway traffic by young and elderly drivers," *Accident Analysis and Prevention*, vol. 41, no. 3, pp. 588–597, 2009.

[4] M. Mukai, H. Natori, and M. Fujita, "Model predictive control with a mixed integer programming for merging path generation on motor way," in *Proc. IEEE Conference on Control Technology and Applications*, pp. 2214–2219, Mauna Lani, 2017.

[5] V. Milanes, J. Godoy, J. Villagra, and J. Perez, "Automated on-ramp merging system for congested traffic situations," *Trans. on Intelligent Transportation Systems*, vol. 12, no. 2, pp. 500–508, 2011.

[6] G. Domingues, J. Cabral, J. Mota, P. Pontes, Z. Kokkinogenis, and R. J. F. Rossetti, "Traffic simulation of lane-merging of autonomous vehicles in the context of platooning," in *IEEE International Smart Cities Conference*, 2018, pp. 1–6.

[7] W. Xiao and C. G. Cassandras, "Decentralized optimal merging control for connected and automated vehicles," in *Proc. of the American Control Conference*, 2019, pp. 3315–3320.

[8] W. Cao, M. Mukai, and T. Kawabe, "Cooperative vehicle path generation during merging using model predictive control with real-time optimization," *Control Eng. Practice*, vol. 34, pp. 98–105, 2015.

[9] I. A. Ntousakis, I. K. Nikolos, and M. Papageorgiou, "Optimal vehicle trajectory planning in the context of cooperative merging on highways," *Transportation Research C*, vol. 71, pp. 464–488, 2016.

[10] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *Proc. of the European Control Conference*, 2019, pp. 3420–3431.

[11] W. Xiao and C. Belta, "Control barrier functions for systems with high relative degree," in *Proc. of 58th IEEE Conference on Decision and Control*, Nice, France, 2019, pp. 474–479.

[12] W. Xiao, C. Belta, and C. G. Cassandras, "Decentralized merging control in traffic networks: A control barrier function approach," in *Proc. ACM/IEEE International Conference on Cyber-Physical Systems*, Montreal, Canada, 2019, pp. 270–279.

[13] 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 22nd Intelligent Transportation Systems Conference*, 2019, pp. 3162–3167.

[14] Y. J. Zhang and C. G. Cassandras, "A decentralized optimal control framework for connected automated vehicles at urban intersections with dynamic resequencing," in *Proc. 57th IEEE Conference on Decision and Control*, pp. 217–222, Miami, 2018.

[15] H. Xu, Y. Zhang, L. Li, and W. Li, "Cooperative driving at unsignalized intersections using tree search," *preprint arXiv:1902.01024*, 2019.

[16] W. Xiao and C. G. Cassandras, "Decentralized optimal merging control for connected and automated vehicles with optimal dynamic resequencing," in *Proc. of the American Control Conference*, 2020, to appear.

[17] K. Vogel, "A comparison of headway and time to collision as safety indicators," *Accident Analysis & Prevention*, vol. 35, no. 3, pp. 427–433, 2003.