2019 IEEE Intelligent Transportation Systems Conference (ITSC)
Auckland, NZ, October 27-30, 2019

Decentralized Merging Control in Traffic Networks with Noisy Vehicle Dynamics: a Joint Optimal Control and Barrier Function Approach

Wei Xiao, Christos G. Cassandras and Calin Belta

Abstract— **We address the problem of optimally controlling Connected and Automated Vehicles (CAVs) arriving from two roads at a merging point where the objective is to jointly minimize the travel time and energy consumption of each CAV subject to a speed-dependent safety constraint and to speed and acceleration constraints. Implementing the decentralized solution to this problem obtained in prior work is limited by the computational cost when constraints become active on an optimal CAV trajectory and by the presence of noise in the vehicle dynamics. In this paper, we combine the unconstrained optimal control solution (treated as a reference trajectory for each CAV) with control barrier functions (CBFs) that guarantee the satisfaction of all constraints and provide robustness to noise. To accomplish this, we design a joint optimal control and barrier function (OCBF) controller where a CBF-based controller tracks the optimal control trajectory for each CAV in the presence of noise. In addition, when considering more complex objective functions for which analytical optimal control solutions are unavailable, we adapt the CBF method to such objectives. Simulation examples are included to compare the performance of the OCBF controller to optimal solutions (when available) and to a baseline provided by human-driven vehicles with results showing significant improvements in both 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 [9], [11], [14]. Advances in next-generation transportation system technologies and the emergence of Connected and Automated Vehicles (CAVs), also known as self-driving cars or autonomous vehicles, 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. One of the very early efforts exploiting the benefit of CAVs was proposed in [5], where an optimal linear feedback regulator was introduced for the merging problem to control a single string of vehicles. An overview of automated vehicle-highway systems was provided in [12].

In our recent work [18] 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 [17], in which case the computational cost depends on the number of constraints, some of which may become recursively active. Thus, we have found this to get 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 precluded nonlinearities.

To address the limitations above, one can adopt on-line control methods such as Model Predictive Control (MPC) or the Control Barrier Function (CBF) method. In MPC (e.g., [3], [7], [8]) time is normally discretized and an optimization problem is solved at each time instant with the addition of appropriate inequality constraints; then the system dynamics are updated. Since both control and state variables are considered as the decision variables in the optimization problem, MPC is very effective for problems with simple (usually linear or linearized) dynamics, objectives and constraints. The CBF approach [10], [2], [15] can overcome some shortcomings of MPC. Unlike MPC, the CBF method does not use states as decision variables in its optimization; instead, any continously differentiable state constraint is mapped onto a new constraint on the control input and can ensure forward invariance of the associated set, i.e., a control input that satisfies this new constraint is guaranteed to also satisfy the original constraint. This allows the CBF method to be effective for complex objectives, nonlinear dynamics, and constraints. We have adopted this approach for the merging problem in recent work [16] and shown that it provides good approximations of the analytically obtained OC solutions.

The contribution of this paper is to combine the OC and CBF methods when the OC solution is available, leading to a joint Optimal Control and Barrier Function (OCBF) method. This allows us to share all the advantages of OC and CBFs. In particular, when noise is present in the vehicle dynamics, the OCBF controller for a CAV tracks the optimal CAV trajectory and handles potential constraint violations. In addition, when the OC solution is unavailable, we show that the CBF method can still be employed and provide solutions which result in significantly better performance than that of human-driven vehicles.

II. PROBLEM FORMULATION

In this section, we first review the problem formulation from [18] so as to set the stage for the extensions to the merging problem which we address in this paper. The merging problem arises when traffic must be joined from two different roads, usually associated with a main lane and a merging lane as shown in Fig.1. We consider the case where

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

all traffic consists of CAVs randomly arriving at the two lanes joined at the Merging Point (MP) M where a collison may occur. The segment from the origin O or O' to the merging point M has a length L for both lanes, and is called the Control Zone (CZ). We assume that CAVs do not overtake each other in the CZ. A coordinator is associated with the MP whose function is to maintain a First-In-First-Out (FIFO) queue 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. The FIFO assumption imposed so that CAVs cross the MP in their order of arrival is made for simplicity and often to ensure fairness, but can be relaxed through dynamic resequencing schemes, e.g., as described in [19].

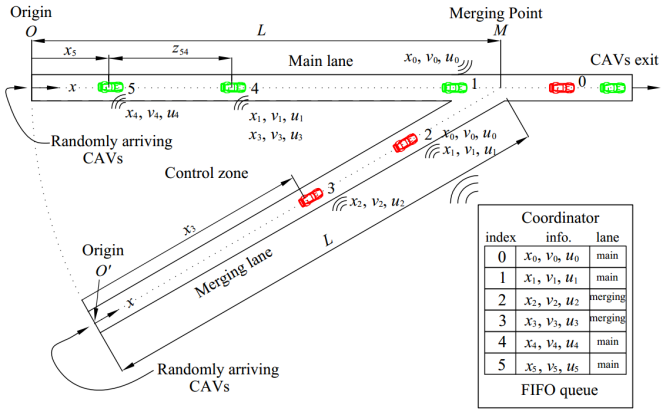


Fig. 1. The merging problem

Let $S(t)$ be the set of the FIFO-ordered indices of all CAVs located in the CZ at time t along with the CAV (whose index is 0 as shown in Fig.1) that has just left the CZ. Let $N(t)$ be the cardinality of $S(t)$. Thus, if a CAV arrives at time t, it is assigned the index $N(t)$. All CAV indices in $S(t)$ decrease by one when a CAV passes over the MP and the vehicle whose index is -1 is dropped.

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

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

where $x_i(t)$ denotes the distance to the origin O (O') along the main (merging) lane if the vehicle i is located in the main (merging) lane, $v_i(t)$ denotes the velocity, and $u_i(t)$ denotes the control input (acceleration). 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 $i \in S(t)$ arrives at the origin O or O' and the merging point M, 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 $i \in S(t)$ expressed as

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

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

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

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

where φ denotes the reaction time (as a rule, $\varphi = 1.8$ is used, e.g., [13]). 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): There should be enough safe space at the MP M for a merging CAV to cut in, i.e.,

$$z_{1,0}(t_1^m) \geq \varphi v_1(t_1^m) + \delta. \tag{4}$$

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

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

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

The 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:

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

Letting $\beta := \frac{\alpha \max\{u_{max}^2, u_{min}^2\}}{2(1-\alpha)}$, we obtain a simplified form:

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

where $\beta \geq 0$ denotes a weight factor that can be adjusted to penalize travel time relative to the energy cost.

Then, we have the following problem formulation:

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

While in [18], we assumed the absence of any noise in (1), in this paper we will include the possibility of system model uncertainties, errors due to signal transmission, as well as computation errors. Therefore, we add two noise terms in (1) to get

$$\begin{bmatrix} \dot{x}_i(t) \\ \dot{v}_i(t) \end{bmatrix} = \begin{bmatrix} v_i(t) + w_{i,1}(t) \\ u_i(t) + w_{i,2}(t) \end{bmatrix} \tag{8}$$

where $w_{i,1}(t), w_{i,2}(t)$ denote two random processes defined in an appropriate probability space.

III. Joint Optimal Control and Barrier Functions

In this section, we use the CBF method to track a CAV trajectory obtained through OC while taking advantage of the robustness to noise that the CBF approach can offer.

We need to distinguish between the following two cases:

- (i) $i_p = i-1$, *i.e.,* i_p is the CAV immediately preceding i in the FIFO queue (such as CAVs 3 and 5 in Fig. 1),
- (ii) $i_p < i-1$ (such as CAVs 2 and 4 in Fig.1), which implies CAV $i-1$ is in a different lane from i.

We can solve Problem 1 for all $i \in S(t)$ in a decentralized way, in the sense that CAV i can solve it using only its own local information (position, velocity and acceleration) along with that of its "neighbor" CAVs $i-1$ and i_p. Observe that if $i_p = i-1$, then (4) is a redundant constraint. Otherwise, we need to consider (3) and (4) independently.

A. Optimal Control Method

We briefly review the OC analysis in [18] so as to make our analysis as self-contained as possible. We consider (7) as our objective function and get the following unconstrained optimal control, speed, and position profiles:

$$u_i^*(t) = a_i t + b_i \tag{9}$$

$$v_i^*(t) = \frac{1}{2}a_i t^2 + b_i t + c_i \tag{10}$$

$$x_i^*(t) = \frac{1}{6}a_i t^3 + \frac{1}{2}b_i t^2 + c_i t + d_i \tag{11}$$

where a_i, b_i, c_i and d_i are integration constants. In case (i), a_i, b_i, c_i, d_i and t_i^m can be solved by five nonlinear algebraic equations, and these five algebraic equations become more complicated in case (ii).

Since we aim for the solution to Problem 1 to be obtained on-board each CAV, it is essential that the computational cost of solving these five algebraic equations for integration constants in (9)-(11) be minimal. If MATLAB is used, it takes less than 1 second to solve these five algebraic equations (Intel(R) Core(TM) i7-8700 CPU @ 3.2GHz 3.2GHz). On the other hand, when the constraints (3), (4), (5) become active, a complete OC solution can still be obtained [17], [6], but the computation time varies between 3 and 30 seconds depending on whether i_p is also safety-constrained or not. This motivates the use of CBFs, as reviewed next, to obtain sub-optimal but still feasible solutions with minimal computational effort.

B. Control Barrier Function Method

As shown in [16], the CBF method allows us to deal with nonlinear systems and to consider more complex objective functions than (7). In particular, we can consider:

$$\min_{u_i(t)} J_i(u_i(t)) := \beta(t_i^m - t_i^0) + \int_{t_i^0}^{t_i^m} f_v(t)dt, \tag{12}$$

where $f_v(t)$ represents a more practically realistic energy model. As an example, we have adopted in [16] the following fuel consumption model from [4], which describes fuel consumed per second as

$$\begin{aligned} f_v(t) &= f_{cruise}(t) + f_{accel}(t), \\ f_{cruise}(t) &= \omega_0 + \omega_1 v_i(t) + \omega_2 v_i^2(t) + \omega_3 v_i^3(t), \\ f_{accel}(t) &= (r_0 + r_1 v_i(t) + r_2 v_i^2(t))u_i(t). \end{aligned} \tag{13}$$

where ω_0, ω_1, ω_2, ω_3, r_0, r_1 and r_2 are positive coefficients (typical values are reported in [4]). It is assumed that during braking, i.e., $u_i(t) < 0$, no fuel is consumed. Note that (12) is hard to solve with OC analysis as in the previous section. However, in the CBF approach this can be done numerically.

In the CBF method, we do not explicitly optimize the travel time shown in (12). Instead, we use a control Lyapunov function (CLF) [1] to drive $v_i(t)$ to a desired speed such that the travel time is optimized. In [16], we define an output $y_i(t) := v_i(t) - v_{max}$, and choose a CLF $V(y_i(t)) = y_i^2(t)$. Any control input $u_i(t)$ should satisfy, for all $t \in [t_i^0, t_i^m]$,

$$L_f V(y_i(t)) + L_g V(y_i(t))u_i(t) + \epsilon V(y_i(t)) \leq \delta_i(t) \tag{14}$$

where $\epsilon > 0$ and $\delta_i(t)$ is a relaxation variable [1] that makes the requirement $v_i(t) = v_{max}$ to be treated as a soft constraint. Thus, we seek to achieve Objective 1 indirectly and consider Objective 2 directly, replacing (12) by:

$$\min_{u_i(t),\delta_i(t)} J_i(u_i(t), \delta_i(t)) := \int_{t_i^0}^{t_i^m} f_v(t) + \beta\delta_i^2(t)dt \tag{15}$$

Since Constraints 1, 2 and part of 3 do not involve the control input, we use CBFs to find new constraints on the control input as detailed in [16]. When these new constraints are satisified, Constraints 1-3 are also guranteed to be satisified. We use the Quadratic Programing (QP) method following from [2] to solve (15), i.e., we transform (15) into a sequence of QPs by partitioning the time interval $[t_i^0, t_i^m]$ into equal time intervals of length Δt: $\{[t_i^0 + k\Delta t, t_i^0 + (k+1)\Delta t)\}, k = 0, 1, 2, \ldots$ At each time step defined by $t = t_i^0 + k\Delta t$, we solve the QP where the decision variables are the control $u_i(t)$ (fixed over the time step t) and the CLF relaxation $\delta_i(t)$, thus obtaining an optimal control $u_i^*(t)$. Then, over interval $[t_i^0 + k\Delta t, t_i^0 + (k+1)\Delta t)$, we update the system dynamics (1) using $u_i^*(t)$. Thus, all CAVs can safely pass over the merging point M while minimizing $J_i(u_i(t), \delta_i(t))$ within each time interval, thus jointly minimizing the energy consumption captured by $f_v(t)$ and travel time (indirectly) through the minimization of δ_i^2. By adjusting the weight β in (15), we can trade off between these two objectives.

C. Joint Optimal Control and Barrier Function Method

Suppose that an OC solution is available for the objective (7), obtained through (9)-(11). Our goal here is to combine this OC solution with a CBF-based controller whose goal is to track the former as closely as possible. Thus, instead of seeking to drive $v_i(t)$ to v_{max} as in the previous section, we aim to track the optimal speed $v_i^*(t)$ obtained through (9)-(11). In particular, we define a controller aiming to drive $v_i(t)$ to $v_{ref}(t)$ where

$$v_{ref}(t) = \frac{x_i^*(t)}{x_i(t)} v_i^*(t) \tag{16}$$

and $x_i^*(t), v_i^*(t)$ are the unconstrained optimal position and speed profiles from (11) and (10). If $x_i(t) > x_i^*(t)$, then $v_{ref}(t) < v_i^*(t)$, thus automatically reducing (or eliminating) the tracking position error. This approach can also address the practical fact, observed in [16], that seeking to drive $v_i(t)$ to v_{max} is overly "aggressive" in reducing travel time in (7) at the expense of the energy component (even with small values of the weight β in (7)).

An alternative form for $v_{ref}(t)$ is

$$v_{ref}(t) = e^{\frac{(x_i^*(t)-x_i(t))}{\sigma}} v_i^*(t) \tag{17}$$

where $\sigma > 0$. While $v_{ref}(t)$ in (16) depends heavily on the exact value of $x_i(t)$, an advantage of (17) is that it allows $v_{ref}(t)$ to depend only on the position error.

Using either form of $v_{ref}(t)$, we can now proceed as in (14) and define an output $y_i(t) := v_i(t) - v_{ref}(t)$ and a CLF $V(y_i(t)) = y_i^2(t)$. Then, any control input $u_i(t)$ should satisfy, for all $t \in [t_i^0, t_i^m]$,

$$L_f V(y_i(t)) + L_g V(y_i(t)) u_i(t) + \epsilon V(y_i(t)) \leq \delta_i(t) \tag{18}$$

where $\delta_i(t)$ is again a relaxation variable that makes the requirement $v_i(t) = v_{ref}(t)$ to be treated as a soft constraint.

Along the same lines, we now seek a control input $u_i(t)$ in the CBF method which tracks the unconstrained optimal control $u_i^*(t)$ through a CBF controller aiming to drive $u_i(t)$ to $u_{ref}(t)$ defined by

$$u_{ref}(t) = \frac{x_i^*(t)}{x_i(t)} u_i^*(t) \tag{19}$$

where $u_i^*(t)$ is the unconstrained optimal control from (9). An alternative, similar to (16), is to define $u_{ref}(t)$ as

$$u_{ref}(t) = e^{\frac{(x_i^*(t)-x_i(t))}{\sigma}} u_i^*(t). \tag{20}$$

Applying the CBF approach as described in Section III-B, we replace the travel time in (7) by the slack variable $\delta_i(t)$ in (18) and consider the objective function:

$$\min_{u_i(t),\delta_i(t)} J_i(u_i(t),\delta_i(t)) = \int_{t_i^0}^{t_i^m} \left(\beta \delta_i^2(t) + \frac{1}{2}(u_i(t) - u_{ref}(t))^2 \right) dt, \tag{21}$$

subject to (8), (3), (4), (5) and (18), the initial and terminal conditions $x_i(t_i^0) = 0$, $x_i(t_i^m) = L$, and given $t_i^0, v_i(t_i^0)$. Thus, we have combined the CBF method and the OC solution by using (19) or (20) to link the optimal position and acceleration to $u_{ref}(t)$, and use (16) or (17) in the CLF $(v_i(t) - v_{ref}(t))^2$ to combine with (21). We refer to the resulting control $u_i(t)$ in (21) as the OCBF control. As described in Section III-B, in order to solve (21) we once again transform all the constraints (3), (4), (5) to constraints on the control only, then we discretize time and solve a QP over each time step with $u_i(t), \delta_i(t)$ as decision variables. The optimal solution, $u_i^*(t)$, for time step t is used to update (8) over the corresponding time interval.

Note that in solving (21) we can apply either (16) or (19) (similarly, (17) or (20)), or we can apply both (16) and (19). As illustrated in our simulation examples in Section IV, the use of only (16), yields an OCBF control which is Lipschitz continous, whereas using both improves performace. For simplicity, we can also set $v_{ref}(t) = v_i^*(t)$ and $u_{ref}(t) = u_i^*(t)$ instead of (16) and (19) ((17) and (20)).

D. Constraint Violation Due to Noise

In this section, we consider the noise model (8). In the presence of noise in the dynamics, constraints 1, 2, 3 may be temporarily violated, which prevents the CBF method from satisfying the forward invariance property [2], [15] if any of these constraints is initially violated. Therefore, we need to find a way to ensure that Constraints 1, 2, 3 are satisfied again in finite time.

Suppose we have a constraint $h_i(\boldsymbol{x}_i(t)) \geq 0$ for vehicle $i \in S(t)$, and this constraint is violated at time t_1 due to the noise, i.e., we have $h_i(\boldsymbol{x}_i(t_1)) < 0$. We need to make sure that $h_i(\boldsymbol{x}_i(t))$ is strictly increasing after time t_1, i.e., $\dot{h}_i(\boldsymbol{x}_i(t)) \geq c$, where $c > 0$. Using Lie derivatives, we evaluate the change in $h_i(\boldsymbol{x}_i(t))$ along the flow defined by the system state vector. Then, any control $u_i(t)$ must satisfy

$$L_f h_i(\boldsymbol{x}_i(t)) + L_g h_i(\boldsymbol{x}_i(t)) u_i(t) \geq c \tag{22}$$

where c should be large enough so that $h_i(\boldsymbol{x}_i(t))$ is strictly increasing even if the system is subject to the worst possible noise case. For this reason, in what follows we assume that the random processes $w_{i,1}(t), w_{i,2}(t)$ in (8) are characterized by probability density functions with finite support, hence there exist upper and lower bounds we can use in determining an appropriate value for c.

Note that several constraints may be violated at the same time. Starting from t_1, we apply the constraint (22) to the CBF optimizer instead of the CBF constraint, and $h_i(\boldsymbol{x}_i(t))$ will be positive again in finite time since it is strictly increasing. When $h_i(\boldsymbol{x}_i(t))$ becomes positive again at t_2, we can once again apply the CBF constraint.

IV. SIMULATION RESULTS

All controllers in this section 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 (21) or (15) and ODE45 to integrate the vehicle dynamics.

Referring to Fig. 1, CAVs arrive according to Poisson processes with arrival rates that we allow to vary in our simulation examples. The initial speed $v_i(t_i^0)$ is also randomly generated with uniform distribution in $[15m/s, 20m/s]$ at the origins O and O', respectively. The parameters for (21) or (15) and (8) are: $L = 400m, \varphi = 1.8s, \delta = 0m, u_{max} = 3.924m/s^2, u_{min} = -3.924m/s^2, v_{max} = 30m/s, v_{min} = 0m/s, \beta = 1, \epsilon = 10, \Delta t = 0.1s, c = 1$, and we consider uniformly distributed noise processes (in [-2, 2] for $w_{i,1}(t)$ and in [-0.2, 0.2] for $w_{i,2}(t)$) for all simulations.

1. OCBF implementation example. First, we provide a simple example of the OCBF controller implementation for a single vehicle which considers (21) as the objective function.

TABLE I

OBJECTIVE FUNCTION COMPARISON WITHOUT NOISE

Items	OC	OCBF			
Track		(16)	(17)	(17)	(16), (19)
σ			4	40	
time (s)	15.01	15.07	15.01	15.01	15.01
$\frac{1}{2}u_i^2(t)$	4.4400	4.4129	4.6962	4.6674	4.4403
objective	33.3356	33.4357	33.5252	33.5039	33.3358

The initial parameters are $t_i^0 = 0s$, $v_i^0 = 20m/s$, $\alpha = 0.26$. If we only apply (16) or (17), set $u_{ref}(t) = 0$ and assume no noise, then we obtain the control profiles shown in Fig. 2(a). The speed reference form (17) tends to achieve a closer track of the OC control (black curve) compared to the form (16) at the expense of larger over-shot; as a result, performnace is worse as shown in Table I (values in red are the best).

If we apply both (16) and (19) without noise, we obtain the control profiles shown in Fig. 2(b) where the OCBF controller's performance is virtually indistinguishable from that of the OC control, as shown in Table I.

With noise added (based on uniform distribution in [-2, 2] for $w_{i,1}(t)$ and in [-0.2, 0.2] for $w_{i,2}(t)$), we show the control profiles under different noise levels in Fig. 2(c) with (16) and (19); and in Fig. 2(d) with (17) and (20). Constraints 1-3 may be temporarily violated but will be forced to be satisfied again in finite time through constraint (22). The speed and control tracking forms (16) and (19) perform better than (17) and (20) as noise increases.

2. Comparison of OC control from [18], CBF control from [16], and OCBF control in this paper. Consider the merging problem with the simple objective function (7) for which we can easily get unconstrained optimal solutions. Then, we employ the CBF method and the OCBF technique (with (16) and (19)) introduced in Sec. III-C. Simulation results under four different trade-off parameters are shown in Table II. We can see that the OCBF method achieves comparable results to OC, even in the presence of noise.

The computation time in MATLAB with the OCBF method for each i at each step is less than 0.01s (Intel(R) Core(TM) i7-8700 CPU @ 3.2GHz 3.2GHz), while the OC method takes between $1s$ and $30s$ for each CAV, depending on whether the constraints are active or not.

We also show in Fig. 3(a) how the travel time and energy consumption vary as the weight factor α in (6) changes, and similarly for the objective function in Fig. 3(b). The significance of Fig. 3(b) is to show how well the OCBF can match the optimal performance obtained through OC.

3. Comparison of CBF control from [16], CBF control with objective (12) in this paper, and human-driven vehicles through Vissim. We now consider the objective function (12) which is too complex to allow the derivation of an OC solution. Thus, we transform (12) into (15) and select a value $\beta = 0.2$ for the weight β in (15) through trial and error to best match the performance in Vissim. We vary the relative traffic arrival rates of the main and merging lane and show our results in Tables III, IV, V.

TABLE II

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	14.6978	26.9178	N/A
OC	0.01	no	**25.4291**	**0.1725**	**2.1288**
OCBF		no	25.6879	1.0582	3.0256
		yes	25.7494	2.2373	4.1976
OC	0.25	no	**17.0472**	**4.9069**	**36.4909**
OCBF		no	17.1176	5.5569	37.1139
		yes	17.1396	6.8959	38.1605
OC	0.40	no	**15.1713**	**10.6508**	**53.1120**
OCBF		no	15.2286	11.3629	53.7157
		yes	15.2527	12.7671	54.6325
OC	0.60	no	**13.1035**	**24.4079**	**70.2922**
OCBF		no	13.1560	25.2468	70.8720
		yes	13.1692	26.6534	71.4938

TABLE III

MAIN LANE ARRIVAL RATE : MERGING LANE ARRIVAL RATE = 1:1

Items	CBF-(7) [16]	CBF-(12)	Vissim
Ave. time(s)	**14.6978**	18.1549	25.0813
Main time(s)	**14.7000**	18.1717	17.9935
Merg. time(s)	**14.6956**	18.1378	32.3267
Ave. fuel(mL)	57.9532	**30.9813**	36.9954
Main fuel(mL)	57.7028	**30.8856**	42.6925
Merg. fuel(mL)	58.2092	**31.0791**	31.1717

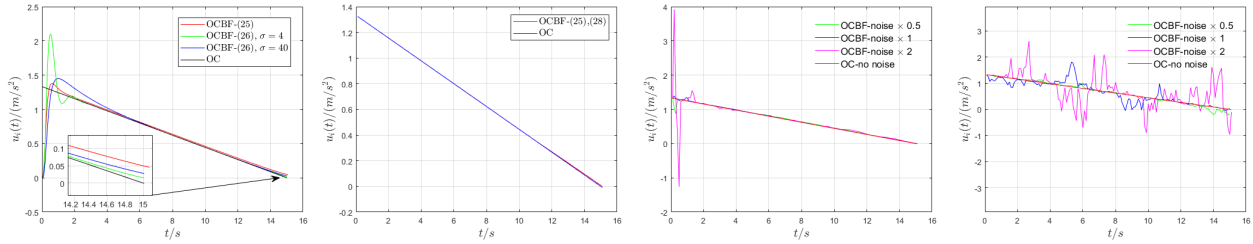
In Tables III and IV, note that both CBF methods outperform human-driven vehicles modeled though Vissim. We also observe that the CBF method developed in this paper using (12) is vastly superior to that of [16] in the energy component with little loss in travel time performance. We also note that without any control (as in Vissim), the main lane vehicles have priority over the merging lane and the merging lane vehicles may even stop before the merging point. Thus, there is heavy congestion in the merging lane when the ratio between the main lane and Merging lane arrival rates is 1:3.

We observe in Table V that the energy consumption vehicles in Vissim is significantly lower compared to the CBF methods. This is due to the fact that the merging lane vehicles frequently stop before the merging point M, thus having low speeds when passing over M. In order to achieve a fair comparison, we consider a longer time horizon over which we measure fuel consumption and travel time. This is accomplished by extending the trip of each vehicle for an additional length L beyond the merging point M, as shown in Table VI. As expected, the energy performance under CBF control is now significantly better (by about 37%) than that of human-driven vehicles.

TABLE IV

MAIN LANE ARRIVAL RATE : MERGING LANE ARRIVAL RATE = 3:1

Items	CBF-(7) [16]	CBF-(12)	Vissim
Ave. time(s)	**14.6578**	18.1189	23.9300
Main time(s)	**14.6794**	18.1413	18.3476
Merg. time(s)	**14.6074**	18.0667	36.9556
Ave. fuel(mL)	60.2624	**31.9754**	39.8587
Main fuel(mL)	61.0934	**32.7556**	42.8554
Merg. fuel(mL)	58.3235	**30.1549**	32.8666



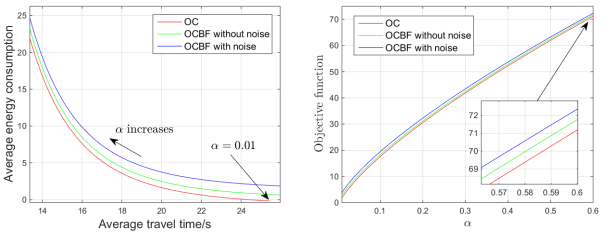
(a) The control profiles with only speed tracking (16) or (17).

(b) The controls with speed tracking (16) and control tracking (19).

(c) The controls with both (16) and (19) under different noise levels.

(d) The controls with (17) and (20), $\sigma = 40$ under different noise levels.

Fig. 2. OCBF implementation examples under different tracking equations and noise levels.



(a) Travel time and energy consumption as the factor α changes.

(b) Objective function as the weight factor α changes.

Fig. 3. Performance metrics variation as the weight factor α changes.

TABLE V

MAIN LANE ARRIVAL RATE : MERGING LANE ARRIVAL RATE = 1:3

Items	CBF-(7) [16]	CBF-(12)	Vissim
Ave. time(s)	**14.6000**	18.0093	29.2035
Main time(s)	**14.7133**	18.1133	17.8667
Merg. time(s)	**14.5761**	17.9873	31.5986
Ave. fuel(mL)	61.1607	33.4848	**30.5212**
Main fuel(mL)	57.3805	**30.9263**	46.5004
Merg. fuel(mL)	61.9593	34.0253	**27.1454**

V. CONCLUSIONS

We have shown how to combine the OC and the CBF method to solve the merging problem for CAVs in order to deal with cases where the OC solution becomes computationally costly, as well as to handle the presence of noise in the vehicle dynamics by exploiting the ability of control barrier functions to add some robustness to an OC controller. In addition, when considering more complex objective functions for which analytical optimal control solutions are unavailable, we have adapted the CBF method to such objectives. Remaining challenges include the proper selection of the weight factor that trades off time and energy and the use of time-varying steps in implementing the OCBF method.

TABLE VI

RATE = 1:3, ADDING A LANE OF LENGTH L AFTER THE MERGING POINT.

Items	CBF-(7) [16]	CBF-(12)	Vissim
Ave. time(s)	**28.7975**	36.3076	50.9987
Main time(s)	**28.9857**	36.3786	38.8643
Merg. time(s)	**28.7569**	36.2923	53.6123
Ave. fuel(mL)	88.2784	**51.6414**	81.6633
Main fuel(mL)	86.6246	**48.7578**	77.8110
Merg. fuel(mL)	88.6347	**52.2625**	82.4930

REFERENCES

[1] Aaron D. Ames, K. Galloway, and J. W. Grizzle. Control lyapunov functions and hybrid zero dynamics. In *Proc. 51rd IEEE Conference on Decision and Control*, pp. 6837–6842, 2012.

[2] Aaron D. Ames, Jessy W. Grizzle, and Paulo Tabuada. Control barrier function based quadratic programs with application to adaptive cruise control. In *Proc. IEEE Conf. on Decision and Control*, 2014.

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

[4] M. Kamal, M. Mukai, J. Murata, and T. Kawabe. Model predictive control of vehicles on urban roads for improved fuel economy. *IEEE Transactions on Control Systems Technology*, 21(3):831–841, 2013.

[5] W. Levine and M. Athans. On the optimal error regulation of a string of moving vehicles. *IEEE Transactions on Automatic Control*, 11(13):355–361, 1966.

[6] A. A. Malikopoulos, C. G. Cassandras, and Y. J. Zhang. A decentralized energy-optimal control framework for connected and automated vehicles at signal-free intersections. *Automatica*, 93(7):244–256, 2018.

[7] 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.

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

[9] B. Schrank, B. Eisele, T. Lomax, and J. Bak. The 2015 urban mobility scorecard. Texas A&M Transportation Institute, 2015.

[10] Keng Peng Tee, Shuzhi Sam Ge, and Eng Hock Tay. Barrier lyapunov functions for the control of output-constrained nonlinear systems. *Automatica*, 45(4):918–927, 2009.

[11] M. Tideman, M.C. 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.

[12] P. Varaiya. Smart cars on smart roads: problems of control. *IEEE Transactions on Automatic Control*, 38(2):195–207, 1993.

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

[14] D. De Waard, C. Dijksterhuis, and K. A. Broohuis. Merging into heavy motorway traffic by young and elderly drivers. *Accident Analysis and Prevention*, 41(3):588–597, 2009.

[15] Wei Xiao and Calin Belta. Control barrier functions for systems with high relative degree. *preprint arXiv:1903.04706*, 2019.

[16] Wei Xiao, Calin Belta, and Christos G. Cassandras. Decentralized merging control in traffic networks: A control barrier function approach. In *Proc. ACM/IEEE International Conference on Cyber-Physical Systems*, pp. 270–279, 2019.

[17] Wei Xiao and Christos G. Cassandras. Decentralized optimal merging control for connected and automated vehicles. *preprint arXiv:1809.07916*, 2018.

[18] Wei Xiao and Christos G. Cassandras. Decentralized optimal merging control for connected and automated vehicles. In *Proc. of the American Control Conference*, 2019.

[19] Yue 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.