Safety-Critical Optimal Control for Autonomous Systems*

XIAO Wei · CASSANDRAS G. Christos · BELTA Calin

DOI: 10.1007/s11424-021-1230-x
Received: 29 June 2021

Abstract This paper presents an overview of the state of the art for safety-critical optimal control of autonomous systems. Optimal control methods are well studied, but become computationally infeasible for real-time applications when there are multiple hard safety constraints involved. To guarantee such safety constraints, it has been shown that optimizing quadratic costs while stabilizing affine control systems to desired (sets of) states subject to state and control constraints can be reduced to a sequence of Quadratic Programs (QPs) by using Control Barrier Functions (CBFs) and Control Lyapunov Functions (CLFs). The CBF method is computationally efficient, and can easily guarantee the satisfaction of nonlinear constraints for nonlinear systems, but its wide applicability still faces several challenges. First, safety is hard to guarantee for systems with high relative degree, and the above mentioned QPs can easily be infeasible if tight or time-varying control bounds are involved. The resulting solution is also sub-optimal due to its myopic solving approach. Finally, this method works conditioned on the system dynamics being accurately identified. The authors discuss recent solutions to these issues and then present a framework that combines Optimal Control with CBFs, hence termed OCBF, to obtain near-optimal solutions while guaranteeing safety constraints even in the presence of noisy dynamics. An application of the OCBF approach is included for autonomous vehicles in traffic networks.

Keywords Control barrier function, optimal control, safety.

1 Introduction

Optimizing a cost function associated with the operation of a dynamical system while also satisfying hard safety constraints at all times is a fundamental and challenging problem. The challenge is even greater when stabilizing some of the system state variables to desired values is an additional requirement. At the same time, the proliferation of autonomous systems implies

XIAO Wei · CASSANDRAS G. Christos · BELTA Calin
Division of Systems Engineering and Center for Information and Systems Engineering, Boston University, Brookline, MA 02446*, USA.* Email: xiaowei@bu.edu; cgc@bu.edu; cbelta@bu.edu.

*This work was supported in part by NSF under Grant Nos. ECCS-1931600, DMS-1664644, CNS-1645681, IIS-1723995, and IIS-2024606, by ARPAE Under Grant No. DE-AR0001282 and by Its NEXTCAR Program Under Grant DE-AR0000796, by AFOSR Under Grant No. FA9550-19-1-0158, and by the MathWorks and by NPRP Grant (12S-0228-190177) from the Qatar National Research Fund (a member of the Qatar Foundation).
◇ *This paper was recommended for publication by Guest Editor ZHANG Ji-Feng.*

the need to provide safety guarantees when operating in autonomous fashion. Safety-critical optimal control problems can usually be decomposed into *planning* and *execution* components. Optimality is central to the trajectory planning phase, while safety is the main consideration during real-time execution in complex environments with real system dynamics that may differ from those used in the planning phase. Typical problems of this type include autonomous driving[1, 2], space exploration[3] and manufacturing automation[4].

Traditional methods used for planning include Rapidly-exploring Randomized Trees (RRT)[5], A*[6] and their variations, originating mostly from the robotics field. These approaches do not usually consider system dynamics or optimality. Optimal control methods, such as Hamiltonian analysis and the calculus of variations[7], are also widely used in planning. These approaches usually consider simplified (normally linear) system dynamics in order to reduce the computational complexity of the two-point-boundary-value problems that they typically reduce to. Even with such simplifications, these problems are still hard to solve, particularly when multiple state and/or control constraints are involved.

When it comes to real-time execution, the PID method is the most widely used in all relevant fields. This approach is case-dependent and requires non-trivial extensive parameter tuning. Another popular method for execution is Model Predictive Control (MPC)[8–10]. This approach formulates a receding horizon control problem repeatedly solved over a sequence of discrete time steps, thus, computational complexity is the main concern, especially for nonlinear models, when computation resources are limited. Moreover, a receding horizon is not crucial when there is an optimal reference available from the planning stage. An alternative approach which is very computationally efficient is based on the use of Barrier Functions (BFs).

The traditional use of BFs is to include them in the cost function[11] treating them as "soft" constraints. In recent work, however, BFs are considered as explicit constraints in the optimization problem and these constraints are Lyapunov-like conditions[12, 13] whose use can be traced back to optimization methodologies[14]. More recently, they have been employed to prove set invariance[15–17] in formal verification and for multi-objective control[18]. In [12], it was proved that if a BF for a given set satisfies Lyapunov-like conditions, then the set is forward invariant, an important property in practice. A less restrictive form of a BF, which is allowed to decrease when far away from the boundary of the unsafe set, was proposed in [19]. Another approach that allows a BF to become zero (the safe set boundary) was proposed in [20]. This simpler form has also been considered in time-varying cases and applied to enforce Signal Temporal Logic (STL) formulas as hard constraints[21].

Control BFs (CBFs) are extensions of BFs for control systems and they are used to map a constraint defined over system *states* onto a constraint on the *control input*. Recently, it has been shown that, to stabilize an affine control system while optimizing a quadratic cost and satisfying state and control constraints, CBFs can be combined with Control Lyapunov Functions (CLFs)[22–25] to form quadratic programs (QPs)[19, 20, 26] which can be very efficiently solved in real time.

It has been recently shown[27] that planning and execution using the CBF method to provide safety guarantees can be combined to obtain solutions of the aforementioned CBF-based QP which closely track the optimal control. As a result, the real-time optimization problem always consists of a sequence of QPs even if the original cost function is not quadratic in control. In addition, we may also define mutiple CLFs to improve the way the system state tracks the optimal state reference. Noisy dynamics can also be included in this framework to achieve robust control by considering the noise bounds in the corresponding CBF constraints.

In this paper, we present an overview for safety-critical optimal control problems. Specifically, we will first overview the CBF method and then discuss current research challenges related to it. We will also present some recent results and ideas for addressing these challenges. We will then present a framework that combines optimal trajectory planning and safety-critical real-time control, by aiming to track a planned control while always guaranteeing safety. This framework combines Optimal Control with Barrier Functions and is, therefore, termed OCBF. Finally, we provide an application of the OCBF approach to the control of autonomous vehicles in traffic networks.

The paper is structured as follows. In Section 2, we review definitions and key results on the CBF method. We discuss research challenges for the CBF method and present some solutions in Section 3. In Section 4, we present the OCBF framework for safety-critical optimal control problems. In Section 5, we provide an application to autonomous vehicles in a traffic merging problem, illustrating its effectiveness through a simulation example. We conclude with Section 6.

2 Background

Definition 2.1 (Class $\mathcal{K}$ function[28]) A Lipschitz continuous function $\alpha : [0, a) \to [0, \infty), a > 0$ is said to belong to class $\mathcal{K}$ if it is strictly increasing and $\alpha(0) = 0$.

We consider affine control systems of the form

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}) + g(\boldsymbol{x})\boldsymbol{u}, \tag{1}$$

with $\boldsymbol{x} \in X \in \mathbb{R}^n$ ((X denotes the state constraint set)) and $f : \mathbb{R}^n \to \mathbb{R}^n$ $g : \mathbb{R}^n \to \mathbb{R}^{n\times q}$ globally Lipschitz, and $\boldsymbol{u} \in U \subset \mathbb{R}^q$ (U denotes the control constraint set). Solutions $\boldsymbol{x}(t)$ of (1), starting at $\boldsymbol{x}(t_0)$, $t \geq t_0$, are forward complete.

Definition 2.2 (Forward invariant set) A set $C \subset \mathbb{R}^n$ is forward invariant for the system (1) if its solutions starting at any $\boldsymbol{x}(t_0) \in C$ satisfy $\boldsymbol{x}(t) \in C$ for $\forall t \geq t_0$.

We are interested in a safety set C of the form

$$C := \{\boldsymbol{x} \in \mathbb{R}^n : b(\boldsymbol{x}) \geq 0\}, \tag{2}$$

where $b : \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable function.

Definition 2.3 (Control barrier function[19–21]) Given a set C as in Equation (2), $b(\boldsymbol{x})$ is a candidate control barrier function (CBF) for the system (1) if there exists a class $\mathcal{K}$ function α such that

$$\sup_{\boldsymbol{u}\in U}[L_f b(\boldsymbol{x}) + L_g b(\boldsymbol{x})\boldsymbol{u} + \alpha(b(\boldsymbol{x}))] \geq 0 \tag{3}$$

for all $\boldsymbol{x} \in C$, where L_f, L_g denote the Lie derivatives[†] along f and g, respectively.

We refer to the CBF in Definition 2.3 as a "candidate" CBF since $\alpha(\cdot)$ is undefined so that there may not exist a $\boldsymbol{u} \in U$ that satisfies (3). A CBF is defined when $\alpha(\cdot)$ is found in (3) so that u is within the control constraint set U .

Theorem 2.4 (see [20, 21]) *Given a CBF b with the associated set C from Equation* (2)*, any Lipschitz continuous controller $\boldsymbol{u}(t), \forall t \geq t_0$ that satisfies* (3) *renders the set C forward invariant for control system* (1)*.*

Definition 2.5 (Control Lyapunov function[25]) A continuously differentiable function $V : \mathbb{R}^n \to \mathbb{R}$ is a globally and exponentially stabilizing control Lyapunov function (CLF) for the system (1) if there exist constants $c_1 > 0, c_2 > 0, c_3 > 0$ and $c_1||\boldsymbol{x}||^2 \leq V(\boldsymbol{x}) \leq c_2||\boldsymbol{x}||^2$ such that, for $\forall \boldsymbol{x} \in \mathbb{R}^n$,

$$\inf_{\boldsymbol{u}\in U}[L_f V(\boldsymbol{x}) + L_g V(\boldsymbol{x})\boldsymbol{u} + c_3 V(\boldsymbol{x})] \leq 0. \tag{4}$$

Theorem 2.6 (see [25]) *Given a CLF V as in Definition* 2.5*, any Lipschitz continuous controller $\boldsymbol{u}(t), \forall t \geq t_0$ that satisfies* (4) *exponentially stabilizes system* (1) *to the origin.*

Definition 2.7 (Relative degree[28]) The relative degree of a (sufficiently) differentiable function $b : \mathbb{R}^n \to \mathbb{R}$ with respect to the system (1) is the number of times we need to differentiate it along the dynamics of (1) until the control $\boldsymbol{u}$ explicitly shows.

In this paper, since function b is used to define a constraint $b(\boldsymbol{x}) \geq 0$, we will also refer to the relative degree of b as the relative degree of the constraint.

Many existing works[19, 21, 29] combine CBFs and CLFs with quadratic costs to formulate optimization problems. The CLF constraint is always slacked (i.e., considered as a soft constraint) when combined with a CBF to make the problem feasible; however, state convergence may not be guaranteed. In other words, suppose we have a **safety-critical optimal control problem** with a cost $\int_{t_0}^{t_f} \boldsymbol{u}^T P \boldsymbol{u} dt$ (P is positive definite), a safety requirement $b(\boldsymbol{x}) \geq 0$ and state convergence captured by a CLF $V(\boldsymbol{x})$. Then, we have the following optimization problem:

$$\begin{aligned} \arg \min_{\boldsymbol{u}(t),\delta(t)} & \int_{t_0}^{t_f} [\boldsymbol{u}(t)^T P \boldsymbol{u}(t) + p\delta^2(t)]dt \\ \text{s.t.} \quad & \boldsymbol{u} \in U \text{ and} \\ & L_f b(\boldsymbol{x}) + L_g b(\boldsymbol{x})\boldsymbol{u} + \alpha(b(\boldsymbol{x})) \geq 0, \\ & L_f V(\boldsymbol{x}) + L_g V(\boldsymbol{x})\boldsymbol{u} + c_3 V(\boldsymbol{x}) \leq \delta, \end{aligned} \tag{5}$$

where $p > 0$, and δ is a relaxation that addresses the possible conflict between the CBF and CLF constraints. Time is discretized, and an optimization problem with constraints given by CBFs and CLFs is solved at each time step. In other words, we partition the time interval $[t_0, t_f]$ into a set of equal time intervals $\{[t_0, t_0 + \Delta t), [t_0 + \Delta t, t_0 + 2\Delta t), \cdots\}$, where $\Delta t > 0$. In each interval $[t_0 + \omega\Delta t, t_0 + (\omega + 1)\Delta t)$ ($\omega = 0, 1, \cdots$), we keep the state constant at its value at $t_0 + \omega\Delta t$, and reformulate the above optimization problem as a sequence of QPs. The optimal

[†] The Lie derivative of a function along a vector field captures the change in the value of the function along the vector field (see, e.g., [28]).

control obtained by solving this QP is applied at the current time step $t_0 + \omega\Delta t$ and held constant for the whole associated time interval $[t_0 + \omega\Delta t, t_0 + (\omega + 1)\Delta t)$. The dynamics (1) are updated and the procedure is repeated. When Δt is small, we achieve higher accuracy without the necessity to consider the inter-sampling effect; this comes at the expense of added computational complexity.

There are still several issues that define challenges for the application of the CBF-based method. First, safety is harder to guarantee for systems with high relative degrees, in which case we need to take multiple derivatives of a constraint in order to make the control show up in the derivative. Second, the problem can easily become infeasible when tight control bounds are involved at a certain time step due to the myopic nature of the solution method above, especially for noisy dynamics and time-varying control bounds. Third, the solution of each QP above is sub-optimal, since the QP is only solved pointwise. Last but not least, this approach heavily depends on the accuracy of the system dynamics used in (1), which could be very hard to ensure for systems that are hard to identify (such as time-varying systems) under limited computational resources. In the following sections, we will discuss how we may address these challenges.

3 Challenges in the Use of CBFs for Guaranteeing Safe State Trajectories

In the following sections, we discuss in more detail the main challenges facing the use of CBFs in providing safety constraint guarantees when executing real-time state trajectories that have been determined at the planning stage of a given problem involving an autonomous system. The resolution of these challenges presented in what follows is based either on recent research results or is the subject of ongoing research.

3.1 High-Order CBFs

In order to guarantee safety for constraints of high relative degree, a backstepping approach was introduced in [30], and it was shown to work for relative degree $m = 2$. A CBF method for position-based constraints with relative degree $m = 2$ was also proposed in [31]. A more general form, which works for arbitrarily high relative degree constraints, was proposed in [29, 32]. The method in [29] employs input-output linearization and finds a pole placement controller with negative poles to stabilize the barrier function to zero. The resulting barrier function is exponential. The authors in [33] proposed an approach to define another function that is with relative degree $m = 1$ from the original high-relative-degree constraint. This approach does not include all the states in the definition of a CBF, which may reduce the problem feasibility. Another more general approach for arbitrary relative degree constraints is the definition of a high-order CBF[34], as described next.

For a constraint $b(\boldsymbol{x}) \geq 0$ with relative degree m, $b : \mathbb{R}^n \to \mathbb{R}$, and $\psi_0(\boldsymbol{x}) := b(\boldsymbol{x})$, we define a sequence of functions $\psi_i : \mathbb{R}^n \to \mathbb{R}, i \in \{1, 2, \cdots, m\}$:

$$\psi_i(\boldsymbol{x}) := \dot{\psi}_{i-1}(\boldsymbol{x}) + \alpha_i(\psi_{i-1}(\boldsymbol{x})), \quad i \in \{1, 2, \cdots, m\}, \tag{6}$$

where $\alpha_i(\cdot), i \in \{1, 2, \cdots, m\}$ denotes a $(m-i)$th order differentiable class $\mathcal{K}$ function.

We further define a sequence of sets $C_i, i \in \{1, 2, \cdots, m\}$ associated with (6) in the form:

$$C_i := \{\boldsymbol{x} \in \mathbb{R}^n : \psi_{i-1}(\boldsymbol{x}) \geq 0\}, \quad i \in \{1, 2, \cdots, m\}. \tag{7}$$

Definition 3.1 (High Order Control Barrier Function (HOCBF)[34]) Let $C_1, C_2, \cdots, C_m$ be defined by (7) and $\psi_1(\boldsymbol{x}), \psi_2(\boldsymbol{x}), \cdots, \psi_m(\boldsymbol{x})$ be defined by (6). A function $b : \mathbb{R}^n \to \mathbb{R}$ is a High Order Control Barrier Function (HOCBF) of relative degree m for the system (1) if there exist $(m-i)$th order differentiable class $\mathcal{K}$ functions $\alpha_i, i \in \{1, 2, \cdots, m-1\}$ and a class $\mathcal{K}$ function α_m such that

$$\sup_{\boldsymbol{u} \in U} [L_f^m b(\boldsymbol{x}) + [L_g L_f^{m-1} b(\boldsymbol{x})]\boldsymbol{u} + S(b(\boldsymbol{x})) + \alpha_m(\psi_{m-1}(\boldsymbol{x}))] \geq 0, \tag{8}$$

for all $\boldsymbol{x} \in C_1 \cap C_2 \cap \cdots \cap C_m$. In (8), L_f^m denotes the Lie derivative along f m times, and

$$S(b(\boldsymbol{x})) = \sum_{i=1}^{m-1} L_f^i (\alpha_{m-i} \circ \psi_{m-i-1})(\boldsymbol{x}),$$

where $\circ$ denotes the composition of functions. Further, $b(\boldsymbol{x})$ is such that $L_g L_f^{m-1} b(\boldsymbol{x}) \neq 0$ on the boundary of the set $C_1 \cap C_2 \cap \cdots \cap C_m$.

The HOCBF is a general form of the relative degree $m = 1$ CBF[19–21] (setting $m = 1$ reduces the HOCBF to the common CBF form in [19–21]), and it is also a more general form of the exponential CBF[29]. Note that we can define $\alpha_i(\cdot), i \in \{1, 2, \cdots, m\}$ in Definition 3.1 to be the extended class $\mathcal{K}$ functions ($\alpha : [-a, a] \to [-\infty, \infty]$ as in Definition 2.1) to ensure robustness of an HOCBF to perturbations[33]; this is due to the fact that the HOCBF constraint becomes a Lyapunov-like condition with extended class $\mathcal{K}$ functions. However, the use of extended class $\mathcal{K}$ functions cannot ensure a constraint to be satisfied if it is initially violated.

Theorem 3.2 (see [34]) *Given an HOCBF $b(\boldsymbol{x})$ from Definition* 3.1 *with the associated sets $C_1, C_2, \cdots, C_m$ defined by* (7)*, if $\boldsymbol{x}(0) \in C_1 \cap C_2 \cap \cdots \cap C_m$, then any Lipschitz continuous controller $\boldsymbol{u}(t)$ that satisfies* (8)*, $\forall t \geq 0$ renders $C_1 \cap C_2 \cap \cdots \cap C_m$ forward invariant for the system* (1).

The satisfaction of the CBF constraint (3) or the HOCBF constraint (8) is only a sufficient condition for the satisfaction of the original safety constraint $b(\boldsymbol{x}) \geq 0$. Therefore, the CBF (HOCBF) method introduces some conservativeness to the system operation. In order to alleviate this drawback, we may properly define the class $\mathcal{K}$ functions of a CBF (HOCBF), as shown in [35]. In order to apply HOCBFs to guarantee the satisfaction of high-relative-degree constraints, we simply replace the CBF constraint in the QP (5) by (8). When tight control bounds are involved, the QP can easily become infeasible at a certain time step since it is solved in a myopic way, as explained at the end of Section 2. In order to address this potential infeasibility problem, one approach is to try to find sufficient conditions for feasibility guarantees, as detailed in the next section.

3.2 Sufficient Conditions for Feasibility

In order to guarantee the feasibility of the optimization problem (5), one obvious way is to derive explicit solutions of (5). It is indeed possible to accomplish this under certain assumptions as shown in [33]. However, for most constrained optimal control problems, it is hard to find such explicit solutions. As an alternative, we can discretize time and problem (5) is replaced by a sequence of QPs, one for each time step. In this case, we need to guarantee that the QP is always feasible at each time step. To address this, we define an additional *feasibility constraint*[36]:

Definition 3.3 (Feasibility constraint) Suppose the QP (5) at each time step is feasible at the current state $\boldsymbol{x}(\bar{t}), \bar{t} \in [0, T)$. A constraint $b_F(\boldsymbol{x}) \geq 0$, where $b_F : \mathbb{R}^n \to \mathbb{R}$, is a feasibility constraint if it makes the QP corresponding to the next time interval feasible.

In order to ensure that the QP (5) is feasible at the next time interval, a feasibility constraint $b_F(\boldsymbol{x}) \geq 0$ should have two important features: (i) It guarantees that the CBF constraint (3) (HOCBF constraint (8)) and that the control constraint imposed through bounds:

$$\boldsymbol{u}_{\min} \leq \boldsymbol{u} \leq \boldsymbol{u}_{\max} \tag{9}$$

do not conflict, and (ii) the feasibility constraint itself does not conflict with both (3) (or (8)) and (9) at the same time.

An illustrative example of how a feasibility constraint works is shown in Figure 1. A robot whose control is determined by solving the QP (5), will run close to an obstacle at the following time step. The next state may be infeasible for the QP associated with that next step. For example, the state denoted by the red dot in Figure 1 may involve too large a speed for the robot to find a feasible control (i.e., a large enough deceleration) to avoid the obstacle in the next step. However, if a feasibility constraint can prevent the robot from reaching this state, then the QP is feasible.

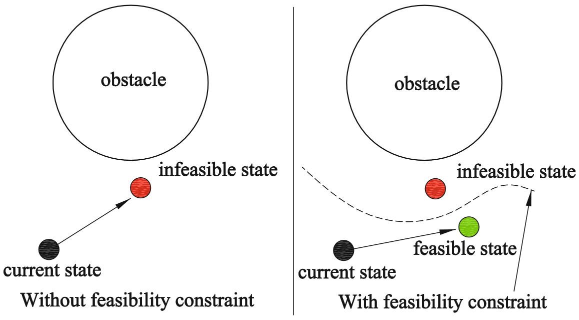


Figure 1 An illustration of how a feasibility constraint works for a robot control problem. A feasibility constraint prevents the robot from going into an infeasible state (red dot). It guides it instead to a feasible state (green dot) below the infeasible state boundary separating feasible from infeasible states

After determining such a feasibility constraint $b_F(\boldsymbol{x}) \geq 0$, we can enforce it through a CBF (similar to the way we enforce any $b(\boldsymbol{x}) \geq 0$ through an associated CBF as in (3)) and include it as an additional constraint for (5) to guarantee feasibility given system state $\boldsymbol{x}$.

If a CBF constraint (3) conflicts with the lower control bound in (9), then by multiplying $\boldsymbol{u}_{\min} \leq \boldsymbol{u}$ with $-L_g b(\boldsymbol{x})$ (suppose $L_g b(\boldsymbol{x}) \geq \boldsymbol{0}$), and comparing with the CBF constraint (3), we can obtain a feasibility constraint:

$$b_F(\boldsymbol{x}) = L_f b(\boldsymbol{x}) + \alpha(b(\boldsymbol{x})) + L_g b(\boldsymbol{x})\boldsymbol{u}_{\min} \geq 0. \tag{10}$$

In order to make sure that the feasibility constraint (10) does not conflict with (3) (or (8)) and (9) at the same time, we use another CBF to enforce (10) and reformulate it so as to have a form similar to (3) (or (8)) such that they are less likely to conflict with each other. Then, we can impose some additional conditions on the reformulated CBF that guarantee (10). These conditions are called *sufficient conditions*, and they are enforced by a CBF and added to the QP (5) to guarantee its feasibility. Additional details (including how other possible cases are handled) are given in [36].

For the adaptive cruise control example considered in [36], a sufficient condition for guaranteeing feasibility is actually an intuitively appealing speed constraint of the ego vehicle. This speed constraint depends on the speed of the preceding vehicle, the lower bound $\boldsymbol{u}_{\min}$ of the ego vehicle, and the definition of the CBF (HOCBF) that guarantees safety. However, safety and feasibility become hard to guarantee for noisy dynamics and/or time-varying control bounds with this approach. This additional complication can be resolved using *adaptive CBFs* as discussed in the next section.

3.3 Adaptive CBFs

Adaptive CBFs (aCBFs) have been proposed in [37] for systems with parameter uncertainties, and a less conservative Robust aCBF (RaCBF)[38] which is combined with a data-driven method has been proposed to adaptively achieve safety. Machine learning techniques have also been applied to adaptively achieve safety for systems with parameter uncertainties[39, 40]. However, the associated QPs can still easily be infeasible when both state constraints (enforced by CBFs or HOCBFs) and tight control bounds are involved. In order to address this, we have recently introduced another formulation of adaptive CBFs[41].

The key idea in converting a regular CBF into an adaptive one is to incorporate penalty terms in a CBF as shown in [34] and then replace them by time-varying functions with suitable properties as detailed next. Starting with a relative degree m function $b : \mathbb{R}^n \to \mathbb{R}$, let $\psi_0(\boldsymbol{x}) := b(\boldsymbol{x})$. Then, instead of using a constant penalty $p_i > 0, i \in \{1, 2, \cdots, m\}$ for each class $\mathcal{K}$ function $\alpha_i(\cdot)$ in the definition of an HOCBF[34], we define a time-varying **penalty function** $p_i(t) \geq 0$, $i \in \{1, 2, \cdots, m\}$, and use it as a multiplicative factor for each class $\mathcal{K}$ function $\alpha_i(\cdot)$. Let $\boldsymbol{p}(t) := (p_1(t), p_2(t), \cdots, p_m(t))$. Similar to (6), we define a sequence of functions

$\psi_i : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}, i \in \{1, 2, \cdots, m\}$ in the form:

$$\begin{aligned} \psi_1(\boldsymbol{x}, \boldsymbol{p}(t)) &:= \dot{\psi}_0(\boldsymbol{x}) + p_1(t)\alpha_1(\psi_0(\boldsymbol{x})), \\ \psi_i(\boldsymbol{x}, \boldsymbol{p}(t)) &:= \dot{\psi}_{i-1}(\boldsymbol{x}, \boldsymbol{p}(t)) + p_i(t)\alpha_i(\psi_{i-1}(\boldsymbol{x}, \boldsymbol{p}(t))), \\ & \qquad\qquad\qquad\qquad i \in \{2, 3, \cdots, m\}, \end{aligned} \tag{11}$$

where $\alpha_i(\cdot), i \in \{1, 2, \cdots, m-1\}$ is a $(m-i)$th order differentiable class $\mathcal{K}$ function, and $\alpha_m(\cdot)$ is a class $\mathcal{K}$ function.

We further define a sequence of sets $C_i, i \in \{1, 2, \cdots, m\}$ associated with (11) in the form:

$$\begin{aligned} C_1 &:= \{\boldsymbol{x} \in \mathbb{R}^n : \psi_0(\boldsymbol{x}) \geq 0\}, \\ C_i &:= \{(\boldsymbol{x}, \boldsymbol{p}(t)) \in \mathbb{R}^n \times \mathbb{R}^m : \psi_{i-1}(\boldsymbol{x}, \boldsymbol{p}(t)) \geq 0\}, \quad i \in \{2, 3, \cdots, m\} \end{aligned} \tag{12}$$

The remaining question is how to choose $p_i(t), i \in \{1, 2, \cdots, m\}$. We require that $p_i(t) \geq 0, \forall i \in \{1, 2, \cdots, m-1\}$, therefore we define each $p_i(t)$ to be an HOCBF, similar to the definition of $b(\boldsymbol{x}) \geq 0$ in Definition 3.1. Just like $b(\boldsymbol{x})$ is associated with the dynamic system (1), we need to introduce an *auxiliary dynamic system* for $p_i(t)$. Moreover, as in Definition 3.1, each penalty function $p_i(t), i \in \{1, 2, \cdots, m-1\}$ will be differentiated $m-i$ times, while $p_m(t)$ is not differentiated. Thus, we start by defining $\boldsymbol{\pi}_i(t) := (\pi_{i,1}(t), \pi_{i,2}(t), \cdots, \pi_{i,m-i}(t)) \in \mathbb{R}^{m-i}, i \in \{1, 2, \cdots, m-2\}$, where $\pi_{i,j} \in \mathbb{R}$, $j \in \{1, 2, \cdots, m-i\}$ are the auxiliary state variables. Next, we define $\boldsymbol{\pi}_{m-1}(t) = p_{m-1}(t) \in \mathbb{R}$ which needs to be differentiated only once. Finally, we set $p_m(t) \geq 0$ as some function to be determined and set $\pi_{i,1}(t) = p_i(t)$ in (11). We define input-output linearizable and controllable auxiliary dynamics for each p_i (we henceforth omit the time variable t for simplicity) through the auxiliary state $\boldsymbol{\pi}_i$ in the form:

$$\begin{aligned} \dot{\boldsymbol{\pi}}_i &= F_i(\boldsymbol{\pi}_i) + G_i(\boldsymbol{\pi}_i)\nu_i, \quad i \in \{1, 2, \cdots, m-1\}, \\ y_i &= p_i, \end{aligned} \tag{13}$$

where y_i denotes the output, $F_i : \mathbb{R}^{m-i} \to \mathbb{R}^{m-i}$, $G_i : \mathbb{R}^{m-i} \to \mathbb{R}^{m-i}$, and $\nu_i \in \mathbb{R}$ denotes the control input for the auxiliary dynamics (13). The exact form of F_i, G_i is mainly used to guarantee the non-negative property of p_i shown later, and it will determine the conservativeness of this *Parameter Adaptive CBF* (PACBF) method. For simplicity, we usually adopt linear forms. For example, we define $\dot{p}_{m-2} = \pi_{m-2,2}$, $\dot{\pi}_{m-2,2} = \nu_{m-2}$ since we need to differentiate p_{m-2} twice as in Definition 3.1, and define $\dot{p}_{m-1} = \nu_{m-1}$ since we need to differentiate p_{m-1} once. We can initialize $\boldsymbol{\pi}_i(0)$ to any vector as long as $p_i(0) > 0$.

An alternative way of viewing (13) is by defining a set of additional state variables which cause the dynamic system (1) to be augmented. In particular, let $\boldsymbol{\Pi} := (\boldsymbol{\pi}_1, \boldsymbol{\pi}_2, \cdots, \boldsymbol{\pi}_{m-1})$, $\boldsymbol{\nu} := (\nu_1, \nu_2, \cdots, \nu_{m-1})$, where $\nu_i, i \in \{1, 2, \cdots, m-1\}$ are the controls in the auxiliary dynamics (13). In order to properly define the PACBF, we augment the system (1) with the auxiliary dynamics (13) in the form:

$$\begin{bmatrix} \dot{\boldsymbol{x}} \\ \dot{\boldsymbol{\Pi}} \end{bmatrix} = \underbrace{\begin{bmatrix} f(\boldsymbol{x}) \\ \boldsymbol{F}_0(\boldsymbol{\Pi}) \end{bmatrix}}_{F(\boldsymbol{x},\boldsymbol{\Pi})} + \underbrace{\begin{bmatrix} g(\boldsymbol{x}) & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{G}_0(\boldsymbol{\Pi}) \end{bmatrix}}_{G(\boldsymbol{x},\boldsymbol{\Pi})} \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{\nu} \end{bmatrix}, \tag{14}$$

where $\boldsymbol{F}_0(\boldsymbol{\Pi}) = (F_1(\boldsymbol{\pi}_1), F_1(\boldsymbol{\pi}_2), \cdots, F_{m-1}(\boldsymbol{\pi}_{m-1}))$ and $\boldsymbol{G}_0(\boldsymbol{\Pi})$ is a matrix composed by $G_i(\boldsymbol{\pi}_i), i \in \{1, 2, \cdots, m-1\}$ with dimension $\frac{m(m-1)}{2} \times (m-1)$. $F : \mathbb{R}^{n+\frac{m(m-1)}{2}} \to \mathbb{R}^{n+\frac{m(m-1)}{2}}, G : \mathbb{R}^{n+\frac{m(m-1)}{2}} \to \mathbb{R}^{(n+\frac{m(m-1)}{2})\times(q+m-1)}$ are the augmented dynamics functions (matrix).

Since p_i is an HOCBF with relative degree $m-i$ for (13), similar to (8), we define a constraint set $U_{cbf}(\boldsymbol{\Pi})$ for $\boldsymbol{\nu}$:

$$\begin{aligned} U_{cbf}(\boldsymbol{\Pi}) = \{\boldsymbol{\nu} \in \mathbb{R}^{m-1} : L_{F_i}^{m-i} p_i + [L_{G_i} L_{F_i}^{m-i-1} p_i]\nu_i + S(p_i) \\ + \alpha_{m-i}(\psi_{i,m-i-1}(p_i)) \geq 0, \quad \forall i \in \{1, 2, \cdots, m-1\}\}, \end{aligned} \tag{15}$$

where $\psi_{i,m-i-1}(\cdot)$ is defined similar to (6).

Definition 3.4 (see [41]) Let $C_i, i \in \{1, 2, \cdots, m\}$ be defined by (12), $\psi_i(\boldsymbol{x}, \boldsymbol{p}), i \in \{1, 2, \cdots, m\}$ be defined by (11), and the auxiliary dynamics be defined by (13). A function $b : \mathbb{R}^n \to \mathbb{R}$ is a *Parameter Adaptive Control Barrier Function* (PACBF) with relative degree m for (1) if every p_i, $i \in \{1, 2, \cdots, m-1\}$ is an HOCBF with relative degree $m-i$ for the auxiliary dynamics (13), and there exist $(m-i)$th order differentiable class $\mathcal{K}$ functions $\alpha_i, i \in \{1, 2, \cdots, m-1\}$, and a class $\mathcal{K}$ function α_m such that

$$\sup_{\boldsymbol{u} \in U, \boldsymbol{\nu} \in U_{cbf}} [L_F^m b(\boldsymbol{x}) + [L_G L_F^{m-1} b(\boldsymbol{x})]\boldsymbol{u} + S(b(\boldsymbol{x}), \boldsymbol{p}, \boldsymbol{\nu}) + p_m \alpha_m(\psi_{m-1}(\boldsymbol{x}, \boldsymbol{p}))] \geq 0, \tag{16}$$

for all $\boldsymbol{x} \in C_1, (\boldsymbol{x}, \boldsymbol{p}) \in C_2 \cap C_3 \cap \cdots \cap C_m$, and all $p_m \geq 0$. In (16), $S(b(\boldsymbol{x}), \boldsymbol{p})$ denotes the remaining Lie derivative terms of $b(\boldsymbol{x})$ (or $\boldsymbol{p}$) along f (or F_i, $i \in \{1, 2, \cdots, m-1\}$) with degree less than m (or $m-i$), similar to the form in (8).

Given a PACBF $b(\boldsymbol{x})$, we consider all control values $(\boldsymbol{u}, \boldsymbol{\nu}) \in U \times U_{cbf}(\boldsymbol{\Pi})$ that satisfy:

$$\begin{aligned} K_{acbf}(\boldsymbol{x}, \boldsymbol{\Pi}) = \{(\boldsymbol{u}, \boldsymbol{\nu}) \in U \times U_{cbf}(\boldsymbol{\Pi}) : L_F^m b(\boldsymbol{x}) + [L_G L_F^{m-1} b(\boldsymbol{x})]\boldsymbol{u} \\ + S(b(\boldsymbol{x}), \boldsymbol{p}, \boldsymbol{\nu}) + p_m \alpha_m(\psi_{m-1}(\boldsymbol{x}, \boldsymbol{p})) \geq 0\}. \end{aligned} \tag{17}$$

Theorem 3.5 (see [41]) *Given a PACBF $b(\boldsymbol{x})$ from Definition* 3.4 *with the associated sets $C_1, C_2, \cdots, C_m$ defined by* (12)*, if $\boldsymbol{x}(0) \in C_1$ and $(\boldsymbol{x}(0), \boldsymbol{p}(0)) \in C_2 \cap C_3 \cap \cdots \cap C_m$, then any Lipschitz continuous controller $(\boldsymbol{u}(t), \boldsymbol{\nu}(t)) \in K_{acbf}(\boldsymbol{x}(t), \boldsymbol{\Pi}(t)), \forall t \geq 0$ renders the set C_1 forward invariant for the system* (1) *and $C_2 \cap C_3 \cap \cdots \cap C_m$ forward invariant for the systems* (1), (13)*, respectively.*

Remark 3.6 (Adaptivity of PACBFs) In the PACBF constraint (16), the control $\boldsymbol{u}$ of the system (1) depends on the controls $\nu_i, \forall i \in \{1, 2, \cdots, m-1\}$ of the auxiliary dynamics (13). The control ν_i is only constrained by the HOCBF constraint in (15) since we require that p_i is a HOCBF, and there are no control bounds on ν_i. Therefore, we partially relax the constraints on the control input of the system (1) in the PACBF constraint (16) by allowing the penalty function $p_i(t), \forall i \in \{1, 2, \cdots, m\}$ to change through $\boldsymbol{\nu}$. However, the forward invariance of the set C_1 is still guaranteed, i.e., the original constraint $b(\boldsymbol{x}) \geq 0$ is guaranteed to be satisfied. This is how a PACBF provides "adaptivity". Note that we may not need to define a penalty function p_i for every class $\mathcal{K}$ function $\alpha_i(\cdot)$ in (11); we can instead define penalty functions for only some of them.

By properly defining the auxiliary dynamics (13), we can show that the satisfaction of the PACBF constraint (16) is a necessary and sufficient condition for the satisfaction of the original safety constraint $b(\boldsymbol{x}) \geq 0$. This implies that we can address the conservativeness of the existing CBF method with the PACBF method. As a result, we can show that the PACBF can guarantee problem feasibility under time-varying control bounds and noisy dynamics[41]. The definition of the auxiliary dynamics (13) depends on the exact problem, and is an ongoing research topic. In addition, we would also like to stabilize all the penalty functions to some desired values using CLFs in order to make sure the system is stable, as described in [41]. Another form of adaptive CBF, which is simpler and is called *Relaxation Adaptive CBF* (RACBF), is also proposed in [41].

Up to this point, we have assumed that the system dynamics are accurately modeled. This is actually an assumption which may be strong to make; in fact, accurate dynamics are often hard to identify, especially for time-varying systems. One approach for addressing this issue is presented in the next section.

3.4 Safety Guarantees for Systems with Unknown Dynamics: An Event-Driven Approach

In order to determine accurate dynamics for systems with uncertainties, the use of machine learning techniques was proposed in [42]. This, however, is computationally expensive and is not guaranteed to yield sufficiently accurate dynamics for the CBF method. Alternatively, the use of piecewise linear systems was proposed in [43], which is also computationally expensive. These approaches fail to work for systems (such as time-varying systems) that require on-line model identification. We describe next a recently introduced approach[44] which can guarantee safety for systems with unknown dynamics. This approach still relies on the CBF-based QP method[19]. The complete solution consists of four steps:

Step 1 Define adaptive affine dynamics. Our motivation is that we need affine dynamics of the form (1) in order to apply the CBF-based QP approach. We define affine dynamics that have the same relative degree for the safety constraint $b(\boldsymbol{x}) \geq 0$ as the real system (assuming this information is available) in order to estimate the real unknown dynamics in the form:

$$\dot{\overline{\boldsymbol{x}}} = f_a(\overline{\boldsymbol{x}}) + g_a(\overline{\boldsymbol{x}})\boldsymbol{u} \tag{18}$$

where $f_a : \mathbb{R}^n \to \mathbb{R}, g_a : \mathbb{R}^n \to \mathbb{R}^{n\times q}$, and $\overline{\boldsymbol{x}} \in X \subset \mathbb{R}^n$ is the state vector corresponding to $\boldsymbol{x}$ in the unknown dynamics. Since $f_a(\cdot), g_a(\cdot)$ in (18) can be adaptively updated to accommodate the real unknown dynamics, as shown in [44], we refer to (18) as the *adaptive affine dynamics*. The real unknown dynamics and (18) are related through the error states obtained from the real-time measurements of the system and the integration of (18). Theoretically, we can take any affine dynamics in (18) to model the real system as long as their states are of the same dimension and with the same physical interpretation within the plant. Clearly, we would like the adaptive dynamics (18) to "stay close" to the real dynamics.

Step 2 Find an HOCBF that guarantees $b(\boldsymbol{x}) \geq 0$. Based on (18), the error state and its derivatives, we use an HOCBF to enforce $b(\boldsymbol{x}) \geq 0$.

Step 3 Formulate the CBF-based QP. We formulate the problem using a CBF-CLF-QP approach[19] as shown in (5), with a CBF replaced by an HOCBF[34] if $m > 1$.

Step 4 Determine the events required to solve the QP and the condition that guarantees the satisfaction of $b(\boldsymbol{x}) \geq 0$ between events. Since there is obviously a difference between the adaptive affine dynamics (18) and the real unknown dynamics, in order to guarantee safety in the real system, we bound the state of the adaptive dynamics, the error state between (18) and the real (observed) system state, and the derivatives of the error state at a certain time step. Then we need to properly define the following events to solve the QP:

- **Event 1** the error state is about to exit the defined bound.
- **Event 2** the derivative of the error state is about to exit the defined bound.
- **Event 3** the state of (18) reaches the boundaries of the defined bound.

In other words, these events are equivalent to determining the times $t_k, k = 1, 2, \cdots (t_1 = 0)$ at which the QP must be solved in order to guarantee the satisfaction of $b(\boldsymbol{x}) \geq 0$ for the real unknown dynamics.

The proposed solution framework is shown in Figure 2 where we note that we apply the same control from the QP to both the real unknown dynamics and (18). Technical details of this framework and simulation examples illustrating its effectiveness are given in [44].

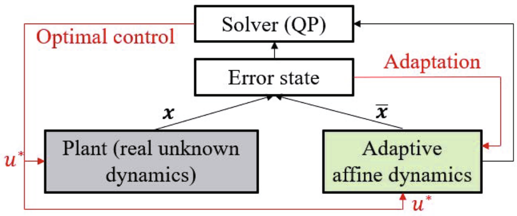


Figure 2 The solution framework for an optimal control problem with safety constraints and the connection between the real unknown dynamics and the adaptive affine dynamics (18). The state $\boldsymbol{x}$ is obtained from the sensor measurements of the plant

Thus far, the solution to the problem (5) is focused on ensuring safety guarantees for an optimal control problem whose objective includes several aspects not included in (5). Therefore, these solutions are sub-optimal relative to the original optimal control problem of interest. To address this issue, we describe next a joint Optimal Control and Barrier Function (OCBF) framework.

4 Bridging the Gap Between Optimal Planning and Safety-Critical Control: The OCBF Approach

In this section, we first present a general-purpose safety-critical optimal control problem, and then overview a joint Optimal Control and Barrier Function (OCBF) framework introduced in [27] to solve it.

Objective (Cost minimization) Consider an optimal control problem for the system (1) with the cost defined as:

$$J = \int_{t_0}^{t_f} [\beta + \mathcal{C}(\boldsymbol{x}, \boldsymbol{u}, t)]\, dt + p||\boldsymbol{x}(t_f) - \overline{\boldsymbol{X}}||^2, \tag{19}$$

where t_0, t_f denote the initial and final times, respectively, and $\mathcal{C} : \mathbb{R}^n \times \mathbb{R}^q \times [t_0, t_f] \to \mathbb{R}^+$ is a cost function. The parameter $\beta \geq 0$ is used to capture a trade-off between the minimization of the time interval $(t_f - t_0)$ and the operational cost $\mathcal{C}(\boldsymbol{x}, \boldsymbol{u}, t)$. $p > 0, \overline{\boldsymbol{X}} \in X$, and the terminal time t_f is generally free (unspecified). It is also possible that all or some of the terminal state variables in $\boldsymbol{x}(t_f)$ are constrained, in which case we include $x_j(t_f) = x_f$ for all such state variables x_j.

Constraint 1 (Safety constraints) Let S_o denote an index set for a set of safety constraints. System (1) should always satisfy

$$b_j(\boldsymbol{x}(t)) \geq 0, \quad \forall t \in [t_0, t_f], \tag{20}$$

where each $b_j : \mathbb{R}^n \to \mathbb{R}, j \in S_o$ is continuously differentiable.

Constraint 2 (Control constraints) These are provided by the control constraint set defind as

$$\boldsymbol{u}_{\min} \leq \boldsymbol{u} \leq \boldsymbol{u}_{\max}, \tag{21}$$

where the inequality is interpreted componentwise, and $\boldsymbol{u}_{\min} \in \mathbb{R}^q, \boldsymbol{u}_{\max} \in \mathbb{R}^q$.

Constraint 3 (State constraints) System (1) should always satisfy the state constraints (componentwise):

$$\boldsymbol{x}_{\min} \leq \boldsymbol{x}(t) \leq \boldsymbol{x}_{\max}, \quad \forall t \in [t_0, t_f], \tag{22}$$

where $\boldsymbol{x}_{\min} \in \mathbb{R}^n$ and $\boldsymbol{x}_{\max} \in \mathbb{R}^n$. Note that we distinguish the state constraints from the safety constraints in (20) since the latter are viewed as hard, while the former usually capture system capability limitations that can be relaxed to improve the problem feasibility; for example, in traffic networks vehicles are constrained by upper and lower speed limits.

Problem 4.1 Find a control policy for the system (1) such that the cost (19) is minimized, constraints (20), (21), and (22) are strictly satisfied.

The cost in (19) can be properly normalized by defining $\beta := \frac{\alpha \sup_{\boldsymbol{x} \in X, \boldsymbol{u} \in U, \tau \in [t_0, t_f]} \mathcal{C}(\boldsymbol{x}, \boldsymbol{u}, \tau)}{(1-\alpha)}$ where $\alpha \in [0, 1)$ and then multiplying (19) by $\frac{\alpha}{\beta}$. Thus, we construct a convex combination as follows:

$$J = \int_{t_0}^{t_f} \left(\alpha + \frac{(1-\alpha)\mathcal{C}(\boldsymbol{x}, \boldsymbol{u}, t)}{\sup_{\boldsymbol{x} \in X, \boldsymbol{u} \in U, \tau \in [t_0, t_f]} \mathcal{C}(\boldsymbol{x}, \boldsymbol{u}, \tau)} \right) dt. \tag{23}$$

If $\alpha = 1$, then we solve (19) as a minimum time problem. The normalized cost (23) facilitates a trade-off analysis between the two metrics. However, we will use the simpler cost expression (19) throughout this paper. Thus, we can take $\beta \geq 0$ as a weight factor that can be adjusted to penalize time relative to the cost $\mathcal{C}(\boldsymbol{x}, \boldsymbol{u}, t)$ in (19).

Approach Step 1 We use Hamiltonian analysis[7] to obtain an optimal control $\boldsymbol{u}^*(t)$ and optimal state $\boldsymbol{x}^*(t)$, $t \in [t_0, t_f]$ for the cost (19) and the system (1), under the terminal state constraint in (19), the safety constraints (20), and the control and state constraints (21), (22). The goal here is to derive a tractable analytical solution to the problem within some given real-time computational constraints. To accomplish this, we may linearize or appropriately simplify the dynamics (1)[45]. We may also omit some or all of the state and control constraints in the problem. The final solution is denoted by $\boldsymbol{u}^*(t)$ and the corresponding state trajectory by $\boldsymbol{x}^*(t)$.

Step 2 There are usually unmodelled dynamics and measurement noise in (1). Thus, we consider a modified version of the system (1) to denote the real dynamics:

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}) + g(\boldsymbol{x})\boldsymbol{u} + \boldsymbol{w}, \tag{24}$$

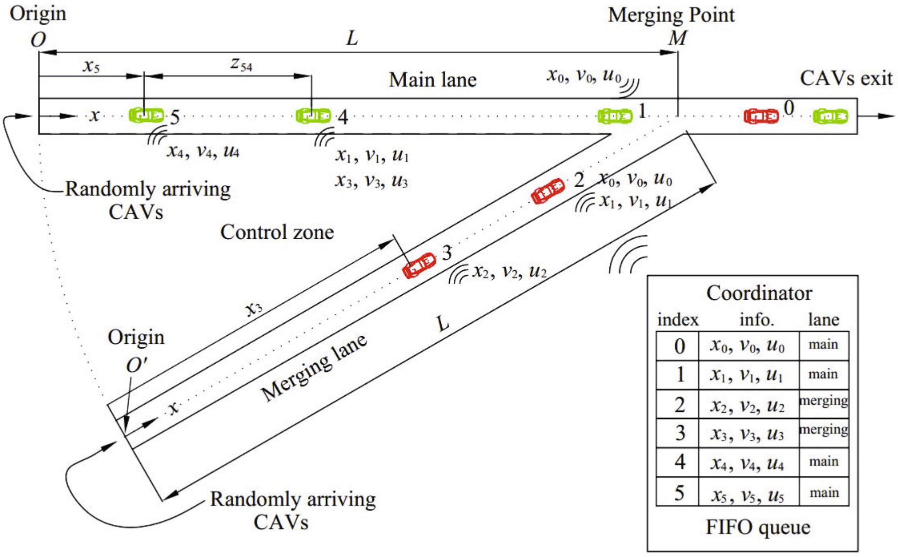
where $\boldsymbol{w} \in \mathbb{R}^n$ denotes all unmodeled uncertainties in the dynamics. We consider $\boldsymbol{x}$ as a measured state which includes the effects of such unmodelled dynamics and measurement noise and which can be used in what follows. Allowing for the noisy dynamics (24), we set $\boldsymbol{u}_{ref}(t) = h(\boldsymbol{u}^*(t), \boldsymbol{x}^*(t), \boldsymbol{x}(t))$ where $h : \mathbb{R}^q \times \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^q$ is a state feedback control. In the simplest possible case, we set $\boldsymbol{u}_{ref}(t) = \boldsymbol{u}^*(t)$. More generally, as in the traffic merging control problem[27] presented in the next section, $u_{ref}(t)$ depends on the optimal position, optimal control and the actual vehicle position. We then use the CBF method to track the optimal control as a reference, i.e.,

$$\min_{\boldsymbol{u}(t)} \int_{t_0}^{t_f} ||\boldsymbol{u}(t) - \boldsymbol{u}_{ref}(t)||^2 dt \tag{25}$$

subject to (i) the CBF constraints (3) corresponding to the safety constraints (20), (ii) the state constraints (22), and (iii) the control constraints (21). In order to better track the optimal state $\boldsymbol{x}^*(t)$ and minimize the deviation $||\boldsymbol{x}(t_f) - \overline{\boldsymbol{X}}||^2$ from the terminal state constraint, we also define a CLF $V(\boldsymbol{x} - \boldsymbol{x}^*)$. Thus, the cost (25) is also subject to the corresponding CLF constraint (4). The resulting problem can then be solved by the approach described at the end of Section 2. Full details are given in [27].

5 An OCBF Application: The Traffic Merging Control Problem

In this section, we present an application of the OCBF framework which arises in a traffic merging problem. This problem occurs when traffic must be joined from two different roads, usually associated with a main lane and a merging lane as shown in Figure 3. We consider the case where all traffic consists of Connected Autonomous Vehicles (CAVs) randomly arriving at the two lanes joined at the Merging Point (MP) M where a collision 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.



index	info.	lane
0	x_0, v_0, u_0	main
1	x_1, v_1, u_1	main
2	x_2, v_2, u_2	merging
3	x_3, v_3, u_3	merging
4	x_4, v_4, u_4	main
5	x_5, v_5, u_5	main

Figure 3 The merging problem: A lateral collision may occur at the MP and rear-end collisions may occur everywhere within the CZ

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 Figure 3) 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

$$\begin{bmatrix} \dot{x}_i(t) \\ \dot{v}_i(t) \end{bmatrix} = \begin{bmatrix} v_i(t) \\ u_i(t) \end{bmatrix}, \tag{26}$$

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

$$J_i(u_i(t)) = \int_{t_i^0}^{t_i^m} \mathcal{C}(u_i(t))dt, \tag{27}$$

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 $i \in S(t)$:

$$z_{i,i_p}(t) \geq \varphi v_i(t) + \delta_0, \quad \forall t \in [t_i^0, t_i^m], \tag{28}$$

where φ denotes the reaction time (as a rule, $\varphi = 1.8$ is used, e.g., [46]). 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_0 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.,

$$z_{1,0}(t_1^m) \geq \varphi v_1(t_1^m) + \delta_0. \tag{29}$$

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

$$\begin{aligned} v_{\min} \leq v_i(t) \leq v_{\max}, \quad \forall t \in [t_i^0, t_i^m], \\ u_{\min} \leq u_i(t) \leq u_{\max}, \quad \forall t \in [t_i^0, t_i^m], \end{aligned} \tag{30}$$

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:

$$\min_{u_i(t)} J_i(u_i(t)) = \int_{t_i^0}^{t_i^m} \left(\alpha + \frac{(1-\alpha)\frac{1}{2}u_i^2(t)}{\frac{1}{2}\max\{u_{\max}^2, u_{\min}^2\}} \right) dt. \tag{31}$$

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

$$\min_{u_i(t)} J_i(u_i(t)) := \beta(t_i^m - t_i^0) + \int_{t_i^0}^{t_i^m} \frac{1}{2} u_i^2(t) dt, \tag{32}$$

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 5.1 For each CAV $i \in S(t)$ governed by dynamics (26), determine a control law such that (32) is minimized subject to (26), (28), (29), (30), given the initial time t_i^0 and the initial and final conditions $x_i(t_i^0) = 0$, $x_i(t_i^m) = L$, $v_i(t_i^0)$.

In addition, we may include the possibility of system model uncertainties, errors due to signal transmission, as well as computation errors. Therefore, we add two noise terms in (26) to get

$$\begin{bmatrix} \dot{x}_i(t) \\ \dot{v}_i(t) \end{bmatrix} = \begin{bmatrix} v_i(t) + w_{i,1}(t) \\ u_i(t) + w_{i,2}(t) \end{bmatrix}, \tag{33}$$

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

Referring to Figure 3, in our simulation examples CAVs arrive according to Poisson processes with arrival rates that we allow to vary. The initial speed $v_i(t_i^0)$ is also randomly generated with uniform distribution in $[15, 20]$ m/s at the origins O and O', respectively. The simulation parameters are: $L = 400$ m, $\varphi = 1.8$ s, $\delta_0 = 0$ m, $u_{\max} = 3.924$ m/s^2, $u_{\min} = -3.924$ m/s^2, $v_{\max} = 30$ m/s, $v_{\min} = 0$ m/s, $\beta = 1$, $c_3 = 10$, $\Delta t = 0.1$ s, 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. The value of Δt is chosen as small as possible, depending on computational resources available, in order to address the inter-sampling effect on the HOCBFs and maintain a guaranteed satisfaction of all constraints.

We show in Figure 4 how the travel time and energy consumption vary as the weight factor α in (31) changes. The significance of Figure 4 is to show how closely the OCBF controller can match the optimal performance (upper bound) obtained through optimal control OC. Examples of the barrier function profiles for the safety constraint (28) under known (the CBF formulation is given in [27, 47]) and unknown noise bound $\boldsymbol{W}$ of $\boldsymbol{w}$ are shown in Figure 5. If $\boldsymbol{W}$ is known, the safety constraint (28) is guaranteed with some conservativeness; otherwise, the safety constraint (28) is satisfied most of the time without conservativeness.

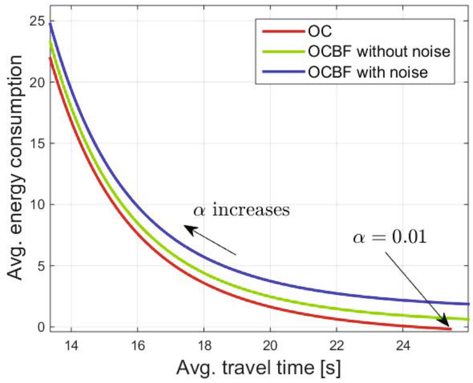


Figure 4 Travel time and energy consumption as the factor α changes

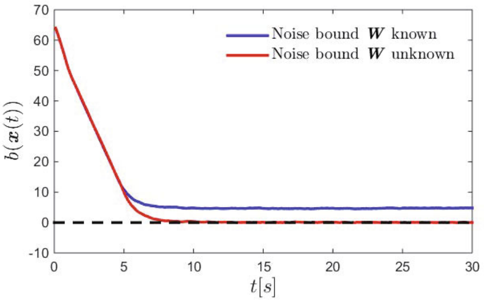


Figure 5 Barrier function $b(\boldsymbol{x})$ under noise $w_{i,1}(t) \in [-4, 4]$ m/s, $w_{i,2}(t) \in [-0.4, 0.4]$ m/s^2. $b(\boldsymbol{x}) \geq 0$ denotes the satisfaction of the safety constraint (28)

6 Conclusion

We have provided an overview of safety-critical optimal control problems where a planning phase determines a trajectory to be followed and a real-time execution phase is primarily responsible for guaranteeing that strict safety constraints are always satisfied. For this second phase, we have described how Control Barrier Functions (CBFs) and Control Lyapunov Functions (CLFs) can be used to generate a sequence of QPs which are computationally efficient to solve in real time. The applicability of CBFs rests on several remaining research challenges which we have discussed how to address, including how to ensure the feasibility of the QPs and how to deal with unknown system dynamics through the use of event-driven methods. We have also overviewed a framework that combines Optimal Control with CBFs, giving rise to OCBF controllers which can lead to near-optimal solutions while guaranteeing safety constraints even in the presence of noisy dynamics.

References

[1] Emuna R, Borowsky A, and Biess A, Deep reinforcement learning for human-like driving policies in collision avoidance tasks of self-driving cars, arXiv preprint: 2006.04218, 2020.

[2] Xiao W, Mehdipour N, Collin A, et al., Rule-based optimal control for autonomous driving, *Proceedings of the ACM/IEEE* 12*th International Conference on Cyber-Physical Systems*, 2021, 143–154.

[3] Breeden J and Panagou D, Quadratic programs for high relative degree spatial constraints and spatiotemporal specifications with spacecraft applications, 59*th IEEE Conference on Decision and Control*, 2020, 1496–1502.

[4] Pepyne D and Cassandras C, Optimal control of hybrid systems in manufacturing, *Proceedings of the IEEE*, 2000, **88**(7): 1108–1123.

[5] LaValle S M, Kuffner J, and James J, Randomized kinodynamic planning, *International Journal of Robotics Research*, 2001, **20**(5): 378–400.

[6] Hart P E, Nilsson N J, and Raphael B, A formal basis for the heuristic determination of minimum cost paths, *IEEE Transactions on Systems Science and Cybernetics*, 1968, **4**(2): 100–107.

[7] Bryson and Ho, *Applied Optimal Control*, Ginn Blaisdell, Waltham, MA, 1969.

[8] Garcia C E and Prett D M, Model predictive control: Theory and practice, *Automatica*, 1989, **25**(3): 335–348.

[9] Mayne D, Model predictive control: Recent developments and future promise, *Automatica*, 2014, **50**(12): 2967–2986.

[10] Bemporad A, Borrelli F, and Morari M, Model predictive control based on linear programming, the explicit solution, *IEEE Transactions on Automatic Control*, 2002, **47**(12): 1974–1985.

[11] Bertsimas D and Tsitsiklis J N, *Introduction to Linear Optimization*, Athena Scientific, Belmont, MA, 1997.

[12] Tee K P, Ge S S, and Tay E H, Barrier Lyapunov functions for the control of output-constrained nonlinear systems, *Automatica*, 2009, **45**(4): 918–927.

[13] Wieland P and Allgower F, Constructive safety using control barrier functions, *Proc. of* 7*th IFAC Symposium on Nonlinear Control System*, 2007.

[14] Boyd S P and Vandenberghe L, *Convex Optimization*, Cambridge University Press, New York, 2004.

[15] Aubin J P, *Viability Theory*, Springer, New York, 2009.

[16] Prajna S, Jadbabaie A, and Pappas G J, A framework for worst-case and stochastic safety verification using barrier certificates, *IEEE Transactions on Automatic Control*, 2007, **52**(8): 1415–1428.

[17] Wisniewski R and Sloth C, Converse barrier certificate theorem, *Proc. of* 52*nd IEEE Conference on Decision and Control*, Florence, Italy, 2013, 4713–4718.

[18] Panagou D, Stipanovic D M, and Voulgaris P G, Multi-objective control for multi-agent systems using lyapunov-like barrier functions, *Proc. of* 52*nd IEEE Conference on Decision and Control*, Florence, Italy, 2013, 1478–1483.

[19] Ames A D, Grizzle J W, and Tabuada P, Control barrier function based quadratic programs with application to adaptive cruise control, *Proc. of* 53*rd IEEE Conference on Decision and Control*, 2014, 6271–6278.

[20] Glotfelter P, Cortes J, and Egerstedt M, Nonsmooth barrier functions with applications to multi-robot systems, *IEEE Control Systems Letters*, 2017, **1**(2): 310–315.

[21] Lindemann L and Dimarogonas D V, Control barrier functions for signal temporal logic tasks, *IEEE Control Systems Letters*, 2019, **3**(1): 96–101.

[22] Sontag E, A Lyapunov-like characterization of asymptotic controllability, *SIAM Journal on Control and Optimization*, 1983, **21**(3): 462–471.

[23] Artstein Z, Stabilization with relaxed controls, *Nonlinear Analysis: Theory, Methods & Applications*, 1983, **7**(11): 1163–1173.

[24] Freeman R A and Kokotovic P V, *Robust Nonlinear Control Design*, Birkhauser, Berlin, 1996.

[25] Ames A D, Galloway K, and Grizzle J W, Control lyapunov functions and hybrid zero dynamics, *Proc. of* 51*st IEEE Conference on Decision and Control*, 2012, 6837–6842.

[26] Galloway K, Sreenath K, Ames A D, et al., Torque saturation in bipedal robotic walking through control Lyapunov function based quadratic programs, *IEEE Access*, 2015, **3**: 323–332.

[27] Xiao W, Cassandras C G, and Belta C, Bridging the gap between optimal trajectory planning and safety-critical control with applications to autonomous vehicles, *Automatica*, 2021, 129.109592.

[28] Khalil H K, *Nonlinear Systems*, Third Edition, Prentice Hall, Upper Saddle River, New Jersey, 2002.

[29] Nguyen Q and Sreenath K, Exponential control barrier functions for enforcing high relative-degree safety-critical constraints, *Proc. of the American Control Conference*, 2016, 322–328.

[30] Hsu S C, Xu X, and Ames A D, Control barrier function based quadratic programs with application to bipedal robotic walking, *Proc. of the American Control Conference*, 2015, 4542–4548.

[31] Wu G and Sreenath K, Safety-critical and constrained geometric control synthesis using control Lyapunov and control barrier functions for systems evolving on manifolds, *Proc. of the American Control Conference*, 2015, 2038–2044.

[32] Xu X, Constrained control of inputoutput linearizable systems using control sharing barrier functions, *Automatica*, 2018, **87**: 195–201.

[33] Ames A D, Xu X, Grizzle J W, et al., Control barrier function based quadratic programs for safety critical systems, *IEEE Transactions on Automatic Control*, 2017, **62**(8): 3861–3876.

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

[35] Xiao W, Belta C, and Cassandras C G, Feasibility guided learning for constrained optimal control problems, *Proc. of* 59*th IEEE Conference on Decision and Control*, 2020, 1896–1901.

[36] Xiao W, Balta C, and Cassandras C G, Sufficient conditions for feasibility of optimal control problems using control barrier functions, *Automatica* (in print), arXiv: 2011.08248, 2021.

[37] Taylor A J and Ames A D, Adaptive safety with control barrier functions, *Proc. of the American Control Conference*, 2020, 1399–1405.

[38] Lopez B T, Slotine J J E, and How J P, Robust adaptive control barrier functions: An adaptive and data-driven approach to safety, *IEEE Control Systems Letters*, 2020, **5**(3): 1031–1036.

[39] Fan D D, Nguyen J, Thakker R, et al., Bayesian learning-based adaptive control for safety critical systems, 2020 *IEEE International Conference on Robotics and Automation*, 2020, 4093–4099.

[40] Khojasteh M J, Dhiman V, Franceschetti M, et al., Probabilistic safety constraints for learned high relative degree system dynamics, *Learning for Dynamics and Control*, 2020, 781–792.

[41] Xiao W, Belta C, and Cassandras C G, Adaptive control barrier functions, *IEEE Transactions on Automatic Control*, 2021, DOI: 10.1109/TAC.2021.3074895.

[42] Taylor A J, Singletary A, Yue Y, et al., Learning for safety-critical control with control barrier functions, *Proc. of Conf. on Learning for Dynamics and Control*, 2020, 708–717.

[43] Sadraddini S and Belta C, Formal guarantees in data-driven model identification and control synthesis, *Proc. of the* 21*st Conference on Hybrid Systems: Computation and Control*, 2018, 147–156.

[44] Xiao W, Belta C, and Cassandras C G, Event-triggered safety-critical control for systems with unknown dynamics, *Proc. of* 60*th IEEE Conference on Decision and Control*, preprint arXiv: 2103.15874, 2021.

[45] Xiao W and Cassandras C G, Decentralized optimal merging control for connected and automated vehicles with safety constraint guarantees, *Automatica*, 2021, **123**(5): 109333.

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

[47] Lindemann L and Dimarogonas D V, Control barrier functions for multi-agent systems under conflicting local signal temporal logic tasks, *IEEE Control Systems Letters*, 2019, **3**(3): 757–762.