Event-Triggered Control for Safety-Critical Systems with Unknown Dynamics

Wei Xiao, *Member, IEEE* Calin Belta, *Fellow, IEEE* and Christos G. Cassandras *Fellow, IEEE*

***Abstract*—This paper addresses the problem of safety-critical control for multi-agent systems with unknown dynamics in unknown environments. It has been shown that stabilizing affine control systems to desired (sets of) states while optimizing quadratic costs 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). One of the main challenges in this approach is obtaining accurate system dynamics of all components in the system, which is especially difficult when online model identification is required given limited computational resources and system data. We address this problem by proposing a robust framework (to unknown dynamics including uncertainties) through defining adaptive affine control dynamics that are updated based on the error states obtained by real-time sensor measurements. We define a CBF for a safety requirement on the unmodelled agents based on the adaptive dynamics and error states, and reformulate the safety-critical control problem as the above mentioned sequence of QPs. Then we determine a set of events that trigger the QPs and ensure safety when solving them. We also derive a condition that guarantees the satisfaction of a CBF constraint between events. The proposed framework can also be used for state convergence guarantees for systems with unknown dynamics based on CLFs. We illustrate the effectiveness of the proposed framework on a robot control problem, an adaptive cruise control problem and a traffic merging problem using autonomous vehicles. We also compare the proposed event-driven method with the classical time-driven approach.**

***Index Terms*—Unknown Dynamics, Event-Driven Control, Control Barrier Function, Optimal Control.**

I. INTRODUCTION

CONSTRAINED optimal control problems with safety specifications are central to increasingly widespread safety-critical autonomous and cyber physical systems. Traditional Hamiltonian analysis [1] and dynamic programming [2] cannot accommodate the size and nonlinearities of such systems, and they only work efficiently for small-scale linear systems. Model predictive control (MPC) [3] methods have been shown to work for large, non-linear systems that can be easily linearized. However, safety requirements are hard to guarantee. Motivated by these limitations, barrier and control barrier functions enforcing hard safety constraints have received increased attention in recent years [4] [5] [6].

W. Xiao is with the Computer Science and Artificial Intelligence Lab, Massachusetts Institute of Technology weixy@mit.edu

C. Belta and C. G. Cassandras are with the Division of Systems Engineering and Center for Information and Systems Engineering, Boston University, Brookline, MA, 02446, USA {cbelta,cgc}@bu.edu

This work was supported in part by NSF under grants IIS-1723995, CPS-1446151, ECCS-1931600, DMS-1664644, CNS-1645681, by AFOSR under grant FA9550-19-1-0158, by ARPAE under grant DE-AR0001282, and by the MathWorks.

Barrier functions (BFs) are Lyapunov-like functions [7], [8], whose use can be traced back to optimization problems [9]. More recently, they have been employed to prove set invariance [10], [11], [12] and for multi-objective control [13]. In [7], it was proved that if a BF for a given set satisfies Lyapunov-like conditions, then the set is forward invariant. Control BFs (CBFs) are extensions of BFs for control systems, and are used to map a constraint defined over system states to a constraint on the control input. The CBFs from [4] and [5] work for constraints that have relative degree one with respect to the system dynamics. The exponential CBF [14], which works for arbitrarily high relative degree constraints, employs input-output linearization and finds a pole placement controller with negative poles. The high order CBF (HOCBF) proposed in [6] is simpler and more general than the exponential CBF [14].

CBFs are usually based on the assumption that the control system is affine in controls and the cost is quadratic in controls. Convergence to desired states is achieved by Control Lyapunov Functions (CLFs) [15]. The time domain is discretized, and the state and control are assumed to be constant over each time interval. The optimal control problem is thus reduced to a (possibly large) sequence of Quadratic Program (QPs), one for each time interval [16], where the control is kept constant for the whole interval.

Most existing works based on the QP approach use a uniform time discretization. One of the challenges is to adapt this process (i.e., determine the next time when a QP needs to be solved) to guarantee safety. The work in [17] proposes an approach based on the Lipschitz constants of the system. The authors of [18] use a procedure inspired from event-triggered control for Lyapunov functions [19]. The prescribed performance is used to trigger the update of control using CBFs [20]. All these approaches assume that the dynamics are accurately modelled, which is often not the case in reality. To infer dynamics, machine learning techniques can be used [21] [22], which are computationally expensive and not guaranteed to yield sufficiently accurate dynamics for the CBF method. Although it is possible to still guarantee safety though uncertainty bounds [21], or for probabilistic satisfaction [22], the conservativeness issue is difficult to address. The work in [23] uses piecewise linear systems to estimate the system dynamics, which is also computationally expensive. All these approaches fail to work for systems (such as time-varying systems) that require online model identification. Instead, we focus on how to address the safety-critical problem with unknown dynamics in an online and less-conservative fashion. CBFs for multi-agent systems have also been extensively studied, as in [24], [25]. However, the dynamics of other agents are even harder to identify, which we also address here.

In this paper, we propose a robust framework by defining adaptive affine dynamics that are updated in a time-efficient way to approximate the actual unmodelled dynamics. The adaptive and real dynamics are related through the error states obtained by real-time sensor measurements. We define a HOCBF for a safety requirement on the actual system based on the adaptive dynamics and error states, and reformulate the safety-critical control problem as the above mentioned sequence of QPs. We determine a set of events required to trigger each QP solution in order to ensure safety and derive a condition that guarantees the satisfaction of the HOCBF constraint between events. The triggering of events is based on the value of the HOCBF. The adaptive dynamics are updated at each event to accommodate the real dynamics according to the error states. We illustrate our approach and compare with the classical time-driven method on a robot control problem, an ACC problem and a traffic merging problem using autonomous vehicles.

The contributions of this paper can be summarized as follows. First, our computationally efficient event-triggered framework is robust, and guarantees safety for systems with unknown dynamics, especially for systems requiring online identification (for dynamics), in which case computationally expensive modelling approaches [21] [23] fail to work. Second, we make the events in the proposed framework adaptive such that their triggers depend on the values of the safety metrics (i.e., how far the system state is from the unsafe set). This makes our method significantly less conservative (quantified by how much the system state can approach the unsafe set boundary) and computationally more efficient (fewer QPs) than other methods. Third, we address the singularity problem prevalent in all CBF/CLF methods (i.e., the control coefficients corresponding to a CBF/CLF constraint can become 0 at a certain state) by making it an additional event in the proposed framework. Fourth, the proposed framework also addresses the inter-sampling effect of the CBF method (i.e., safety guarantees between event times), and can be used to guarantee state convergence for CLFs as well.

This paper builds on but significantly extends [26] which proposed a conservative event-triggered framework guaranteeing safety for a single system. Here, we extend the framework to multi-agent systems, alleviate the conservativeness of the framework using adaptive events, consider state convergence guarantees, and address the singularity problem of the CBF/CLF method in the proposed framework with more case studies.

The remaining of the paper is organized as follows. In Sec. II, we present preliminaries on HOCBFs and CLFs. Sec. III formulates a general optimal control problem. We derive the event triggered control for safety-critical systems in Sec. IV, and show how the proposed framework can be applied to state convergence guarantees in Sec. V. We present case studies and results in Sec. VI, and conclude this paper in Sec. VII.

II. PRELIMINARIES

Definition 1. *(Class $\mathcal{K}$ function [27]) A continuous function $\alpha : [0, a) \to [0, \infty), a > 0$ is said to belong to class $\mathcal{K}$ if it is strictly increasing and $\alpha(0) = 0$.*

Consider an affine control system (assumed to be known in this section) of the form

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

where $\boldsymbol{x} \in X \subset \mathbb{R}^n$, where X is a closed state constraint set, $f : \mathbb{R}^n \to \mathbb{R}^n$ and $g : \mathbb{R}^n \to \mathbb{R}^{n\times q}$ are Lipschitz continuous, and $\boldsymbol{u} \in U \subset \mathbb{R}^q$ is a closed control constraint set defined as

$$U := \{\boldsymbol{u} \in \mathbb{R}^q : \boldsymbol{u}_{min} \leq \boldsymbol{u} \leq \boldsymbol{u}_{max}\}. \tag{2}$$

with $\boldsymbol{u}_{min}, \boldsymbol{u}_{max} \in \mathbb{R}^q$ and the inequalities are interpreted componentwise.

Definition 2. *A set $C \subset \mathbb{R}^n$ is forward invariant for system (1) if its solutions starting at any $\boldsymbol{x}(0) \in C$ satisfy $\boldsymbol{x}(t) \in C$, $\forall t \geq 0$.*

Definition 3. *(Relative degree) The relative degree of a (sufficiently many times) differentiable function $b : \mathbb{R}^n \to \mathbb{R}$ with respect to system (1) is the number of times it needs to be differentiated along its dynamics until the control $\mathbf{u}$ explicitly shows in the corresponding derivative.*

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.

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 $\psi_i : \mathbb{R}^n \to \mathbb{R}, i \in \{1, \dots, m\}$:

$$\psi_i(\boldsymbol{x}) := \dot{\psi}_{i-1}(\boldsymbol{x}) + \alpha_i(\psi_{i-1}(\boldsymbol{x})), i \in \{1, \dots, m\}, \tag{3}$$

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

We further define a sequence of sets $C_i, i \in \{1, \dots, m\}$ associated with (3) in the form:

$$C_i := \{\boldsymbol{x} \in X : \psi_{i-1}(\boldsymbol{x}) \geq 0\}, i \in \{1, \dots, m\}. \tag{4}$$

Definition 4. *(High Order Control Barrier Function (HOCBF) [6]) Let $C_1, \dots, C_m$ be defined by (4) and $\psi_1(\boldsymbol{x}), \dots, \psi_m(\boldsymbol{x})$ be defined by (3). A function $b : \mathbb{R}^n \to \mathbb{R}$ is a High Order Control Barrier Function (HOCBF) of relative degree m for system (1) if there exist $(m-i)^{th}$ order differentiable class $\mathcal{K}$ functions $\alpha_i, i \in \{1, \dots, m-1\}$ and a class $\mathcal{K}$ function α_m such that*

$$\sup_{\boldsymbol{u}\in U}[L_f^m b(\boldsymbol{x}) + L_g L_f^{m-1} b(\boldsymbol{x})\boldsymbol{u} + R(b(\boldsymbol{x})) + \alpha_m(\psi_{m-1}(\boldsymbol{x}))] \geq 0, \tag{5}$$

for all $\boldsymbol{x} \in C_1 \cap, \dots, \cap C_m$. In (5), L_f^m (L_g) denotes Lie derivatives along f (g) m (one) times, and $R(\cdot)$ denotes the remaining Lie derivatives along f with degree less than or equal to $m-1$ (omitted for simplicity, see [6]). Moreover, it is assumed that $L_g L_f^{m-1} b(\boldsymbol{x}) \neq 0$ when $b(\boldsymbol{x}) = 0$.

The HOCBF is a general form of the relative degree one CBF [4], [5], [28], i.e., setting $m = 1$ reduces the HOCBF to the common CBF form (it is assumed that $L_g b(\boldsymbol{x}) \neq 0$ when $b(\boldsymbol{x}) = 0$):

$$L_f b(\boldsymbol{x}) + L_g b(\boldsymbol{x})\boldsymbol{u} + \alpha_1(b(\boldsymbol{x})) \geq 0, \tag{6}$$

and it is also a general form of the exponential CBF [14].

Theorem 1. *([6]) Given a HOCBF $b(\boldsymbol{x})$ from Def. 4 with the associated sets $C_1, \dots, C_m$ defined by (4), if $\boldsymbol{x}(0) \in C_1\cap, \dots, \cap C_m$, then any Lipschitz continuous controller $\boldsymbol{u}(t)$ that satisfies (5), $\forall t \geq 0$ renders $C_1\cap, \dots, \cap C_m$ forward invariant for system (1).*

Definition 5. *(Control Lyapunov Function (CLF) [15]) A continuously differentiable function $V : \mathbb{R}^n \to \mathbb{R}$ is an exponentially stabilizing control Lyapunov function (CLF) for system (1) if there exist constants $c_1 > 0, c_2 > 0, c_3 > 0$ such that for $\forall \boldsymbol{x} \in \mathbb{R}^n$, $c_1||\boldsymbol{x}||^2 \leq V(\boldsymbol{x}) \leq c_2||\boldsymbol{x}||^2$,*

$$\inf_{\boldsymbol{u}\in U}[L_f V(\boldsymbol{x}) + L_g V(\boldsymbol{x})\boldsymbol{u} + c_3 V(\boldsymbol{x})] \leq 0. \quad (7)$$

Many existing works [4], [14], [17] combine CBFs for systems with relative degree one with quadratic costs to form optimization problems. An explicit solution to such problems can be obtained based on some assumptions [29]. Alternatively, we can discretize time and an optimization problem with constraints given by the CBFs (inequalities of the form (5)) is solved at each time step. The inter-sampling effect in this approach is considered in [17]. If convergence to a state is desired, then a CLF constraint of the form (7) is added, as in [4] [17]. Note that these constraints are linear in the control since the state value is fixed at the beginning of the interval. Therefore, each optimization problem is a quadratic program (QP). The optimal control obtained by solving each QP is applied at the current time step and held constant for the whole interval. The state is updated using dynamics (1), and the procedure is repeated. Replacing CBFs by HOCBFs allows us to handle constraints with arbitrary relative degree [6]. Throughout the paper, we will refer to this method as the *time-driven* approach. The CBF method works if (1) is an accurate model for the system. However, this is often not the case in reality, especially for time-varying systems. In what follows, we show how we can find a safety-guaranteed controller for systems with unknown dynamics.

III. PROBLEM FORMULATION AND APPROACH

We consider a multi-agent system with a controlled agent (state $\boldsymbol{x} \in X$ and control $\boldsymbol{u} \in U$) whose dynamics are unknown, and with a set S_a of other agents (state $\boldsymbol{y}_i \in X$ for agent $i \in S_a$) whose dynamics are also unknown. For instance, the controlled agent could be the ego vehicle in autonomous driving, and other agents are either other vehicles or obstacles. The controlled agent only has onboard sensors to detect its own state and that of other agents. For the unknown dynamics of the controlled agent and other agents, we make the following assumption:

Assumption 1. *The relative degree of each component of $\boldsymbol{x}$ is known with respect to the real unknown dynamics[1], and the same applies to $\boldsymbol{y}_i, i \in S_a$.*

A typical example for the above assumption is the autonomous driving problem, in which case we have a controlled (ego) vehicle and some other vehicles around the controlled one. If the position of the controlled vehicle (whose dynamics are unknown) is a component in $\boldsymbol{x}$ and the control is acceleration, then the relative degree of the position with respect to the unknown vehicle dynamics is two by Newton's law. The same applies to other vehicles that the controlled vehicle may interact with. We assume that we have sensors to monitor $\boldsymbol{x}$ and its derivatives, as well as to monitor $\boldsymbol{y}_i$ and its derivatives for all $i \in S_a$. Measuring derivatives of $\boldsymbol{x}$ can be challenging, but accurate measurements may not be necessary: we can relax this requirement by limiting measurement accuracy within some bounds, as shown later.

Note that the controlled agent can be swapped with any of the other agents, i.e., we may choose to consider some other agent as the controlled one and select the remaining agents to form the set S_a.

Objective 1: (Minimizing cost) Consider an optimal control problem for the real unknown dynamics of the controlled agent with the cost:

$$\min_{\boldsymbol{u}(t)} \int_0^T \mathcal{C}(||\boldsymbol{u}(t)||)dt \quad (8)$$

where $||\cdot||$ denotes the 2-norm of a vector, $\mathcal{C}(\cdot)$ is a strictly increasing function of its argument. $T > 0$.

Objective 2: (Terminal state constraint) We wish the terminal state $\boldsymbol{x}(T)$ to reach a point $\boldsymbol{K}$, where $\boldsymbol{K} \in \mathbb{R}^n$. This cost can be easily extended to multiple terminal costs for different state components.

Safety requirements: The real unknown dynamics of the controlled agent should always satisfy a safety requirement with respect to another agent $i \in S_a$ whose dynamics are also unknown:

$$b(\boldsymbol{x}(t), \boldsymbol{y}_i(t)) \geq 0, \forall t \in [0, T]. \quad (9)$$

where $b : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable and has relative degree $m \in \mathbb{N}$ with respect to the real controlled agent. The above safety constraint is defined pair-wise, but can be extended to involve more than two agents. The relative degree m is known by Assumption 1.

Control constraints: The controlled agent should always satisfy control bounds in the form of (2).

A control policy for the real unknown dynamics of the controlled agent is $\boldsymbol{feasible}$ if constraints (9) and (2) are satisfied at all times. Note that state limitations are particular forms of (9) that only depend on the state $\boldsymbol{x}$. In this paper, we consider the following problem:

Problem 1. *Find a feasible control policy for the real unknown dynamics of the controlled agent such that the cost (8) is minimized.*

Approach: Problem 1 is a general problem formulation. In order to solve it in an online fashion, we replace the safety constraints above by HOCBF constraints, and solve it through a sequence of discretization steps (conservatively). Thus, Problem 1 is solved by point-wise optimization, which leads to suboptimal solutions compared to the original problem. There are four steps involved in the solution:

Step 1: define adaptive affine dynamics for the controlled agent and adaptive dynamics for other agents. We need

[1]The relative degree is defined similarly to Def. 3 for the real unknown dynamics.

affine dynamics of the form (1) in order to apply the CBF-based QP approach to solve Problem 1. Under Assump. 1, we define affine dynamics that have the same relative degree for (9) as the real controlled agent and we estimate through $\bar{\boldsymbol{x}}$ the actual state $\boldsymbol{x}$ using the dynamics:

$$\dot{\bar{\boldsymbol{x}}} = f_a(\bar{\boldsymbol{x}}) + g_a(\bar{\boldsymbol{x}})\boldsymbol{u} \tag{10}$$

where $f_a : \mathbb{R}^n \to \mathbb{R}, g_a : \mathbb{R}^n \to \mathbb{R}^{n\times q}$, and $\bar{\boldsymbol{x}} \in X \subset \mathbb{R}^n$ is the state vector corresponding to $\boldsymbol{x}$ in the unknown dynamics. Since $f_a(\cdot), g_a(\cdot)$ in (10) can be adaptively updated to accommodate the real unknown dynamics, as shown in the next section, we call (10) *adaptive affine dynamics*. The real unknown dynamics and (10) are related through the error states $\boldsymbol{e} := \boldsymbol{x} - \bar{\boldsymbol{x}}$ obtained from the real-time measurements of the agent and the integration of (10) as f_a, g_a are known. There are user-defined bounds for these errors, and the convergence between the adaptive affine dynamics (10) and the real dynamics depends on the update events. Theoretically, we can take any affine dynamics in (10) to model the agent as long as their states are of the same dimension and with the same physical interpretation as those of the plant. Clearly, we would like the adaptive dynamics (10) to "stay close" to the real dynamics. This notion will be formalized in the next section.

Along the same lines, if other agents have their own (unknown) dynamic model, we also define adaptive dynamics for each agent $i \in S_a$ to estimate its real unknown dynamics in the form:

$$\dot{\bar{\boldsymbol{y}}}_i = h_{a,i}(\bar{\boldsymbol{y}}_i) \tag{11}$$

where $\bar{\boldsymbol{y}}_i \in X$ with $h_{a,i} : \mathbb{R}^n \to \mathbb{R}$, and $\bar{\boldsymbol{y}}_i$ is the state vector corresponding to $\boldsymbol{y}_i$ in the unknown dynamics. Note that $h_{a,i}(\cdot)$ will also be adaptively updated and we refer to (11) as *adaptive dynamics*. The real unknown dynamics of agent $i \in S_o$ and (11) are also related through the error states obtained from the real-time measurements of the agent and the integration of (11) (as $h_{a,i}$ is known). Observe that it is possible that $h_{a,i}(\cdot)$ also includes the control of agent i (which is omitted for simplicity), in which case, we have a multi-agent control problem. Our approach in this paper can also work for such multi-agent control problems (in which case (11) is also affine in control). We focus only on decentralized multi-agent control problems in this paper for simplicity, but the proposed framework can be applied to cases in which controls should be jointly determined (as in game theory). Therefore, we omit the control component in $h_{a,i}(\cdot)$.

Step 2: find a HOCBF that guarantees (9). Based on (10), (11), the error state and its derivatives, we use a HOCBF to enforce (9). Details are shown in the next section.

Step 3: formulate the CBF-based QP. We use a CLF to have the terminal state approach $\boldsymbol{K}$ in Objective 2. The approach to guarantee convergence for unknown dynamics is similar to the above mentioned HOCBF method, and it will be shown in the following sections. If $\mathcal{C}(||\boldsymbol{u}(t)||) = ||\boldsymbol{u}(t)||^2$ in (8), then we can formulate Problem 1 using a CBF-CLF-QP approach [4], with a CBF replaced by a HOCBF [6] if $m > 1$.

Step 4: determine the events required to specify when to solve the QP and the conditions that guarantee the satisfaction of (9) between events. We need to determine the event times $t_k, k = 1, 2, \ldots (t_1 = 0)$ at which each QP must be solved in order to guarantee the satisfaction of (9) for the real unknown dynamics. Since there is obviously a difference between the adaptive affine dynamics (10) and the real unknown dynamics of the controlled agent (as well as between the adaptive dynamics (11) and other agents), in order to guarantee safety for the controlled agent, we need to properly define events (dependent on the error states, the state of (10) and the state of (11)) to solve the QP.

The proposed solution framework is outlined in Fig. 1 where we note that we apply the same control from the QP to both the real unknown dynamics of the controlled agent and to the adaptive affine dynamics in (10). Note that in a static environment, i.e., other agent states do not change or are known, we can remove the adaptive dynamics and other plant blocks in Fig. 1.

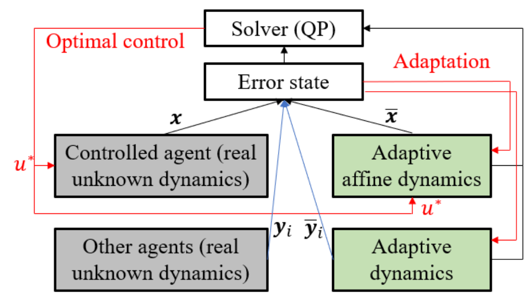


Fig. 1. The solution framework for Problem 1, the connection between the real unknown dynamics of the controlled agent and the adaptive affine dynamics (10), and the connection between the real unknown dynamics of other agents in S_a and the adaptive dynamics (11). The states $\boldsymbol{x}, \boldsymbol{y}_i, i \in S_a$ are from the sensor measurements of the controlled agent and the other agents.

IV. Safety-Critical Control

In this section, we treat the terminal cost in Objective 2 as a soft constraint (the case of a hard terminal constraint will be covered in the next section), and provide the technical details involved in formulating the CBF-based QPs that guarantee the satisfaction of the safety constraint (9) for the controlled agent. We start with a relative-degree-one safety constraint (9).

A. Relative-degree-one Constraints

Suppose the safety constraint in (9) has relative degree one with respect to both dynamics (10) and the actual dynamics of the controlled agent. In this case, since (9) involves the state of both the controlled agent and agent $i \in S_a$, the set C_1 corresponding to (4) takes the form:

$$C_1 = \{(\boldsymbol{x}, \boldsymbol{y}_i) \in X \times X : b(\boldsymbol{x}, \boldsymbol{y}_i) \geq 0\}. \tag{12}$$

Next, we show how to find a CBF that guarantees (9) for the real unknown dynamics. Let

$$\boldsymbol{e}_x := \boldsymbol{x} - \bar{\boldsymbol{x}}, \qquad \boldsymbol{e}_i = \boldsymbol{y}_i - \bar{\boldsymbol{y}}_i, i \in S_a. \tag{13}$$

Note that $\boldsymbol{x}$ and $\bar{\boldsymbol{x}}$ are state vectors from direct measurements of the controlled agent and from the adaptive affine dynamics (10), respectively, and $\boldsymbol{y}_i$ and $\bar{\boldsymbol{y}}_i$ are state vectors from direct measurements of agent $i \in S_a$ and from the adaptive dynamics (11), respectively. Then,

$$b(\boldsymbol{x}, \boldsymbol{y}_i) = b(\bar{\boldsymbol{x}} + \boldsymbol{e}_x, \bar{\boldsymbol{y}}_i + \boldsymbol{e}_i). \tag{14}$$

Differentiating $b(\bar{\boldsymbol{x}} + \boldsymbol{e}_x, \bar{\boldsymbol{y}}_i + \boldsymbol{e}_i)$, we have

$$\frac{db(\bar{\boldsymbol{x}}+\boldsymbol{e}_x, \bar{\boldsymbol{y}}_i+\boldsymbol{e}_i)}{dt} = \frac{\partial b(\bar{\boldsymbol{x}}+\boldsymbol{e}_x, \bar{\boldsymbol{y}}_i+\boldsymbol{e}_i)}{\partial \bar{\boldsymbol{x}}}\dot{\bar{\boldsymbol{x}}} + \frac{\partial b(\bar{\boldsymbol{x}}+\boldsymbol{e}_x, \bar{\boldsymbol{y}}_i+\boldsymbol{e}_i)}{\partial \boldsymbol{e}_x}\dot{\boldsymbol{e}}_x + \frac{\partial b(\bar{\boldsymbol{x}}+\boldsymbol{e}_x, \bar{\boldsymbol{y}}_i+\boldsymbol{e}_i)}{\partial \bar{\boldsymbol{y}}_i}\dot{\bar{\boldsymbol{y}}}_i + \frac{\partial b(\bar{\boldsymbol{x}}+\boldsymbol{e}_x, \bar{\boldsymbol{y}}_i+\boldsymbol{e}_i)}{\partial \boldsymbol{e}_i}\dot{\boldsymbol{e}}_i \tag{15}$$

where $\dot{\boldsymbol{e}}_x = \dot{\boldsymbol{x}} - \dot{\bar{\boldsymbol{x}}}, \dot{\boldsymbol{e}}_i = \dot{\boldsymbol{y}}_i - \dot{\bar{\boldsymbol{y}}}_i$ are evaluated online through $\dot{\boldsymbol{x}}, \dot{\boldsymbol{y}}_i$ (from direct measurements of the actual state derivatives) and $\dot{\bar{\boldsymbol{x}}}, \dot{\bar{\boldsymbol{y}}}_i$ are given through (10), (11), respectively. Measurement uncertainties may be addressed as explained below.

Remark 1. *(Measurement uncertainties) If the measurements $\boldsymbol{x}$ and $\dot{\boldsymbol{x}}$ (or $\boldsymbol{y}_i$ and $\dot{\boldsymbol{y}}_i$) are subject to uncertainties, and the uncertainties are bounded, then we can apply the bounds of $\boldsymbol{x}$ and $\dot{\boldsymbol{x}}$ (or $\boldsymbol{y}_i$ and $\dot{\boldsymbol{y}}_i$) in evaluating the next event time t_{k+1} (introduced later) instead of $\boldsymbol{x}$ and $\dot{\boldsymbol{x}}$ (or $\boldsymbol{y}_i$ and $\dot{\boldsymbol{y}}_i$) themselves. In other words, $\boldsymbol{e}_x(t)$ and $\dot{\boldsymbol{e}}_x(t)$ ($\boldsymbol{e}_i(t)$ and $\dot{\boldsymbol{e}}_i(t)$) are determined by the bounds of $\boldsymbol{x}$, $\dot{\boldsymbol{x}}$ (or $\boldsymbol{y}_i$ and $\dot{\boldsymbol{y}}_i$) and the state values of the adaptive affine system (10) (or (11)).*

The CBF constraint that guarantees (9) for known dynamics (1) is as in (6), which is obtained by replacing $\dot{\boldsymbol{x}}$ with (1). However, for the unknown dynamics, the CBF constraint is:

$$\frac{db(\boldsymbol{x}, \boldsymbol{y}_i)}{dt} + \alpha_1(b(\boldsymbol{x}, \boldsymbol{y}_i)) \geq 0.$$

Equivalently, we have

$$\frac{db(\bar{\boldsymbol{x}}+\boldsymbol{e}_x, \bar{\boldsymbol{y}}_i+\boldsymbol{e}_i)}{dt} + \alpha_1(b(\bar{\boldsymbol{x}}+\boldsymbol{e}_x, \bar{\boldsymbol{y}}_i+\boldsymbol{e}_i)) \geq 0. \tag{16}$$

Combining (15), (16), (10) and (11), we get the CBF constraint that guarantees (9):

$$\frac{\partial b(\boldsymbol{x}, \boldsymbol{y}_i)}{\partial \bar{\boldsymbol{x}}} f_a(\bar{\boldsymbol{x}}) + \frac{\partial b(\boldsymbol{x}, \boldsymbol{y}_i)}{\partial \bar{\boldsymbol{x}}} g_a(\bar{\boldsymbol{x}})\boldsymbol{u} + \frac{\partial b(\boldsymbol{x}, \boldsymbol{y}_i)}{\partial \boldsymbol{e}_x}\dot{\boldsymbol{e}}_x + \frac{\partial b(\boldsymbol{x}, \boldsymbol{y}_i)}{\partial \bar{\boldsymbol{y}}_i} h_{a,i}(\bar{\boldsymbol{y}}_i) + \frac{\partial b(\boldsymbol{x}, \boldsymbol{y}_i)}{\partial \boldsymbol{e}_i}\dot{\boldsymbol{e}}_i + \alpha_1(b(\boldsymbol{x}, \boldsymbol{y}_i)) \geq 0. \tag{17}$$

Recall that we may have the control input showing up in $h_{a,i}(\bar{\boldsymbol{y}}_i)$, in which case the above CBF constraint (17) depends on both the control input $\boldsymbol{u}$ of the current controlled agent and the control $\boldsymbol{u}_i$ of agent $i \in S_a$. In other words, we have a multi-agent control problem that jointly determines all control inputs for safety guarantee, which can still be solved using the proposed framework (not considered in this paper). Then, the satisfaction of (17) implies the satisfaction of $b(\bar{\boldsymbol{x}} + \boldsymbol{e}_x, \bar{\boldsymbol{y}}_i + \boldsymbol{e}_i) \geq 0$ by Thm. 1 and (14). Thus, (9) is guaranteed to be satisfied for the real unknown dynamics of the controlled agent. For the above CBF constraint, we have the following assumption:

Assumption 2. $\frac{\partial b(\boldsymbol{x}, \boldsymbol{y}_i)}{\partial \bar{\boldsymbol{x}}} g_a(\bar{\boldsymbol{x}}) \neq \boldsymbol{0}, \forall \boldsymbol{x} \in X, \forall \boldsymbol{y}_i \in X$ *such that* $b(\boldsymbol{x}, \boldsymbol{y}_i) = 0$.

The above assumption may not be necessary for a safety-enforcing CBF, as the system trajectory with a CBF (HOCBF) for safety will never reach the safe set boundary, as shown in Lemma 1 of [30], if we define power class-$\mathcal{K}$ functions with power no less than 1 in the CBF and if the initial state is inside the safe set. If Assumption 2 does not hold and the CBF constraint (17) is not satisfied (without the $\boldsymbol{u}$ component) at the boundary of the set C_1, then we may shrink the safe set (if the manifold of $\{(\boldsymbol{x}, \boldsymbol{y}_i) | \frac{\partial b}{\partial \boldsymbol{x}} = 0\}$ does not cut through the safe set) or define another CBF so that the system avoids those states [31]; however, this approach can make the system conservative. We may also define a higher relative degree CBF than needed so as to address this problem. A more general non-conservative approach is the subject of future research.

Note that, if $\boldsymbol{y}_i$ in (9) and the dynamics (11) are known (such as the ACC example considered in case studies), we may just consider a single agent control problem, i.e. we just have $b(\boldsymbol{x})$ instead of $b(\boldsymbol{x}, \boldsymbol{y}_i)$, then we obtain the following simple version of the above CBF constraint:

$$\frac{\partial b(\boldsymbol{x})}{\partial \bar{\boldsymbol{x}}} f_a(\bar{\boldsymbol{x}}) + \frac{\partial b(\boldsymbol{x})}{\partial \bar{\boldsymbol{x}}} g_a(\bar{\boldsymbol{x}})\boldsymbol{u} + \frac{\partial b(\boldsymbol{x})}{\partial \boldsymbol{e}_x}\dot{\boldsymbol{e}}_x + \alpha_1(b(\boldsymbol{x})) \geq 0. \tag{18}$$

Now, we can formulate a CBF-based problem in the form:

$$\min_{\boldsymbol{u}(t), \delta(t)} \int_0^T \left(||\boldsymbol{u}(t)||^2 + p\delta^2(t)\right) dt \tag{19}$$

subject to (17), (2), and the relaxed CLF constraint

$$L_{f_a} V(\bar{\boldsymbol{x}}) + L_{g_a} V(\bar{\boldsymbol{x}})\boldsymbol{u} + \epsilon V(\bar{\boldsymbol{x}}) \leq \delta(t), \tag{20}$$

where $V(\bar{\boldsymbol{x}}) = (\bar{\boldsymbol{x}} - \boldsymbol{K})^T P(\bar{\boldsymbol{x}} - \boldsymbol{K})$, P is positive definite, $c_3 = \epsilon > 0$ in Def. 5, $p > 0$, and $\delta(t)$ is a relaxation for the CLF constraint. Since state convergence is relaxed in this section, so we just replace $\boldsymbol{x}$ by $\bar{\boldsymbol{x}}$ in the above CLF constraint.

Following the time-driven approach introduced at the end of Sec. II, we solve the problem (19) at time $t_k, k = 1, 2 \ldots$, and this problem is a QP. However, at time t_k, the QP does not generally know the error states $\boldsymbol{e}_x(t), \boldsymbol{e}_i(t)$ and their derivatives $\dot{\boldsymbol{e}}_x(t), \dot{\boldsymbol{e}}_i(t), \forall t > t_k$. Thus, it cannot guarantee that the CBF constraint (17) is satisfied in the time interval $(t_k, t_{k+1}]$, where t_{k+1} is the next time instant to solve the QP. This is what motivates the introduction of *events* defining the conditions needed to guarantee the satisfaction of (17) $\forall t \in (t_k, t_{k+1}]$ We start by imposing bounds on $\boldsymbol{e}_x = (e_{x,1}, \ldots, e_{x,n})$ and $\dot{\boldsymbol{e}}_x = (\dot{e}_{x,1}, \ldots, \dot{e}_{x,n})$ defined as $\boldsymbol{w} = (w_1, \ldots, w_n) \in \mathbb{R}^n_{>0}$ and $\boldsymbol{\nu} = (\nu_1, \ldots, \nu_n) \in \mathbb{R}^n_{>0}$:

$$|e_{x,j}| \leq w_j, \qquad |\dot{e}_{x,j}| \leq \nu_j, \qquad j \in \{1, \ldots, n\}. \tag{21}$$

These two inequalities can be rewritten in the form $|\boldsymbol{e}_x| \leq \boldsymbol{w}, |\dot{\boldsymbol{e}}_x| \leq \boldsymbol{\nu}$ for simplicity. Similarly, we also bound $\boldsymbol{e}_i, \dot{\boldsymbol{e}}_i$ by

$$|\boldsymbol{e}_i| \leq \boldsymbol{W}_i, \qquad |\dot{\boldsymbol{e}}_i| \leq \boldsymbol{V}_i, \qquad i \in S_a. \tag{22}$$

where the inequalities are interpreted componentwise.

Similar to the bounds we introduced for the error states and their derivatives, we also define the following bounds on the deviations of states from their values $\bar{\boldsymbol{x}}(t_k), \bar{\boldsymbol{y}}_i(t_k), i \in S_a$:

$$\begin{aligned} \bar{\boldsymbol{x}}(t_k) - \boldsymbol{s}(\beta_1(b(\boldsymbol{x}, \boldsymbol{y}_i))) \leq \bar{\boldsymbol{x}} \leq \bar{\boldsymbol{x}}(t_k) + \boldsymbol{s}(\beta_1(b(\boldsymbol{x}, \boldsymbol{y}_i))), \\ \bar{\boldsymbol{y}}_i(t_k) - \boldsymbol{S}_i(\beta_2(b(\boldsymbol{x}, \boldsymbol{y}_i))) \leq \bar{\boldsymbol{y}}_i \leq \bar{\boldsymbol{y}}_i(t_k) + \boldsymbol{S}_i(\beta_2(b(\boldsymbol{x}, \boldsymbol{y}_i))), \end{aligned} \tag{23}$$

where the inequalities are interpreted componentwise, $\boldsymbol{s} : \mathbb{R} \to \mathbb{R}^n_{>0}, \boldsymbol{S}_i : \mathbb{R} \to \mathbb{R}^n_{>0}$, and $\beta_1(\cdot), \beta_2(\cdot)$ are class $\mathcal{K}$ functions.

For simplicity, we can just use constant vectors $\boldsymbol{s} \in \mathbb{R}^n, \boldsymbol{S}_i \in \mathbb{R}^n$ in the above. Their relative advantages and the choice of $\boldsymbol{s}(\cdot), \boldsymbol{S}_i(\cdot)$ will be discussed later. We denote the set of states that satisfy (23) at time t_k by

$$S_x(t_k) = \{\boldsymbol{y}_1 \in X : \bar{\boldsymbol{x}}(t_k) - \boldsymbol{s}(\beta_1(b(\boldsymbol{x}(t_k), \boldsymbol{y}_i(t_k)))) \leq \boldsymbol{y}_1 \leq \bar{\boldsymbol{x}}(t_k) + \boldsymbol{s}(\beta_1(b(\boldsymbol{x}(t_k), \boldsymbol{y}_i(t_k)))), i \in S_a\}. \tag{24}$$

$$S_{y,i}(t_k) = \{\boldsymbol{y}_2 \in \mathbb{R}^n : \bar{\boldsymbol{y}}_i(t_k) - \boldsymbol{S}_i(\beta_2(b(\boldsymbol{x}(t_k), \boldsymbol{y}_i(t_k)))) \leq \boldsymbol{y}_2 \leq \bar{\boldsymbol{y}}_i(t_k) + \boldsymbol{S}_i(\beta_2(b(\boldsymbol{x}(t_k), \boldsymbol{y}_i(t_k))))\}. \tag{25}$$

Now, with (21), (22) and (23), we are ready to find a condition that guarantees the satisfaction of (17) in the time interval $(t_k, t_{k+1}]$. This is done by considering the minimum value of each component in (17), as shown next.

Let $\boldsymbol{z} := (\boldsymbol{y}_1, \boldsymbol{e}_x, \dot{\boldsymbol{e}}_x, \boldsymbol{y}_2, \boldsymbol{e}_i, \dot{\boldsymbol{e}}_i)$, where $\boldsymbol{y}_1 \in S_x(t_k), \boldsymbol{y}_2 \in S_{y,i}(t_k)$. We then define an overall set $S(t_k)$:

$$S(t_k) = \{\boldsymbol{z} \in \mathbb{R}^{6n} : \boldsymbol{y}_1 \in S_x(t_k), |\boldsymbol{e}_x| \leq \boldsymbol{w}, |\dot{\boldsymbol{e}}_x| \leq \boldsymbol{\nu}, \boldsymbol{y}_2 \in S_{y,i}(t_k), |\boldsymbol{e}_i| \leq \boldsymbol{W}_i, |\dot{\boldsymbol{e}}_i| \leq \boldsymbol{V}_i, (\boldsymbol{y}_1 + \boldsymbol{e}_x, \boldsymbol{y}_2 + \boldsymbol{e}_i) \in C_1, i \in S_a\}. \tag{26}$$

Consider the first term in (17) and let $b_{f_a,min}(t_k) \in \mathbb{R}$ be the minimum value of $\frac{\partial b(\bar{\boldsymbol{x}}+\boldsymbol{e}_x, \bar{\boldsymbol{y}}_i+\boldsymbol{e}_i)}{\partial \bar{\boldsymbol{x}}} f_a(\bar{\boldsymbol{x}})$ for the preceding time interval that satisfies $\boldsymbol{z} \in S(t_k)$ starting at time t_k, i.e., let

$$b_{f_a,min}(t_k) = \min_{\boldsymbol{z} \in S(t_k)} \frac{\partial b(\boldsymbol{y}_1 + \boldsymbol{e}_x, \boldsymbol{y}_2 + \boldsymbol{e}_i)}{\partial \boldsymbol{y}_1} f_a(\boldsymbol{y}_1). \tag{27}$$

Similarly, we can also find the four minimum values $b_{\alpha_1,min}(t_k) \in \mathbb{R}, b_{e,min}(t_k) \in \mathbb{R}, b_{h_{a,i},min}(t_k) \in \mathbb{R}, b_{E_i,min}(t_k) \in \mathbb{R}$ of $\alpha_1(b(\boldsymbol{x}, \boldsymbol{y}_i)), \frac{\partial b(\boldsymbol{x},\boldsymbol{y}_i)}{\partial \boldsymbol{e}_x}\dot{\boldsymbol{e}}_x, \frac{\partial b(\boldsymbol{x},\boldsymbol{y}_i)}{\partial \bar{\boldsymbol{y}}_i} h_{a,i}(\bar{\boldsymbol{y}}_i), \frac{\partial b(\boldsymbol{x},\boldsymbol{y}_i)}{\partial \boldsymbol{e}_i}\dot{\boldsymbol{e}}_i, i \in S_a$, respectively, for the preceding time interval $[t_k, t_{k+1}]$ that satisfy $\boldsymbol{z} \in S(t_k)$ starting at time t_k, i.e., let

$$b_{\alpha_1,min}(t_k) = \min_{\boldsymbol{z} \in S(t_k)} \alpha_1(b(\boldsymbol{y}_1 + \boldsymbol{e}_x, \boldsymbol{y}_2 + \boldsymbol{e}_i)), \tag{28}$$

$$b_{e,min}(t_k) = \min_{\boldsymbol{z} \in S(t_k)} \frac{\partial b(\boldsymbol{y}_1 + \boldsymbol{e}_x, \boldsymbol{y}_2 + \boldsymbol{e}_i)}{\partial \boldsymbol{e}_x} \dot{\boldsymbol{e}}_x, \tag{29}$$

$$b_{h_{a,i},min}(t_k) = \min_{\boldsymbol{z} \in S(t_k)} \frac{\partial b(\boldsymbol{y}_1 + \boldsymbol{e}_x, \boldsymbol{y}_2 + \boldsymbol{e}_i)}{\partial \boldsymbol{y}_2} h_{a,i}(\boldsymbol{y}_2), \tag{30}$$

$$b_{E_i,min}(t_k) = \min_{\boldsymbol{z} \in S(t_k)} \frac{\partial b(\boldsymbol{y}_1 + \boldsymbol{e}_x, \boldsymbol{y}_2 + \boldsymbol{e}_i)}{\partial \boldsymbol{e}_i} \dot{\boldsymbol{e}}_i, \tag{31}$$

The above optimizations may be nonlinear programs (NLP). If the safety constraints are linear, then each optimization is just a LP or QP. However, note that even the NLPs are easy to solve since the constraints are mostly linear, as shown in the case studies of Section VI.

This leaves only one remaining term in (17): if $\frac{\partial b(\boldsymbol{x},\boldsymbol{y}_i)}{\partial \bar{\boldsymbol{x}}} g_a(\bar{\boldsymbol{x}})$ is independent of $\bar{\boldsymbol{x}}, \boldsymbol{e}_x, \bar{\boldsymbol{y}}_i, \boldsymbol{e}_i$, then we do not need to find its limit value within the bound $\boldsymbol{z} \in S(t_k)$; otherwise, let $\bar{\boldsymbol{x}} = (\bar{x}_1, \dots, \bar{x}_n) \in \mathbb{R}^n, \boldsymbol{u} = (u_1, \dots, u_q) \in \mathbb{R}^q$ and $g_a = (g_1, \dots, g_q) \in \mathbb{R}^{n \times q}$. The sign of $u_j(t_k), j \in \{1, \dots, q\}, k = 1, 2 \dots$ can be determined by solving the CBF-based QP (19) at time t_k. We can then determine the limit value $b_{g_j,lim}(t_k) \in \mathbb{R}, j \in \{1, \dots, q\}$ of $\frac{\partial b(\boldsymbol{x},\boldsymbol{y}_i)}{\partial \bar{\boldsymbol{x}}} g_j(\bar{\boldsymbol{x}})$ by

$$b_{g_j,lim}(t_k) = \begin{cases} \min\limits_{\boldsymbol{z} \in S(t_k)} \frac{\partial b(\boldsymbol{y}_1+\boldsymbol{e}_x,\boldsymbol{y}_2+\boldsymbol{e}_i)}{\partial \boldsymbol{y}_1} g_j(\boldsymbol{y}_1), & \text{if } u_j(t_k) \geq 0, \\ \max\limits_{\boldsymbol{z} \in S(t_k)} \frac{\partial b(\boldsymbol{y}_1+\boldsymbol{e}_x,\boldsymbol{y}_2+\boldsymbol{e}_i)}{\partial \boldsymbol{y}_1} g_j(\boldsymbol{y}_1), & \text{otherwise} \end{cases} \tag{32}$$

Let $b_{g_a,lim}(t_k) = (b_{g_1,lim}(t_k), \dots, b_{g_q,lim}(t_k)) \in \mathbb{R}^q$, and we set $b_{g_a,lim}(t_k) = \frac{\partial b(\boldsymbol{x},\boldsymbol{y}_i)}{\partial \bar{\boldsymbol{x}}} g_a(\bar{\boldsymbol{x}})$ if $\frac{\partial b(\boldsymbol{x},\boldsymbol{y}_i)}{\partial \bar{\boldsymbol{x}}} g(\bar{\boldsymbol{x}})$ is independent of $\bar{\boldsymbol{x}}, \boldsymbol{e}_x, \bar{\boldsymbol{y}}_i, \boldsymbol{e}_i$ for notational simplicity.

To sum up, the condition that guarantees the satisfaction of (17) in the time interval $(t_k, t_{k+1}]$ is given by

$$b_{f_a,min}(t_k) + b_{g_a,lim}(t_k)\boldsymbol{u}(t_k) + b_{e_x,min}(t_k) + b_{h_{a,i},min}(t_k) + b_{e_i,min}(t_k) + b_{\alpha_1,min}(t_k) \geq 0. \tag{33}$$

We would also like to add the sign condition to the set $S(t_k)$ in order to make this framework work as shown in (32), i.e., we add $\frac{\partial b(\boldsymbol{x},\boldsymbol{y}_i)}{\partial \bar{\boldsymbol{x}}} g_i(\bar{\boldsymbol{x}}) \geq 0$ if $\frac{\partial b(\boldsymbol{x},\boldsymbol{y}_i)}{\partial \bar{\boldsymbol{x}}} g_i(\bar{\boldsymbol{x}})$ is positive at time t_k, and vice versa. Note that, despite Assumption 2, it is still possible that $b_{g_a,lim}(t_k) = \boldsymbol{0}$ by (32) as we consider all possible state $\boldsymbol{z} \in S(t_k)$. If (33) is satisfied even without the control component, then the safety is still guaranteed even if we do not consider (33) in the QP. Otherwise, we can deal with it with the approaches discussed after Assumption 2. In this paper, we tune the parameters of the CBF (i.e., how to define a class-$\mathcal{K}$ function of a CBF) and change the reference path (e.g., changing the parameters (such as c_3 in Def. 5) of a CLF) to avoid these possible "singular" states.

In order to apply the above condition (33) to the problem (19), we just replace (17) by (33) and consider the problem at time t_k, i.e., we have a QP:

$$\min_{\boldsymbol{u}(t_k), \delta(t_k)} ||\boldsymbol{u}(t_k)||^2 + p\delta^2(t_k) \tag{34}$$

subject to (33), (2) and (20). The feasibility of the above QPs can be guaranteed by finding a suitable feasibility constraint, as shown in [32]; briefly, we determine sufficient conditions for feasibility and then enforce them by using another CBF.

Based on the above, we define seven events that determine the condition that triggers an instance of solving the QP (34):

- **Event 1:** $|\boldsymbol{e}_x| \leq \boldsymbol{w}$ is about to be violated.
- **Event 2:** $|\dot{\boldsymbol{e}}_x| \leq \boldsymbol{\nu}$ is about to be violated.
- **Event 3:** the state of (10) reaches the boundaries of $S_x(t_k)$ in (24).
- **Event 4:** $|\boldsymbol{e}_i| \leq \boldsymbol{W}_i$ is about to be violated.
- **Event 5:** $|\dot{\boldsymbol{e}}_i| \leq \boldsymbol{V}_i$ is about to be violated.
- **Event 6:** the state of (11) reaches the boundaries of $S_{y,i}(t_k)$ in (25).
- **Event 7:** $\frac{\partial b(\boldsymbol{x},\boldsymbol{y}_i)}{\partial \bar{\boldsymbol{x}}} g_j(\bar{\boldsymbol{x}}), j \in \{1, \dots, q\}$ changes sign for $t > t_k$ compared to the sign it had at t_k.

In other words, the next time instant $t_{k+1}, k = 1, 2 \dots$ to solve the QP (34) is determined by:

$$\begin{aligned} t_{k+1} = \min\{t > t_k : |\boldsymbol{e}_x(t)| = \boldsymbol{w} \text{ or } |\dot{\boldsymbol{e}}_x(t)| = \boldsymbol{\nu} \\ \text{or } |\bar{\boldsymbol{x}}(t) - \bar{\boldsymbol{x}}(t_k)| = \boldsymbol{s}(\beta_1(b(\boldsymbol{x}(t_k), \boldsymbol{y}_i(t_k)))) \text{ or } |\boldsymbol{e}_i(t)| = \boldsymbol{W}_i \\ \text{or } |\dot{\boldsymbol{e}}_i(t)| = \boldsymbol{V}_i \text{ or } \frac{\partial b(\boldsymbol{x}, \boldsymbol{y}_i)}{\partial \bar{\boldsymbol{x}}} g_j(\bar{\boldsymbol{x}}), j \in \{1, \dots, q\} \text{ changes sign} \\ \text{or } |\bar{\boldsymbol{y}}_i(t) - \bar{\boldsymbol{y}}_i(t_k)| = \boldsymbol{S}_i(\beta_2(b(\boldsymbol{x}(t_k), \boldsymbol{y}_i(t_k)))), i \in S_a\}, \end{aligned} \tag{35}$$

where $t_1 = 0$. Events 1, 2, 4, 5, 7 can be detected by direct sensor measurements, while Events 3, 6 can be detected by monitoring the dynamics (10), (11). The magnitude of each component of $\boldsymbol{s}(\beta_1(b(\boldsymbol{x}(t_k), \boldsymbol{y}_i(t_k)))), \boldsymbol{S}_i(\beta_2(b(\boldsymbol{x}(t_k), \boldsymbol{y}_i(t_k))))$ (as well as other bounds) is selected to capture a tradeoff between the time complexity and the conservativeness of this approach: if the magnitude is large, then the number of events is small but this approach is considerably conservative as we determine the condition (33) through the minimum values as in (27)-(32). If $\boldsymbol{s}(\cdot), \boldsymbol{S}_i(\cdot)$ are some constants, then the control system will be too conservative when its state reaches the boundary of the set C_1 as the states of (10) and (11) change slowly at the boundary. However, when the system states are far from the boundary of C_1, i.e., $b(\boldsymbol{x}(t_k), \boldsymbol{y}_i(t_k))$ takes some large value, this again will make the control system too conservative if we take $\boldsymbol{s}(\cdot), \boldsymbol{S}_i(\cdot)$ as a function of $b(\boldsymbol{x}(t_k), \boldsymbol{y}_i(t_k))$. Therefore, we wish to truncate both $s(\cdot), \boldsymbol{S}_i(\cdot)$ in the form:

$$\boldsymbol{s}(\beta_1(b(\boldsymbol{x}(t_k), \boldsymbol{y}_i(t_k)))) = \begin{cases} \boldsymbol{s}_0, \text{ if } \boldsymbol{s}(\beta_1(b(\boldsymbol{x}(t_k), \boldsymbol{y}_i(t_k)))) \geq \boldsymbol{s}_0, \\ \boldsymbol{s}(\beta_1(b(\boldsymbol{x}(t_k), \boldsymbol{y}_i(t_k)))), \text{ otherwise.} \end{cases} \tag{36}$$

$$\boldsymbol{S}_i(\beta_2(b(\boldsymbol{x}(t_k), \boldsymbol{y}_i(t_k)))) = \begin{cases} \boldsymbol{S}_0, \text{ if } \boldsymbol{S}_i(\beta_2(b(\boldsymbol{x}(t_k), \boldsymbol{y}_i(t_k)))) \geq \boldsymbol{S}_0, \\ \boldsymbol{S}_i(\beta_2(b(\boldsymbol{x}(t_k), \boldsymbol{y}_i(t_k)))), \text{ otherwise.} \end{cases} \tag{37}$$

where $\boldsymbol{s}_0 \in \mathbb{R}^n, \boldsymbol{S}_0 \in \mathbb{R}^n$.

Formally, we have the following theorem to show that the satisfaction of the safety constraint (9) is guaranteed for the real unknown dynamics under condition (33):

Theorem 2. *Given a HOCBF $b(\boldsymbol{x})$ with $m = 1$ as in Def. 4, let $t_{k+1}, k = 1, 2 \ldots$ be determined by (35) with $t_1 = 0$, and (33) be determined by (27)-(32), respectively. Then, under Assumptions 1-2, any control $\boldsymbol{u}(t_k)$ that satisfies (33) and updates the real unknown dynamics and the adaptive dynamics (10) within time interval $[t_k, t_{k+1})$ renders the set C_1 forward invariant for the real unknown dynamics.*

Proof: By (35), we have that

$$\bar{\boldsymbol{x}}(t) \in S_x(t_k), |\boldsymbol{e}_x(t)| \leq \boldsymbol{w}, |\dot{\boldsymbol{e}}_x(t)| \leq \boldsymbol{\nu}, (\boldsymbol{x}(t), \boldsymbol{y}_i(t)) \in C_1, \\ \bar{\boldsymbol{y}}_i(t) \in S_{y,i}(t_k), |\boldsymbol{e}_i(t)| \leq \boldsymbol{W}_i, |\dot{\boldsymbol{e}}_i(t)| \leq \boldsymbol{V}_i,$$

for all $t \in [t_k, t_{k+1}], k = 1, 2 \ldots$. Thus, the limit values $b_{f_a,min}(t_k), b_{\alpha_1,min}(t_k), b_{e,min}(t_k), b_{h_{a,i},min}(t_k), b_{E_i,min}(t_k)$, determined by (27)-(31), respectively, are the minimum values for $\frac{\partial b(\boldsymbol{x},\boldsymbol{y}_i)}{\partial \bar{\boldsymbol{x}}} f_a(\bar{\boldsymbol{x}}(t)), \alpha_1(b(\boldsymbol{x}(t), \boldsymbol{y}_i(t))), \frac{\partial b(\boldsymbol{x},\boldsymbol{y}_i)}{\partial \boldsymbol{e}_x} \dot{\boldsymbol{e}}_x(t)$, $\frac{\partial b(\boldsymbol{x},\boldsymbol{y}_i)}{\partial \bar{\boldsymbol{y}}_i} h_{a,i}(\bar{\boldsymbol{y}}_i(t)), \frac{\partial b(\boldsymbol{x},\boldsymbol{y}_i)}{\partial \boldsymbol{e}_i} \dot{\boldsymbol{e}}_i(t)$, respectively, for all $t \in [t_k, t_{k+1})$. In other words, we have

$$\frac{\partial b(\boldsymbol{x}, \boldsymbol{y}_i)}{\partial \bar{\boldsymbol{x}}} f_a(\bar{\boldsymbol{x}}(t)) + \alpha_1(b(\boldsymbol{x}(t), \boldsymbol{y}_i(t))) + \frac{\partial b(\boldsymbol{x}, \boldsymbol{y}_i)}{\partial \boldsymbol{e}_x} \dot{\boldsymbol{e}}_x(t) \\ + \frac{\partial b(\boldsymbol{x}, \boldsymbol{y}_i)}{\partial \bar{\boldsymbol{y}}_i} h_{a,i}(\bar{\boldsymbol{y}}_i(t)) + \frac{\partial b(\boldsymbol{x}, \boldsymbol{y}_i)}{\partial \boldsymbol{e}_i} \dot{\boldsymbol{e}}_i(t) \geq b_{f_a,min}(t_k) \\ + b_{\alpha_1,min}(t_k) + b_{e,min}(t_k) + b_{h_{a,i},min}(t_k) + b_{E_i,min}(t_k), \\ \forall t \in [t_k, t_{k+1}).$$

where $\boldsymbol{x} = \boldsymbol{e}_x + \bar{\boldsymbol{x}}, \boldsymbol{y}_i = \boldsymbol{e}_i + \bar{\boldsymbol{y}}_i$.

By (32), we have that

$$\frac{\partial b(\boldsymbol{x}, \boldsymbol{y}_i)}{\partial \bar{\boldsymbol{x}}} g_a(\bar{\boldsymbol{x}}(t)) \boldsymbol{u}(t_k) \geq b_{g_a,lim}(t_k) \boldsymbol{u}(t_k), \forall t \in [t_k, t_{k+1}).$$

Following the last two equations and (33), we have

$$\frac{\partial b(\boldsymbol{x}, \boldsymbol{y}_i)}{\partial \bar{\boldsymbol{x}}} f_a(\bar{\boldsymbol{x}}(t)) + \frac{\partial b(\boldsymbol{x}, \boldsymbol{y}_i)}{\partial \bar{\boldsymbol{x}}} g_a(\bar{\boldsymbol{x}}(t)) \boldsymbol{u}(t_k) + \alpha_1(b(\boldsymbol{x}(t), \boldsymbol{y}_i(t))) \\ + \frac{\partial b(\boldsymbol{x}, \boldsymbol{y}_i)}{\partial \boldsymbol{e}_x} \dot{\boldsymbol{e}}_x(t) + \frac{\partial b(\boldsymbol{x}, \boldsymbol{y}_i)}{\partial \bar{\boldsymbol{y}}_i} h_{a,i}(\bar{\boldsymbol{y}}_i(t)) + \frac{\partial b(\boldsymbol{x}, \boldsymbol{y}_i)}{\partial \boldsymbol{e}_i} \dot{\boldsymbol{e}}_i(t) \geq 0, \\ \forall t \in [t_k, t_{k+1}).$$

By (16), (17) and the last equation, we have that

$$\frac{db(\bar{\boldsymbol{x}}(t) + \boldsymbol{e}_x(t), \bar{\boldsymbol{y}}_i(t) + \boldsymbol{e}_i(t))}{dt} + \alpha_1(b(\bar{\boldsymbol{x}}(t) + \boldsymbol{e}_x(t), \bar{\boldsymbol{y}}_i(t) + \boldsymbol{e}_i(t))) \\ \geq 0, \forall t \in [t_k, t_{k+1}), k = 1, 2 \ldots$$

By Thm. 1, we have that $b(\bar{\boldsymbol{x}}(t) + \boldsymbol{e}_x(t), \bar{\boldsymbol{y}}_i(t) + \boldsymbol{e}_i(t)) \geq 0, \forall t \geq 0$ and by (14), we have that C_1 is forward invariant for the real unknown dynamics. ■

Remark 2. *We could also consider the minimum value of $\frac{\partial b(\boldsymbol{y}_1+\boldsymbol{e}_x, \boldsymbol{y}_2+\boldsymbol{e}_i)}{\partial \boldsymbol{y}_1} f_a(\boldsymbol{y}_1) + \frac{\partial b(\boldsymbol{y}_1+\boldsymbol{e}_x, \boldsymbol{y}_2+\boldsymbol{e}_i)}{\partial \boldsymbol{e}_x} \dot{\boldsymbol{e}}_x + \alpha_1(b(\boldsymbol{y}_1 + \boldsymbol{e}_x, \boldsymbol{y}_2 + \boldsymbol{e}_i)) + \frac{\partial b(\boldsymbol{y}_1+\boldsymbol{e}_x, \boldsymbol{y}_2+\boldsymbol{e}_i)}{\partial \boldsymbol{y}_2} h_{a,i}(\boldsymbol{y}_2) + \frac{\partial b(\boldsymbol{y}_1+\boldsymbol{e}_x, \boldsymbol{y}_2+\boldsymbol{e}_i)}{\partial \boldsymbol{e}_i} \dot{\boldsymbol{e}}_i$ within the bound $\boldsymbol{z} \in S(t_k)$ instead of considering them separately as in (27)-(32). This will be less conservative (but more computationally expensive) as the constraint (33) is stronger compared with the CBF constraint (17), and we wish to find the largest possible value of the left-hand side of (17) that can support Thm. 2.*

Events 1 and 2 will be frequently triggered if the modelling of the adaptive affine dynamics (10) has a large error with respect to the real dynamics of the control system and the same is true for Events 4 and 5. Therefore, we would like to model the adaptive affine dynamics (10) and the adaptive dynamics (11) as accurately as possible in order to reduce the number of events required to solve the QP (34).

State synchronization and adaptation of dynamics: An additional important step is to synchronize the state of the real unknown dynamics of the control system with the state of the adaptive affine dynamics in (10) such that we always have $\boldsymbol{e}_x(t_k) = 0$ and make $\dot{\boldsymbol{e}}_x(t_k)$ close to 0. This is done by setting

$$\bar{\boldsymbol{x}}(t_k) = \boldsymbol{x}(t_k), \tag{38}$$

and by updating $f_a(\bar{\boldsymbol{x}}(t))$ in the adaptive affine dynamics (10) right after (t^+) an event occurs at t:

$$f_a(\bar{\boldsymbol{x}}(t^+)) = f_a(\bar{\boldsymbol{x}}(t^-)) + \sum_{i=0}^{k} \dot{\boldsymbol{e}}_x(t_i). \tag{39}$$

where t^+, t^- denote instants right after and before t. In this way, the dynamics (10) are adaptively updated at each event, i.e., at $t_k, k = 1, 2, \ldots$. Note that we may also update $g_a(\cdot)$, which is harder than updating $f_a(\cdot)$ since $g_a(\cdot)$ is multiplied by $\boldsymbol{u}$ that is to be determined, i.e., the update of $g_a(\cdot)$ will depend on $\boldsymbol{u}$. We do not consider the update of $g_a(\cdot)$ in this paper, but it does not diminish the validity of the approach. This possibility is the subject of ongoing work.

Along the same lines, we also synchronize the state of agent $i \in S_a$ and (11), by updating $h_{a,i}(\bar{\boldsymbol{y}}_i(t))$ of the adaptive dynamics (11) right after an event occurs at t (i.e., at t^+):

$$\bar{\boldsymbol{y}}_i(t_k) = \boldsymbol{y}_i(t_k), \tag{40}$$

$$h_{a,i}(\bar{\boldsymbol{y}}_i(t^+)) = h_{a,i}(\bar{\boldsymbol{y}}_i(t^-)) + \sum_{j=0}^{k} \dot{\boldsymbol{e}}_i(t_j). \tag{41}$$

The control obtained by solving a traditional CBF-based QP is Lipschitz if there are no control bounds (2) [33]. However, control bounds exist in our formulation, which is an event-triggered QP. Even so, the safety is still guaranteed in the proposed framework and there exists a minimum inter-event time, as shown next. By (38) and (39), we have that $\boldsymbol{e}_x(t_k) = 0$ and $\dot{\boldsymbol{e}}_x(t_k)$ is close to 0. There exist lower bounds for the occurrence times of Event 1 and Event 3 (the same applies to Events 4 and 6) as the controls are bounded, and they are determined by the limit values of the component of f_a, g_a within X and U, as well as the real unknown dynamics. Assuming the functions that define the real unknown dynamics are Lipschitz continuous, and the functions f_a, g_a in (10) are also assumed to be Lipschitz continuous, it follows that $\dot{\boldsymbol{e}}_x$ is also Lipschitz continuous since $\boldsymbol{u} \in U$. Suppose the largest Lipschitz constant among all the components in $\dot{\boldsymbol{x}}$ is $L, \forall \boldsymbol{x} \in X$, and the smallest Lipschitz constant among all the components in $\dot{\bar{\boldsymbol{x}}}$ is $\bar{L}, \forall \bar{\boldsymbol{x}} \in X$, then the lower bound time for Event 2 is $\frac{\nu_{min}}{L-\bar{L}}$, where $\nu_{min} > 0$ is the minimum component in $\boldsymbol{\nu}$. Similarly, there is also a lower bound time for Events 5 and 7. On the other hand, since the control is constant in each time interval, and there exists a lower bound for the event time, Zeno behavior will not occur. We summarize the event-triggered control scheme in Alg. 1.

Algorithm 1: Event-triggered control

Input: Measurements $\boldsymbol{x}$ and $\dot{\boldsymbol{x}}$ from the controlled agent, measurements $\boldsymbol{y}$ and $\dot{\boldsymbol{y}}_i$ from agent $i \in S_a$, adaptive affine model (10), adaptive model (11), settings for QP (34), $\boldsymbol{w}, \boldsymbol{\nu}, \boldsymbol{s}(\cdot), \boldsymbol{W}_i, \boldsymbol{V}_i, \boldsymbol{S}(\cdot)$.
Output: Event time $t_k, k = 1, 2, \dots$ and $\boldsymbol{u}^*(t_k)$.

```
k = 1, t_k = 0;
while t_k ≤ T do
    Measure x and ẋ from the controlled agent at t_k;
    Measure y and ẏ_i from agent i ∈ S_a at t_k;
    Sync. the state of (10) and the controlled agent by (38),(39);
    Sync. the state of (11) and agent i by (40),(41);
    Evaluate (27)-(32);
    Solve the QP (34) at t_k and get u*(t_k);
    while t ≤ T do
        Apply u*(t_k) to the controlled agent and (10) for t ≥ t_k;
        Measure x and ẋ from the controlled agent;
        Measure y and ẏ_i from agent i ∈ S_a;
        Evaluate t_{k+1} by (35);
        if t_{k+1} is found with ε > 0 error (i.e., the states reach the bounds with ε error in (35)) then
            k ← k + 1, break;
        end
    end
end
```

B. High-relative-degree Constraints

In this subsection, we consider the safety constraint (9) whose relative degree is larger than one with respect to the real unknown dynamics and (10). In other words, we need to consider the HOCBF constraint (5) to find the state feedback control with the HOCBF method. To adapt to the safety constraint (9) that involves other agents with states $\boldsymbol{y}_i$, we have the following definitions for the sequence of functions and sets corresponding to (3) and (4):

$$\psi_j(\boldsymbol{x}, \boldsymbol{y}_i) := \dot{\psi}_{j-1}(\boldsymbol{x}, \boldsymbol{y}_i) + \alpha_j(\psi_{j-1}(\boldsymbol{x}, \boldsymbol{y}_i)), j \in \{1, \dots, m\}, \tag{42}$$

$$C_j := \{(\boldsymbol{x}, \boldsymbol{y}_i) \in X \times X : \psi_{j-1}(\boldsymbol{x}, \boldsymbol{y}_i) \geq 0\}, j \in \{1, \dots, m\}. \tag{43}$$

where $\psi_0(\boldsymbol{x}, \boldsymbol{y}_i) = b(\boldsymbol{x}, \boldsymbol{y}_i)$.

Similar to the last subsection, we find the error state $\boldsymbol{e}_x$ and $\boldsymbol{e}_i$ by (13), and obtain an alternative form of the HOCBF $b(\boldsymbol{x}, \boldsymbol{y}_i)$ as in (14). The HOCBF constraint (5) that guarantees $b(\bar{\boldsymbol{x}} + \boldsymbol{e}_x, \bar{\boldsymbol{y}}_i + \boldsymbol{e}_i) \geq 0$ with respect to the real unknown dynamics is

$$\begin{aligned} &\frac{\partial^m b(\boldsymbol{x}, \boldsymbol{y}_i)}{\partial \bar{\boldsymbol{x}}^m} f_a^{[m]}(\bar{\boldsymbol{x}}) + \frac{\partial^m b(\boldsymbol{x}, \boldsymbol{y}_i)}{\partial \bar{\boldsymbol{x}}^m} f_a^{[m-1]}(\bar{\boldsymbol{x}}) g_a^{[1]}(\bar{\boldsymbol{x}}) \boldsymbol{u} \\ &+ \frac{\partial^m b(\boldsymbol{x}, \boldsymbol{y}_i)}{\partial \boldsymbol{e}_x^m} \boldsymbol{e}_x^{(m)} + \frac{\partial^m b(\boldsymbol{x}, \boldsymbol{y}_i)}{\partial \bar{\boldsymbol{y}}_i^m} h_{a,i}^{[m]}(\bar{\boldsymbol{y}}_i) + \frac{\partial^m b(\boldsymbol{x}, \boldsymbol{y}_i)}{\partial \boldsymbol{e}_i^m} \boldsymbol{e}_i^{(m)} \\ &+ R(b(\boldsymbol{x}, \boldsymbol{y}_i)) + \alpha_m(\psi_{m-1}(\boldsymbol{x}, \boldsymbol{y}_i)) \geq 0, \end{aligned} \tag{44}$$

where $\frac{\partial^m b(\boldsymbol{x}, \boldsymbol{y}_i)}{\partial \bar{\boldsymbol{x}}^m} f_a^{[m]}(\bar{\boldsymbol{x}})$ denotes the m-time partial derivative of $b(\boldsymbol{x}, \boldsymbol{y}_i)$ with respect to $\bar{\boldsymbol{x}}$ along $f_a(\bar{\boldsymbol{x}})$ (similar concept to the Lie derivative in Def. 4), and we have similar definitions for $\frac{\partial^m b(\boldsymbol{x}, \boldsymbol{y}_i)}{\partial \bar{\boldsymbol{x}}^m} f_a^{[m-1]} g_a^{[1]}(\bar{\boldsymbol{x}}) \boldsymbol{u}$ and $\frac{\partial^m b(\boldsymbol{x}, \boldsymbol{y}_i)}{\partial \bar{\boldsymbol{y}}_i^m} h_{a,i}^{[m]}(\bar{\boldsymbol{y}}_i)$. $R(b(\boldsymbol{x}, \boldsymbol{y}_i))$ also contains the remaining time derivatives of $\boldsymbol{e}_x$ and $\boldsymbol{e}_i$ with degree less than m. $\boldsymbol{e}_x^{(j)} = \boldsymbol{x}^{(j)} - \bar{\boldsymbol{x}}^{(j)}, \boldsymbol{e}_i^{(j)} = \boldsymbol{y}_i^{(j)} - \bar{\boldsymbol{y}}_i^{(j)}, j \in \{1, \dots, m\}$ is the j^{th} derivative and is evaluated online by $\boldsymbol{x}^{(j)}$ (from a sensor) of the real system and $\bar{\boldsymbol{x}}^{(j)}$ of (10), and by $\boldsymbol{y}_i^{(j)}$ (from a sensor) of system $i \in S_a$ and $\bar{\boldsymbol{y}}_i^{(j)}$ of (11), respectively. Note that it is difficult to measure high-order state derivatives. However, we do not need to know their exact values in this framework, as discussed in Remark 1. Similar to the relative-degree-one case, we make the following assumption:

Assumption 3. *$\frac{\partial^m b(\boldsymbol{x}, \boldsymbol{y}_i)}{\partial \bar{\boldsymbol{x}}^m} f_a^{[m-1]} g_a^{[1]}(\bar{\boldsymbol{x}}) \neq \boldsymbol{0}, \forall \boldsymbol{x} \in X, \forall \boldsymbol{y}_i \in X$ at the boundary of $C_1 \cap \dots \cap C_m$.*

In order to find a condition that guarantees the satisfaction of (44) in $[t_k, t_{k+1}), k = 1, 2, \dots,$ we let $\boldsymbol{e}_x$ and $\boldsymbol{e}_x^{(j)}, j \in \{1, \dots, m\}$ be bounded by $\boldsymbol{w} \in \mathbb{R}^n_{>0}$ and $\boldsymbol{\nu}_j \in \mathbb{R}^n_{>0}$, and let $\boldsymbol{e}_i$ and $\boldsymbol{e}_i^{(j)}, j \in \{1, \dots, m\}$ be bounded by $\boldsymbol{W}_i \in \mathbb{R}^n_{>0}$ and $\boldsymbol{V}_{i,j} \in \mathbb{R}^n_{>0}$ i.e., we have

$$|\boldsymbol{e}_x| \leq \boldsymbol{w}, \qquad |\boldsymbol{e}_x^{(j)}| \leq \boldsymbol{\nu}_j, j \in \{1, \dots, m\}, \tag{45}$$

$$|\boldsymbol{e}_i| \leq \boldsymbol{W}_i, \qquad |\boldsymbol{e}_i^{(j)}| \leq \boldsymbol{V}_{i,j}, j \in \{1, \dots, m\}, \tag{46}$$

where the inequalities are interpreted componentwise and the absolute function $|\cdot|$ applies to each component. We also consider the sets of states similar in form to (24) and (25).

Let $\boldsymbol{z} := (\boldsymbol{y}_1, \boldsymbol{e}_x, \boldsymbol{e}_x^{(1)}, \dots, \boldsymbol{e}_x^{(m)}, \boldsymbol{y}_2, \boldsymbol{e}_i, \boldsymbol{e}_i^{(1)}, \dots, \boldsymbol{e}_i^{(m)})$, where $\boldsymbol{y}_1 \in S_x(t_k), \boldsymbol{y}_2 \in S_{y,i}(t_k)$. We further define a set $S_h(t_k)$ in the form

$$\begin{aligned} S_h(t_k) = \{&\boldsymbol{z} : \boldsymbol{y}_1 \in S_x(t_k), |\boldsymbol{e}_x| \leq \boldsymbol{w}, \\ &\wedge_{j=1}^m (|\boldsymbol{e}_x^{(j)}| \leq \boldsymbol{\nu}_j), \boldsymbol{y}_2 \in S_{y,i}(t_k), |\boldsymbol{e}_i| \leq \boldsymbol{W}_i, \\ &\wedge_{j=1}^m (|\boldsymbol{e}_i^{(j)}| \leq \boldsymbol{V}_{i,j}), (\boldsymbol{y}_1 + \boldsymbol{e}_x, \boldsymbol{y}_2 + \boldsymbol{e}_i) \in \cap_{j=1}^m C_j, i \in S_a\}, \end{aligned} \tag{47}$$

where $\wedge_{i=1}^m$ denotes the conjunction from 1 to m.

Then, we can find the minimum values $b_{f_a^m, min}(t_k) \in \mathbb{R}, b_{\alpha_m, min}(t_k) \in \mathbb{R}, b_{e^m, min}(t_k) \in \mathbb{R}, b_{h_{a,i}^m, min}(t_k) \in$

$\mathbb{R}, b_{E_i^m,min}(t_k) \in \mathbb{R}, b_{R,min}(t_k) \in \mathbb{R}$ for the preceding time interval that satisfies $\boldsymbol{z} \in S_h(t_k)$ starting at time t_k by

$$b_{f_a^m,min}(t_k) = \min_{\boldsymbol{z}\in S_h(t_k)} \frac{\partial^m b(\boldsymbol{y}_1 + \boldsymbol{e}_x, \boldsymbol{y}_2 + \boldsymbol{e}_i)}{\partial \boldsymbol{y}_1{}^m} f_a^{[m]}(\boldsymbol{y}_1) \quad (48)$$

$$b_{\alpha_m,min}(t_k) = \min_{\boldsymbol{z}\in S_h(t_k)} \alpha_m(\psi_{m-1}(\boldsymbol{y}_1 + \boldsymbol{e}_x, \boldsymbol{y}_2 + \boldsymbol{e}_i)) \quad (49)$$

$$b_{e^m,min}(t_k) = \min_{\boldsymbol{z}\in S_h(t_k)} \frac{\partial^m b(\boldsymbol{y}_1 + \boldsymbol{e}_x, \boldsymbol{y}_2 + \boldsymbol{e}_i)}{\partial \boldsymbol{e}_x^m} \boldsymbol{e}_x^{(m)} \quad (50)$$

$$b_{h_{a,i}^m,min}(t_k) = \min_{\boldsymbol{z}\in S_h(t_k)} \frac{\partial^m b(\boldsymbol{y}_1 + \boldsymbol{e}_x, \boldsymbol{y}_2 + \boldsymbol{e}_i)}{\partial \boldsymbol{y}_2{}^m} h_{a,i}^{[m]}(\boldsymbol{y}_2) \quad (51)$$

$$b_{E_i^m,min}(t_k) = \min_{\boldsymbol{z}\in S_h(t_k)} \frac{\partial^m b(\boldsymbol{y}_1 + \boldsymbol{e}_x, \boldsymbol{y}_2 + \boldsymbol{e}_i)}{\partial \boldsymbol{e}_i^m} \boldsymbol{e}_i^{(m)} \quad (52)$$

$$b_{R,min}(t_k) = \min_{\boldsymbol{z}\in S_h(t_k)} R(b(\boldsymbol{y}_1 + \boldsymbol{e}_x, \boldsymbol{y}_2 + \boldsymbol{e}_i)) \quad (53)$$

If $\frac{\partial^m b(\boldsymbol{x},\boldsymbol{y}_i)}{\partial \bar{\boldsymbol{x}}^m} f_a^{[m-1]}(\bar{\boldsymbol{x}}) g_a^{[1]}(\bar{\boldsymbol{x}})$ is independent of $\bar{\boldsymbol{x}}$, $\boldsymbol{e}_x$ and $\bar{\boldsymbol{y}}_i$, $\boldsymbol{e}_i$, then we do not need to find its limit value within the set $S_h(t_k)$; otherwise, we proceed as follows. The sign of $u_j(t_k), j \in \{1,\dots,q\}$ can be determined as in the relative-degree-one case. We can then determine its limit value $b_{g_j,lim}(t_k) \in \mathbb{R}, j \in \{1,\dots,q\}$ (defined similar to the relative-degree-one case) by

$$b_{g_j,lim}(t_k) = \begin{cases} \min_{\boldsymbol{z}\in S_h(t_k)} \frac{\partial^m b(\boldsymbol{y}_1+\boldsymbol{e}_x,\boldsymbol{y}_2+\boldsymbol{e}_i)}{\partial \boldsymbol{y}_1{}^m} f_a^{[m-1]}(\boldsymbol{y}_1) g_j^{[1]}(\boldsymbol{y}_1), & \text{if } u_j(t_k) \geq 0, \\ \max_{\boldsymbol{z}\in S_h(t_k)} \frac{\partial^m b(\boldsymbol{y}_1+\boldsymbol{e}_x,\boldsymbol{y}_2+\boldsymbol{e}_i)}{\partial \boldsymbol{y}_1{}^m} f_a^{[m-1]}(\boldsymbol{y}_1) g_j^{[1]}(\boldsymbol{y}_1), & \text{otherwise} \end{cases} \quad (54)$$

Let $b_{g_a,lim}(t_k) = (b_{g_1,lim}(t_k),\dots,b_{g_q,lim}(t_k)) \in \mathbb{R}^{1\times q}$, and we set $b_{g_a,lim}(t_k) = \frac{\partial^m b(\boldsymbol{x},\boldsymbol{y}_i)}{\partial \bar{\boldsymbol{x}}^m} f_a^{[m-1]}(\bar{\boldsymbol{x}}) g_a^{[1]}(\bar{\boldsymbol{x}})$ if $\frac{\partial^m b(\boldsymbol{x},\boldsymbol{y}_i)}{\partial \bar{\boldsymbol{x}}^m} f_a^{[m-1]}(\bar{\boldsymbol{x}}) g_a^{[1]}(\bar{\boldsymbol{x}})$ is independent of $\bar{\boldsymbol{x}}$, $\boldsymbol{e}_x$ and $\bar{\boldsymbol{y}}_i$, $\boldsymbol{e}_i$ for notational simplicity.

The condition that guarantees the satisfaction of (44) in the time interval $[t_k, t_{k+1})$ is then given by

$$b_{f_a^m,min}(t_k) + b_{g_a,lim}(t_k)\boldsymbol{u}(t_k) + b_{e^m,min}(t_k) + b_{h_{a,i}^m,min}(t_k) + b_{E_i^m,min}(t_k) + b_{\alpha_m,min}(t_k) + b_{R,min}(t_k) \geq 0. \quad (55)$$

We also add the sign condition to $S_h(t_k)$ as in the relative degree one case. In order to apply the above condition to problem (19), we just replace (17) by (55) and consider the problem at time t_k, i.e., we have a QP:

$$\min_{\boldsymbol{u}(t_k),\delta(t_k)} ||\boldsymbol{u}(t_k)||^2 + p\delta^2(t_k) \quad (56)$$

subject to (55), (2) and (20). The feasibility of the above QPs can also be guaranteed by finding a proper feasibility constraint [32].

Based on the above, we also have seven events that determine the triggering conditions for solving the QP (56):

- **Event 1:** $|\boldsymbol{e}_x| \leq \boldsymbol{w}$ is about to be violated.
- **Event 2:** $|\boldsymbol{e}_x^{(i)}| \leq \boldsymbol{\nu}_i$ is about to be violated for each $i \in \{1,\dots,m\}$.
- **Event 3:** the state of (10) reaches the boundaries of $S_x(t_k)$.
- **Event 4:** $|\boldsymbol{e}_i| \leq \boldsymbol{W}_i$ is about to be violated.
- **Event 5:** $|\boldsymbol{e}_i^{(j)}| \leq \boldsymbol{V}_{i,j}$ is about to be violated for each $j \in \{1,\dots,m\}$.
- **Event 6:** the state of (11) reaches the boundaries of $S_{y,i}(t_k)$.
- **Event 7:** $\frac{\partial^m b(\boldsymbol{x},\boldsymbol{y}_i)}{\partial \bar{\boldsymbol{x}}^m} f_a^{[m-1]}(\bar{\boldsymbol{x}}) g_j^{[1]}(\bar{\boldsymbol{x}}), j \in \{1,\dots,q\}$ changes sign for $t > t_k$ compared with t_k.

The next time instant $t_{k+1}, k = 1,2\dots$ $(t_1 = 0)$ to solve the QP (56) is determined by:

$$\begin{aligned} t_{k+1} = \min\{t > t_k : |\boldsymbol{e}_x(t)| = \boldsymbol{w} \text{ or } |\boldsymbol{e}_x^{(j)}(t)| = \boldsymbol{\nu}_j, \\ j \in \{1,\dots,m\} \text{ or } |\bar{\boldsymbol{x}}(t) - \bar{\boldsymbol{x}}(t_k)| = \boldsymbol{s}(\beta_1(b(\boldsymbol{x}(t_k), \boldsymbol{y}_i(t_k)))) \\ \text{or } |\bar{\boldsymbol{y}}_i(t) - \bar{\boldsymbol{y}}_i(t_k)| = \boldsymbol{S}_i(\beta_2(b(\boldsymbol{x}(t_k), \boldsymbol{y}_i(t_k)))) \\ \text{or } \frac{\partial^m b(\boldsymbol{x},\boldsymbol{y}_i)}{\partial \bar{\boldsymbol{x}}^m} f_a^{[m-1]}(\bar{\boldsymbol{x}}) g_j^{[1]}(\bar{\boldsymbol{x}}), j \in \{1,\dots,q\} \text{ changes sign} \\ \text{or } |\boldsymbol{e}_i(t)| = \boldsymbol{W}_i \text{ or } |\boldsymbol{e}_i^{(j)}(t)| = \boldsymbol{V}_{i,j}, j \in \{1,\dots,m\}, i \in S_a\}, \end{aligned} \quad (57)$$

Formally, we have the following theorem that shows the satisfaction of the safety constraint (9) for the real unknown dynamics:

Theorem 3. *Given a HOCBF $b(\boldsymbol{x})$ as in Def. 4. Let $t_{k+1}, k = 1,2\dots$ be determined by (57) with $t_1 = 0$, and (55) be determined by (48)-(54), respectively. Then, under Assumptions 1, 3, any control $\boldsymbol{u}(t_k)$ that satisfies (55) and updates the real unknown dynamics and (10) within time interval $[t_k, t_{k+1})$ renders the set $C_1 \cap \dots \cap C_m$ forward invariant for the real unknown dynamics.*

Proof: Similar to the proof of Thm. 2, we have

$$\frac{\partial^m b(\boldsymbol{x}(t), \boldsymbol{y}_i(t))}{\partial t^m} + R(b(\boldsymbol{x}(t), \boldsymbol{y}_i(t))) + \alpha_m(\psi_{m-1}(\boldsymbol{x}(t), \boldsymbol{y}_i(t))) \geq 0, \forall t \in [t_k, t_{k+1}], k = 1,2\dots$$

which is equivalent to the HOCBF constraint (5) in Def. 4. Then, by Thm. 1, (14) and $\boldsymbol{e}_x^{(j)} = \boldsymbol{x}^{(j)} - \bar{\boldsymbol{x}}^{(j)}, \boldsymbol{e}_i^{(j)} = \boldsymbol{y}_i^{(j)} - \bar{\boldsymbol{y}}_i^{(j)}, j \in \{1,\dots,m\}$, we can recursively show that $C_i, i \in \{1,\dots,m\}$ are forward invariant, i.e., the set $C_1 \cap \dots \cap C_m$ is forward invariant for the real unknown dynamics. ■

The process can be summarized through an algorithm similar to Alg. 1 and we can deal with measurement uncertainties as in Remark 1. We also synchronize the state of the real unknown dynamics with the state of the adaptive affine dynamics in (10) (as well as the state of agent $i \in S_a$ and (11)) as in (38) and (39) (as well as (40) and (41)) such that we always have $\boldsymbol{e}_x(t_k) = 0, \boldsymbol{e}_i(t_k) = 0, \boldsymbol{e}_i^{(j)}(t_k) = 0$ and $\boldsymbol{e}_x^{(j)}(t_k), j \in \{1,\dots,m\}$ stay close to 0.

V. State Convergence

In this section, we propose a framework similar to that of solving safety-critical control problems whose goal is to achieve guaranteed state convergence for systems with unknown dynamics. In other words, we define a CLF

$$V(\boldsymbol{x}) = (\boldsymbol{x} - \boldsymbol{K})^T P(\boldsymbol{x} - \boldsymbol{K}), \quad (58)$$

where P is positive definite, for the state convergence requirement in Objective 2, and wish $V(\boldsymbol{x}(t))$ to be decreasing for all $t \geq 0$. For state convergence, we set an infinite horizon, i.e., $T = \infty$ in (8).

Next, we show how to find a CLF constraint that guarantees (58) is decreasing. We first consider an error state similar to (13), and we denote it as $\boldsymbol{e}$ instead of $\boldsymbol{e}_x$ for simplicity. Let

$$\boldsymbol{e} := \boldsymbol{x} - \bar{\boldsymbol{x}}. \quad (59)$$

Note that $\boldsymbol{x}$ and $\bar{\boldsymbol{x}}$ are state vectors from direct measurements of the control system (plant) and from the adaptive affine dynamics (10), respectively. Then

$$V(\boldsymbol{x}) = V(\bar{\boldsymbol{x}} + \boldsymbol{e}). \tag{60}$$

Differentiating $V(\bar{\boldsymbol{x}} + \boldsymbol{e})$, we have

$$\frac{dV(\bar{\boldsymbol{x}}+\boldsymbol{e})}{dt} = \frac{\partial V(\bar{\boldsymbol{x}}+\boldsymbol{e})}{\partial \bar{\boldsymbol{x}}}\dot{\bar{\boldsymbol{x}}} + \frac{\partial V(\bar{\boldsymbol{x}}+\boldsymbol{e})}{\partial \boldsymbol{e}}\dot{\boldsymbol{e}}. \tag{61}$$

where $\dot{\boldsymbol{e}} = \dot{\boldsymbol{x}} - \dot{\bar{\boldsymbol{x}}}$ is evaluated online from $\dot{\boldsymbol{x}}$ (from direct measurements of the actual state derivative) and $\dot{\bar{\boldsymbol{x}}}$ (given through (10)).

The CLF constraint that guarantees state convergence for known dynamics (1) is as in (7), which is done by replacing $\dot{\boldsymbol{x}}$ with (1). However, for unknown dynamics, the CLF constraint is:

$$\frac{\partial V(\boldsymbol{x})}{\partial t} + c_3 V(\boldsymbol{x}) \leq 0,$$

Equivalently, we have

$$\frac{\partial V(\bar{\boldsymbol{x}} + \boldsymbol{e})}{\partial t} + c_3 V(\bar{\boldsymbol{x}} + \boldsymbol{e}) \leq 0, \tag{62}$$

Combining (61), (62) and (10), we get the CLF constraint that guarantees state convergence:

$$\frac{\partial V(\boldsymbol{x})}{\partial \bar{\boldsymbol{x}}} f_a(\bar{\boldsymbol{x}}) + \frac{\partial V(\boldsymbol{x})}{\partial \bar{\boldsymbol{x}}} g_a(\bar{\boldsymbol{x}})\boldsymbol{u} + \frac{\partial V(\boldsymbol{x})}{\partial \boldsymbol{e}}\dot{\boldsymbol{e}} + c_3 V(\boldsymbol{x}) \leq 0, \tag{63}$$

If the CBF (HOCBF) constraint (33) (or (55)) that guarantees (9) does not conflict with the convergence requirement, we can formulate the following problem:

$$\min_{\boldsymbol{u}(t)} \int_0^\infty ||\boldsymbol{u}(t)||^2 dt \tag{64}$$

subject to (33) (or (55)), (2), and the CLF constraint (63).

Following the approach introduced at the end of Sec. II, we solve problem (64) at time $t_k, k = 1, 2 \dots$, and this problem is a QP. However, at time t_k, the QP does not generally know the error state $\boldsymbol{e}(t)$ and its derivative $\dot{\boldsymbol{e}}(t), \forall t > t_k$. Thus, it cannot guarantee that the CLF constraint (63) is satisfied in the time interval $(t_k, t_{k+1}]$, where t_{k+1} is the next time instant to solve the QP. In order to find a condition that guarantees the satisfaction of (63) $\forall t \in (t_k, t_{k+1}]$, we bound the error state and its derivative as in (21).

We now consider the state $\bar{\boldsymbol{x}}$ at time t_k, which satisfies:

$$\bar{\boldsymbol{x}}(t_k) - \boldsymbol{s}_l(V(\boldsymbol{x}(t_k))) \leq \bar{\boldsymbol{x}} \leq \bar{\boldsymbol{x}}(t_k) + \boldsymbol{s}_l(V(\boldsymbol{x}(t_k))), \tag{65}$$

where the inequalities are interpreted componentwise, $\boldsymbol{s}_l : \mathbb{R} \to \mathbb{R}^n_{>0}$. The choice of $\boldsymbol{s}_l(\cdot)$ is similar to the safety-critical control case in the last section, and we also truncate it like (36). We denote the set of states that satisfy (65) at time t_k by

$$S_l(t_k) = \{\boldsymbol{y} \in X : \bar{\boldsymbol{x}}(t_k) - \boldsymbol{s}_l(V(\boldsymbol{x}(t_k))) \leq \boldsymbol{y} \leq \bar{\boldsymbol{x}}(t_k) + \boldsymbol{s}_l(V(\boldsymbol{x}(t_k)))\}. \tag{66}$$

We also consider the set of states $\boldsymbol{x}$ such that $V(\boldsymbol{x}) \leq V(\boldsymbol{x}(t_k))$ as the Lyapunov function will keep decreasing after t_k:

$$S_d(t_k) = \{(\boldsymbol{y}, \boldsymbol{e}) : \boldsymbol{y} \in S_l(t_k), |\boldsymbol{e}| \leq \boldsymbol{w}, V(\boldsymbol{y}+\boldsymbol{e}) \leq V(\boldsymbol{x}(t_k))\} \tag{67}$$

Now, with the above bounds, we are ready to find a condition that guarantees the satisfaction of (63) in the time interval $(t_k, t_{k+1}]$. This is also done by considering the maximum value of each component in (63), as shown next.

We find the maximum values $V_{f_a,max}(t_k) \in \mathbb{R}, V_{c_3,max}(t_k) \in \mathbb{R}, V_{e,max}(t_k) \in \mathbb{R}$ of $\frac{\partial V(\boldsymbol{x})}{\partial \bar{\boldsymbol{x}}} f_a(\bar{\boldsymbol{x}}), c_3 V(\boldsymbol{x}), \frac{\partial V(\boldsymbol{x})}{\partial \boldsymbol{e}}\dot{\boldsymbol{e}}$, respectively, for the preceding time interval that satisfies $(\boldsymbol{y}, \boldsymbol{e}) \in S_d(t_k), |\dot{\boldsymbol{e}}| \leq \boldsymbol{\nu}$ starting at time t_k, i.e., let

$$V_{f_a,max}(t_k) = \max_{(\boldsymbol{y},\boldsymbol{e}) \in S_d(t_k), |\dot{\boldsymbol{e}}| \leq \boldsymbol{\nu}} \frac{\partial V(\boldsymbol{y}+\boldsymbol{e})}{\partial \boldsymbol{y}} f_a(\boldsymbol{y}) \tag{68}$$

$$b_{c_3,max}(t_k) = \max_{(\boldsymbol{y},\boldsymbol{e}) \in S_d(t_k), |\dot{\boldsymbol{e}}| \leq \boldsymbol{\nu}} c_3 V(\boldsymbol{y}+\boldsymbol{e}) \tag{69}$$

$$V_{e,max}(t_k) = \max_{(\boldsymbol{y},\boldsymbol{e}) \in S_d(t_k), |\dot{\boldsymbol{e}}| \leq \boldsymbol{\nu}} \frac{\partial V(\boldsymbol{y}+\boldsymbol{e})}{\partial \boldsymbol{e}} \dot{\boldsymbol{e}} \tag{70}$$

For the remaining term in (63), if $\frac{\partial V(\boldsymbol{x})}{\partial \bar{\boldsymbol{x}}} g_a(\bar{\boldsymbol{x}})$ is independent of $\bar{\boldsymbol{x}}, \boldsymbol{e}$, then we do not need to find its limit value within the bound $(\boldsymbol{y}, \boldsymbol{e}) \in S_d(t_k), |\dot{\boldsymbol{e}}| \leq \boldsymbol{\nu}$; otherwise, we proceed as follows. We can determine the limit value $V_{g_i,lim}(t_k) \in \mathbb{R}, i \in \{1, \dots, q\}$ of $\frac{\partial V(\boldsymbol{x})}{\partial \bar{\boldsymbol{x}}} g_i(\bar{\boldsymbol{x}})$ by

$$V_{g_j,lim}(t_k) = \begin{cases} \max\limits_{(\boldsymbol{y},\boldsymbol{e}) \in S_d(t_k), |\dot{\boldsymbol{e}}| \leq \boldsymbol{\nu}} \frac{\partial V(\boldsymbol{y}+\boldsymbol{e})}{\partial \boldsymbol{y}} g_j(\boldsymbol{y}), & \text{if } u_j(t_k) \geq 0, \\ \min\limits_{(\boldsymbol{y},\boldsymbol{e}) \in S_d(t_k), |\dot{\boldsymbol{e}}| \leq \boldsymbol{\nu}} \frac{\partial V(\boldsymbol{y}+\boldsymbol{e})}{\partial \boldsymbol{y}} g_j(\boldsymbol{y}), & \text{otherwise} \end{cases} \tag{71}$$

Let $V_{g_a,lim}(t_k) = (V_{g_1,lim}(t_k), \dots, V_{g_q,lim}(t_k)) \in \mathbb{R}^q$, and we set $V_{g_a,lim}(t_k) = \frac{\partial V(\boldsymbol{x})}{\partial \bar{\boldsymbol{x}}} g_a(\bar{\boldsymbol{x}})$ if $\frac{\partial V(\boldsymbol{x})}{\partial \bar{\boldsymbol{x}}} g_a(\bar{\boldsymbol{x}})$ is independent of $\bar{\boldsymbol{x}}$ and $\boldsymbol{e}$ for notational simplicity.

The condition that guarantees the satisfaction of (17) in the time interval $(t_k, t_{k+1}]$ is then given by

$$V_{f_a,max}(t_k) + V_{g_a,lim}(t_k)\boldsymbol{u}(t_k) + V_{e,max}(t_k) + V_{c_3,max}(t_k) \leq 0. \tag{72}$$

In order to apply the above condition to problem (64), we just replace (63) by (72) and consider the problem at time t_k, i.e., we have a QP:

$$\min_{\boldsymbol{u}(t_k)} ||\boldsymbol{u}(t_k)||^2 \tag{73}$$

subject to (33) (or (55)), (2), and the CLF condition (72).

Based on the above, we define one additional event (in addition to the seven events from the safety-critical control case in the last section) that determine the triggering conditions for solving the QP (73):

- **Event a:** the state of (10) reaches the boundaries of $S_l(t_k)$.
- **Event b:** the coefficients of the controls in the CLF constraint change sign, similar to Event 7 in the safety guarantee case.

In other words, the next time instant $t_{k+1}, k = 1, 2 \dots$ to solve the QP (73) is determined by the minimum value between the t_{k+1} from (35) (or (57)) and the t_a defined below:

$$t_a = \min\{t > t_k : |\bar{\boldsymbol{x}}(t) - \bar{\boldsymbol{x}}(t_k)| = \boldsymbol{s}_l(V(\boldsymbol{x}(t_k))), \text{ or the triggering time } t \text{ of Event b}\}, \tag{74}$$

where $t_1 = 0$.

Formally, we have the following theorem to show that the convergence for the CLF (58) is guaranteed for the real unknown dynamics with the condition (72):

Theorem 4. *Given a CLF $V(\boldsymbol{x})$ in (58), let $\tau_{k+1}, k = 1, 2 \ldots$ be determined by the minimum value between the t_{k+1} from (35) (or (57)) and the t_a, and (72) be determined by (68)-(71), respectively. Then, under Assumption 1, any control $\boldsymbol{u}(t_k)$ that satisfies (72) and updates the real unknown dynamics and the adaptive dynamics (10) within time interval $[t_k, \tau_{k+1})$ guarantees that the system state converges to the origin.*

Proof: The proof is similar to Thm. 2.

The process can be summarized through an algorithm similar to Alg. 1 and we can deal with measurement uncertainties as in Remark 1. We also synchronize the state of the real unknown dynamics and (10) as in (38) and (39) such that we always have $\boldsymbol{e}(t_k) = 0$ and $\dot{\boldsymbol{e}}(t_k)$ stay close to 0.

If the CLF constraint (72) conflicts with the CBF (HOCBF) constraint (33) (or (55)), one typical way is to relax (72) by replacing 0 with δ, and include δ in the cost function as in (19). Another way to address this is to define a proper CLF (58), e.g., by properly defining the matrix P. This possibility is the subject of ongoing work.

VI. Applications

In this section, we consider three classes of problems where the proposed approach is applied: a robot control problem, an Adaptive Cruise Control (ACC) problem and a traffic merging problem. All the computations and simulations were conducted in MATLAB. We used quadprog to solve the quadratic programs and ode45 to integrate the dynamics.

A. State convergence for a robot control problem

In this case, to simplify the problem, we assume unicycle dynamics for a robot which are known and are given below. Our goal is to study how the inter-sampling effect could affect the performance of the system by comparing the time-driven and event-driven methods for solving this problem.

$$\dot{x} = v\cos\theta, \quad \dot{y} = v\sin\theta, \quad \dot{\theta} = u_1, \quad \dot{v} = u_2, \tag{75}$$

where $(x, y) \in \mathbb{R}^2$ denotes the coordinates of the robot in the Cartesian frame, v denotes its speed, and θ denotes its heading. The inputs u_1, u_2 denote rotation rate control and acceleration control, respectively. $g(\boldsymbol{x}) = (g_1(\boldsymbol{x}), g_2(\boldsymbol{x}))$.

The problem we consider in this case is to make the robot location be stabilized to the origin while minimizing the control $\int_0^\infty \left(u_1^2(t) + u_2^2(t)\right) dt$, given an initial state $\boldsymbol{x}(0)$.

We only require the robot to go to the origin, i.e., the terminal state requirement applies to the robot position. If we define a CLF $V(\boldsymbol{x}) = x^2 + y^2$, then the relative degree of the CLF is 2, and the CLF method cannot work properly as we only take the Lie derivative once in this CLF. In order to address this problem, we combine the state feedback control and the CLF method.In other words, in order to drive the robot to the origin, the desired states for $v\cos\theta$ and $v\sin\theta$ are $-k_1 x$ and $-k_2 y, k_1 > 0, k_2 > 0$, respectively. We then define a CLF:

$$V(\boldsymbol{x}) = (v\cos\theta + k_1 x)^2 + (v\sin\theta + k_2 y)^2 \tag{76}$$

The above CLF has relative degree 1 with respect to both u_1 and u_2 for (75).

In Def. 5, we choose $c_3 = \epsilon > 0$, and have the following CLF constraint:

$$\begin{aligned} &\underbrace{2(v\cos\theta + x)v\cos\theta + 2(v\sin\theta + y)v\sin\theta}_{L_f V(\boldsymbol{x})} \\ &+ \underbrace{2(-(v\cos\theta + x)v\sin\theta + (v\sin\theta + y)v\cos\theta)}_{L_{g_1} V(\boldsymbol{x})} u_1 \\ &+ \underbrace{2((v\cos\theta + x)\cos\theta + (v\sin\theta + y)\sin\theta)}_{L_{g_2} V(\boldsymbol{x})} u_2 + \epsilon V(\boldsymbol{x}) \le 0 \end{aligned} \tag{77}$$

Since the dynamics are known, the only relevant events are Events a and b in Sec. V. At time $t_k, k = 0, 1, \ldots$, we consider the state of (75) bounded by:

$$\begin{aligned} x(t_k) - \gamma_1 V(\boldsymbol{x}(t_k)) \le x \le x(t_k) + \gamma_1 V(\boldsymbol{x}(t_k)), \\ y(t_k) - \gamma_2 V(\boldsymbol{x}(t_k)) \le y \le y(t_k) + \gamma_2 V(\boldsymbol{x}(t_k)), \\ \theta(t_k) - \gamma_3 V(\boldsymbol{x}(t_k)) \le \theta \le \theta(t_k) + \gamma_3 V(\boldsymbol{x}(t_k)), \\ v(t_k) - \gamma_4 V(\boldsymbol{x}(t_k)) \le v \le v(t_k) + \gamma_4 V(\boldsymbol{x}(t_k)), \end{aligned} \tag{78}$$

where $\gamma_1 V(\boldsymbol{x}(t_k)), \gamma_2 V(\boldsymbol{x}(t_k)), \gamma_3 V(\boldsymbol{x}(t_k)), \gamma_4 V(\boldsymbol{x}(t_k))$ are all truncated by $s_0 > 0$, as shown in (36).

Then, we can find the maximum (limit) values of $L_f V(\boldsymbol{x}), \epsilon V(\boldsymbol{x})$ $(L_g V(\boldsymbol{x}))$, respectively, within the bounds defined by (78) and $V(\boldsymbol{x}) \le V(\boldsymbol{x}(t_k))$, and get the CLF constraint as in (72). Note that each process of finding the maximum (limit) value is a NLP as all the costs are nonlinear functions, although the constraints in (78) are linear. We use fmincon in Matlab to solve these NLPs. The sign of $\boldsymbol{u}(t), t \in [t_k, t_{k+1}]$ is determined by the solution of the QP $u_1^2(t) + u_2^2(t)$, s.t., (77) at time $t = t_k$.

The simulation parameters are $\boldsymbol{x}(0) = (-10, -10, -\pi, 2), \epsilon = 0.2, \gamma_1 = \gamma_2 = \gamma_4 = 1, \gamma_3 = 0.5, s_0 = 0.1, k_1 = 1, k_2 = 1$. We first present results of the time-driven method with different discretization time steps Δt, as shown in Fig. 2. We can see that the robot becomes unstable when it gets close to the origin under all three possible Δt values. In order to address this problem, we can further decrease Δt to $0.02s$, as shown in Fig. 3. The stability is improved at the cost of additional computation effort.

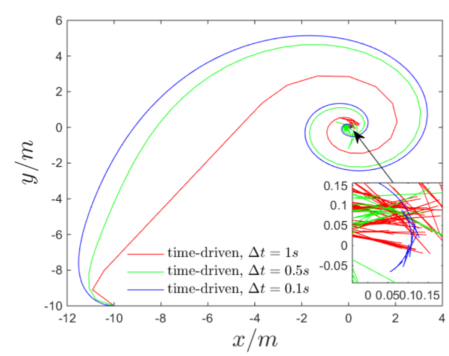


Fig. 2. Trajectories of the robot with the time-driven method under three different discretization time Δt.

We further compare the proposed event-driven method and the time-driven method, as shown in Fig. 3. The trajectory of the time-driven method is spiral, and the system converges to the origin much slower than the event-driven method. This is because the event-driven method always searches for the largest gradient of the CLF within the bound defined by (78), while the time-driven method only evaluates the gradient of the CLF at time t_k. If we choose $\gamma_1 V(\boldsymbol{x}(t_k))$, $\gamma_2 V(\boldsymbol{x}(t_k))$, $\gamma_3 V(\boldsymbol{x}(t_k))$, $\gamma_4 V(\boldsymbol{x}(t_k))$ in (78) to be some constant values, the trajectory of the event-driven method is also spiral, and the robot may also be unstable when it gets close to the origin.

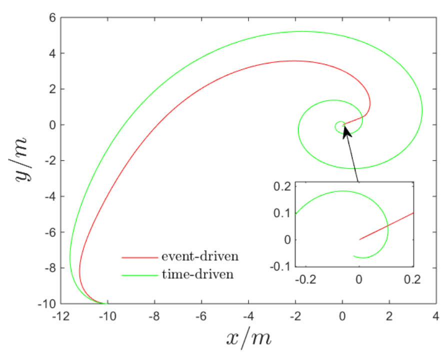


Fig. 3. Trajectories of the robot with both the time-driven (under the discretization time $\Delta t = 0.02$) and event-driven methods.

B. Safety guarantee for an ACC problem

In ACC, we assume that the preceding vehicle has a constant speed $v_p > 0$, and thus, only Events 1-3 in Sec. IV are involved. The real dynamics are **unknown** to the controller:

$$\begin{bmatrix} \dot{v}(t) \\ \dot{z}(t) \end{bmatrix} = \begin{bmatrix} \sigma_1(t) + \frac{\sigma_3(t)}{M}u(t) - \frac{1}{M}F_r(v(t)) \\ \sigma_2(t) + v_p - v(t) \end{bmatrix} \tag{79}$$

where $\boldsymbol{x} = (v, z)$ and $z(t)$ denotes the distance between the preceding and the ego vehicle, $v_p > 0, v(t)$ denote the velocities of the preceding and ego vehicles along the lane, respectively, and $u(t)$ is the control of the ego vehicle. $\{\sigma_1(t)\}, \{\sigma_2(t)\}, \{\sigma_3(t)\}$ denote three random processes whose pdf's have finite support. M denotes the mass of the ego vehicle and $F_r(v(t))$ denotes the resistance force, which is expressed [27] as: $F_r(v(t)) = f_0 sgn(v(t)) + f_1 v(t) + f_2 v^2(t)$, where $f_0 > 0$, $f_1 > 0$ and $f_2 > 0$ are unknown.

The adaptive affine dynamics will be automatically updated as shown in (39), and are in the form:

$$\underbrace{\begin{bmatrix} \dot{\bar{v}}(t) \\ \dot{\bar{z}}(t) \end{bmatrix}}_{\dot{\bar{\boldsymbol{x}}}(t)} = \underbrace{\begin{bmatrix} h_1(t) - \frac{1}{M}F_n(\bar{v}(t)) \\ h_2(t) + v_p - \bar{v}(t) \end{bmatrix}}_{f_a(\bar{\boldsymbol{x}}(t))} + \underbrace{\begin{bmatrix} \frac{1}{M} \\ 0 \end{bmatrix}}_{g_a(\bar{\boldsymbol{x}}(t))} u(t) \tag{80}$$

where $h_1(t) \in \mathbb{R}, h_2(t) \in \mathbb{R}$ denote the two adaptive terms in (39) (also see (84) below), $h_1(0) = 0, h_2(0) = 0$. $\bar{z}(t)$ denotes the distance between the preceding and the ego vehicle for the above adaptive dynamics, and $\bar{v}(t)$ denotes the velocity. $F_n(\bar{v}(t))$ denotes the resistance force, which is *different* from F_r in (79) and is expressed as: $F_n(\bar{v}(t)) = g_0 sgn(\bar{v}(t)) + g_1\bar{v}(t) + g_2\bar{v}^2(t)$, where $g_0 > 0$, $g_1 > 0$ and $g_2 > 0$ are empirically determined.

The control bound is defined as: $-c_d Mg \leq u(t) \leq c_a Mg, \quad \forall t \geq 0$, where $c_a > 0$ and $c_d > 0$ are the maximum acceleration and deceleration coefficients, respectively.

We require that the distance $z(t)$ between the ego vehicle (real dynamics) and its immediately preceding vehicle be always greater than $l_p > 0$, i.e.,

$$z(t) \geq l_p, \quad \forall t \geq 0. \tag{81}$$

The objective is to minimize the cost $\int_0^T \left(\frac{u(t) - F_r(v(t))}{M}\right)^2 dt$. The ego vehicle is also trying to achieve a desired speed $v_d > 0$, which is implemented by a CLF $V(\bar{\boldsymbol{x}}) = (\bar{v} - v_d)^2$ with $c_3 = \epsilon$ as in Def. 5. Since the relative degree of the rear-end safety constraint (81) is two, we define a HOCBF $b(\boldsymbol{x}) = z - l_p$ with $\alpha_1(b(\boldsymbol{x})) = b(\boldsymbol{x})$ and $\alpha_2(\psi_1(\boldsymbol{x})) = \psi_1(\boldsymbol{x})$ as in Def. 4 to implement the safety constraint. Then, the HOCBF constraint (5) which in this case is (with respect to the real dynamics (79)): $\ddot{b}(\boldsymbol{x}) + 2\dot{b}(\boldsymbol{x}) + b(\boldsymbol{x}) \geq 0$. Combining (13), (80) and this equation, we have a HOCBF constraint in the form:

$$\begin{aligned} &\underbrace{-h_1(t) + \frac{F_n(\bar{v}(t))}{M}}_{\frac{\partial^2 b(\boldsymbol{x})}{\partial \bar{\boldsymbol{x}}^2} f_a(\bar{\boldsymbol{x}})} + \underbrace{\frac{-1}{M}}_{\frac{\partial^2 b(\boldsymbol{x})}{\partial \bar{\boldsymbol{x}}^2} g_a(\bar{\boldsymbol{x}})} u(t) + \underbrace{\ddot{e}_2(t)}_{\frac{\partial^2 b(\boldsymbol{x})}{\partial \boldsymbol{e}^2}\ddot{\boldsymbol{e}}} \\ &+ \underbrace{2(h_2(t) + v_p - \bar{v}(t) + \dot{e}_2(t)) + \bar{z}(t) + e_2(t) - l_p}_{R(b(\boldsymbol{x})) + \alpha_2(\psi_1(\boldsymbol{x}))} \geq 0 \end{aligned} \tag{82}$$

where $\boldsymbol{e} = (e_1, e_2), e_1 = v - \bar{v}, e_2 = z - \bar{z}$.

Constant $\boldsymbol{s}(\cdot)$ function in (24): We first consider the case that the $\boldsymbol{s}(\cdot)$ function in (24) is a constant vector. As in (23) and (45), we consider the state and bound the errors at step $t_k, k = 1, 2 \ldots$ for the above HOCBF constraint in the form:

$$\begin{aligned} \bar{v}(t_k) - s_1 \leq \bar{v} \leq \bar{v}(t_k) + s_1, \ \bar{z}(t_k) - s_2 \leq \bar{z} \leq \bar{z}(t_k) + s_2, \\ |e_2| \leq w_2, \quad |\dot{e}_2| \leq \nu_{2,1}, \quad |\ddot{e}_2| \leq \nu_{2,2} \end{aligned} \tag{83}$$

where $s_1 > 0, s_2 > 0, w_2 > 0, \nu_{2,1} > 0, \nu_{2,2} > 0$.

Motivated by (24) and (39), we also synchronize the state and update the adaptive dynamics (80) at step $t_k, k = 1, 2 \ldots$ in the form:

$$\begin{aligned} &\bar{v}(t_k) = v(t_k), \quad \bar{z}(t_k) = z(t_k), \\ &h_1(t^+) = h_1(t^-) - \sum_{i=0}^{k} \ddot{e}_2(t_i), \quad h_2(t^+) = h_2(t^-) + \sum_{i=0}^{k} \dot{e}_2(t_i), \end{aligned} \tag{84}$$

where $\dot{e}_2(t_k) = \dot{z}(t_k) - (h_2 + v_p - \bar{v}(t_k)), \ddot{e}_2(t_k) = \ddot{z}(t_k) - \frac{F_n(\bar{v}(t_k)) - u(t_k^-)}{M} + h_1(t_k), u(t_k^-) = u(t_{k-1})$ and $u(t_0) = 0$. $\dot{z}(t_k), \ddot{z}(t_k)$ are estimated by a sensor that measures the ego vehicle real dynamics (79) at time t_k.

Then, we can find the limit values as in (48)-(54), solve the QP (56) at each time step $t_k, k = 1, 2 \ldots$, and evaluate the next time step t_{k+1} by (57) afterwards. In the evaluation of t_{k+1}, we have $e_2 = z - \bar{z}, \dot{e}_2 = \dot{z} - (h_2 + v_p - \bar{v}), \ddot{e}_2 = \ddot{z} - \frac{F_n(\bar{v}) - u(t_k)}{M} + h_1$, where $z, \dot{z}, \ddot{z}$ are estimated by a sensor that measures the ego vehicle real dynamics (79), and $u(t_k)$ is already obtained by solving the QP (56) and is held constant until we find t_{k+1}. The optimization of (48) is also a QP, while the optimizations of (49)-(54) are LPs. Therefore, they can all be efficiently solved. Each QP or LP can be solved with a computational time $< 0.01s$ in MATLAB (Intel(R) Core(TM) i7-8700 CPU @ 3.2GHz×2).

In the simulation, the initial states of the real dynamics (79) and the adaptive dynamics (80) are $\boldsymbol{x}(0) = \bar{\boldsymbol{x}}(0) = (20m/s, 100m)$. The final time is $T = 30s$. Other simulation parameters are $v_p = 13.89m/s, v_d = 24m/s, M = 1650kg, g = 9.81m/s^2, f_0 = 0.1N, f_1 = 5Ns/m, f_2 = 0.25Ns^2/m, g_0 = 0.3N, g_1 = 10Ns/m, g_2 = 0.5Ns^2/m, s_1 = 0.4m/s, s_2 = 0.5m, w_2 = 1m, \nu_{2,1} = 0.5m/s, \nu_{2,2} = 0.2m/s^2, c_a = 0.6, c_d = 0.6, p = 1, \epsilon = 10$.

The pdf's of $\sigma_1(t), \sigma_2(t), \sigma_3(t)$ are uniform over the intervals $[-0.2, 0.2]m/s^2, [-2, 2]m/s, [0.9, 1.1]$, respectively. The sensor sampling rate is 20Hz. We compare the proposed event-driven framework with the time driven approach. The discretization time step for the time-driven approach is $\Delta t = 0.1$.

The results are shown in Figs. 4 and 5. Note that in the event-driven approach (blue lines), the control exhibits large variations in order to be responsive to the random processes in the real dynamics. The control is constant in each time interval. If we decrease the uncertainty levels by a factor of 10, the control is smoother (magenta lines). Thus, highly accurately modelled adaptive dynamics can smooth the control.

It follows from Fig. 5 that the set $C_1 \cap C_2$ is forward invariant for the real vehicle dynamics (79), i.e., the safety constraint (81) is guaranteed with the proposed event-driven approach. However, the safety is not guaranteed even with state synchronization under the time-driven approach.

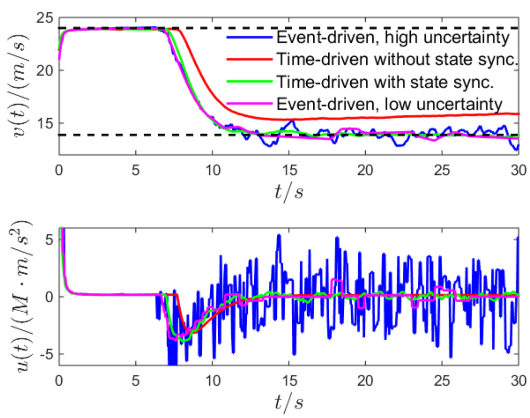


Fig. 4. Speed and control profiles for the proposed event-driven framework and time-driven with or without state synchronization.

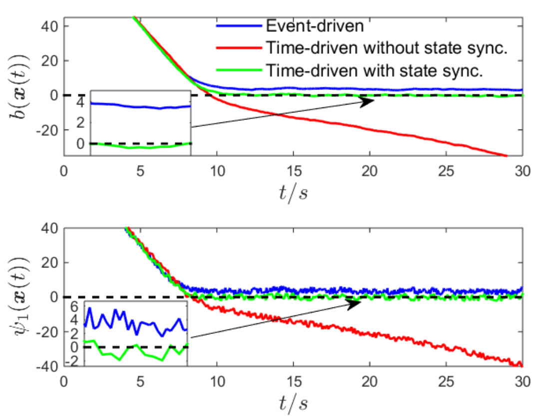


Fig. 5. The variations of functions $b(\boldsymbol{x}(t))$ and $\psi_1(\boldsymbol{x}(t))$ for the proposed event-driven framework and time-driven with or without state synchronization. $b(\boldsymbol{x}(t)) \geq 0$ and $\psi_1(\boldsymbol{x}(t)) \geq 0$ imply the forward invariance of $C_1 \cap C_2$. The set $C_1 \cap C_2$ is forward invariant for the real dynamics (79) with the proposed event-driven framework. Both the set C_1 and C_2 are not forward invariant under time-driven approach with or without state synchronization.

In the event-driven approach, the number of QPs (events) within $[0, T]$ is reduced by about 50% compared with the time-driven approach. If we multiply the bounds of the random processes $\sigma_1(t), \sigma_2(t)$ by 2, then the number of events increases by about 23% for both the 20Hz and 100Hz sensor sampling rate, which shows that accurate adaptive dynamics can reduce the number of events, and thus improves the computational efficient. We also present the event times and state errors between the real dynamics and the nominal dynamics in the case when uncertainty levels decrease by a factor of 10 in Fig. 6. The events are less frequently triggered after the ego vehicle approaches its preceding vehicle with similar speed. The state errors significantly increase when events are not triggered for a long time, but reset to 0 when a new event is triggered.

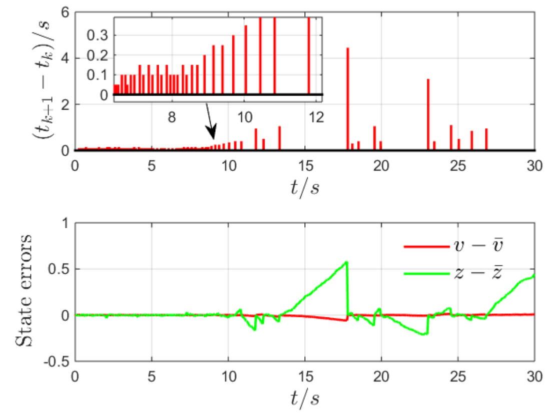


Fig. 6. Event times and state errors for event triggered control between the real dynamics and the adaptive affine control dynamics. The state errors increase when no events are triggered, and reset to 0 when a new event is triggered.

Comparison between constant $\boldsymbol{s}(\cdot)$ function and time-varying $\boldsymbol{s}(\cdot)$ in (24): Next, we consider the comparison between a constant $\boldsymbol{s}(\cdot)$ function (as in the above case) and a time-varying $\boldsymbol{s}(\cdot)$ in (24). In other words, in the time-varying case, we replace the state bounds in (83) by:

$$\begin{aligned} \bar{v}(t_k) - \gamma_1 b(\boldsymbol{x}(t_k)) \leq \bar{v} \leq \bar{v}(t_k) + \gamma_1 b(\boldsymbol{x}(t_k)), \\ \bar{z}(t_k) - \gamma_2 b(\boldsymbol{x}(t_k)) \leq \bar{z} \leq \bar{z}(t_k) + \gamma_2 b(\boldsymbol{x}(t_k)) \end{aligned} \tag{85}$$

where $\gamma_1 > 0, \gamma_2 > 0$. $\gamma_1 b(\boldsymbol{x}(t_k)), \gamma_2 b(\boldsymbol{x}(t_k))$ are truncated by 0.4, 0.5, respectively, as shown in (36).All other settings are the same as before. The pdf's of $\sigma_1(t), \sigma_2(t), \sigma_3(t)$ are uniform over the intervals $[-0.02, 0.02] m/s^2, [-0.2, 0.2] m/s, [0.99, 1.01]$, respectively.

We consider different γ_1, γ_2 values and show how they may affect the conservativeness of the vehicle in Fig. 7. The time-varying $\boldsymbol{s}(\cdot)$ in (24) makes the system less conservative compared with the constant case. This is because $\boldsymbol{s}(\cdot)$ depends on the value of the barrier function $b(\boldsymbol{x})$ in the time-varying case, and the magnitude of $\boldsymbol{s}(\cdot)$ is small when the system state gets close to the safety set boundary. Meanwhile, when the system state gets close to the safety set boundary, the system state variation is small, and smaller $\boldsymbol{s}(\cdot)$ can make the system state stay closer to the set boundary. It follows from Fig. 7 that smaller γ_1, γ_2 can force the system state to stay closer to the set boundary.

C. *Traffic merging (decentralized multi-agent control)*

The merging problem arises when traffic must be joined from two roads, usually associated with a main lane and a merging lane as shown in Fig. 8. We consider the case where all traffic consists of vehicles randomly arriving at the two lanes joined at the Merging Point (MP) M. The segment from the origin O or O' to the merging point M has a length L

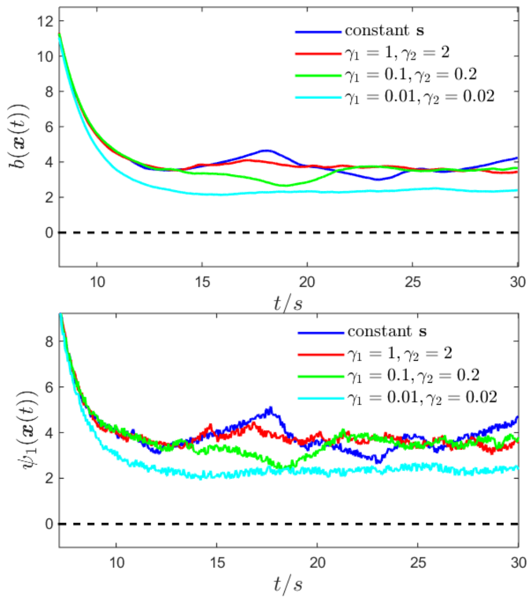


Fig. 7. The variations of functions $b(\boldsymbol{x}(t))$ and $\psi_1(\boldsymbol{x}(t))$ for the proposed event-driven framework with constant and time-varying $\boldsymbol{s}(\cdot)$ in (24). $b(\boldsymbol{x}(t)) \geq 0$ and $\psi_1(\boldsymbol{x}(t)) \geq 0$ imply the forward invariance of $C_1 \cap C_2$.

for both lanes, and is called the control zone. In order to share or obtain other vehicle information, each vehicle can use its onboard sensors or communicate with a coordinator associated with the MP whose main function is to collect and share vehicle states. A more detailed merging problem setup is given in [34]. In contrast to the problem considered in [34] where the vehicle dynamics are assumed known, in real traffic merging each vehicle does not know the dynamics of other vehicles and may also not have accurate dynamics of its own. Therefore, the safe merging constraint becomes critical and hard to be guaranteed.

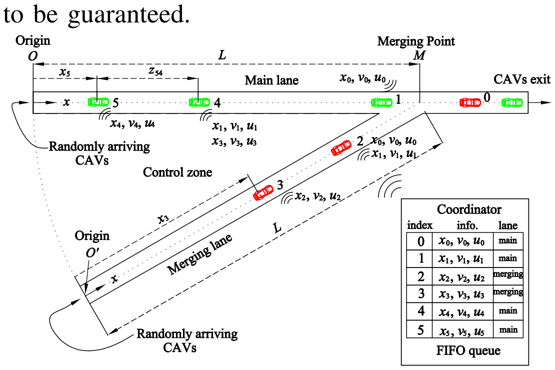


Fig. 8. The merging problem, a collision may happen at the merging point.

The real dynamics for i are **unknown** to the controller:

$$\dot{x}_i(t) = \sigma_{i,1}(t) + v_i(t),\ \dot{v}_i(t) = \sigma_{i,2}(t) + \sigma_{i,3}(t)u_i(t), \quad (86)$$

where $\boldsymbol{x}_i = (x_i, v_i)$ and $x_i(t)$ denotes the along lane distance of vehicle i with respect to the origin, $v_i(t)$ denotes its the velocity, and $u(t)$ is its control (acceleration). $\sigma_1(t), \sigma_2(t), \sigma_3(t)$ denote three random processes whose pdf's have finite support.

As in [34], we consider the following double integrator as the initial adaptive model:

$$\dot{\bar{x}}_i(t) = h_{i,1}(t) + \bar{v}_i(t), \quad \dot{\bar{v}}_i(t) = h_{i,2}(t) + u_i(t), \quad (87)$$

where $\bar{\boldsymbol{x}}_i = (\bar{x}_i, \bar{v}_i)$. $h_{i,1}(t) \in \mathbb{R}, h_{i,2}(t) \in \mathbb{R}$ denote the two adaptive terms in (39), $h_{i,1}(0) = 0, h_{i,2}(0) = 0$.

The objective is to jointly minimize the travel time and energy consumption for each vehicle i in the form $\int_{t_i^0}^{t_i^m} \left(\beta + u_i^2(t)\right) dt$, where t_i^0, t_i^m denote the arrival time of vehicle i at the origin and at the merging point M, respectively. $\beta > 0$ is a weight parameter that captures the time-energy consumption trade-off. The rear-end safety constraint between vehicle i and its preceding vehicle i_p is similar to the ACC example. For simplicity, we only consider the safe merging constraints for two CAVs $i, i-1$ coming from different roads:

$$x_{i-1}(t_i^m) - x_i(t_i^m) \geq \varphi v_i(t_i^m) + l_0 \quad (88)$$

where $\varphi > 0$ is the headway time and $l_0 \geq 0$. For simplicity, we consider high speed traffic merging, and take $l_0 = 0$. In order to use the CBF method to implement the above safe merging constraint, we convert it to a continously differentiable constraint:

$$x_{i-1}(t) - x_i(t) \geq \varphi \frac{x_i(t)}{L} v_i(t), \forall t \in [t_i^0, t_i^m], \quad (89)$$

where $x_{i-1}(t) - x_i(t) \geq 0$ when $x_i(t) = 0$, which means the two vehicles $i-1, i$ are allowed arrive at the same time at the origins O and O', respectively. Moreover, note that $x_{i-1}(t) - x_i(t) \geq \varphi v_i(t)$ when $x_i(t) = L$, which satisfies the safe merging constraint (88) when i arrives at the merging point M ($i-1$ has already passed the merging point).

We take i as the ego vehicle in the merging problem. We assume vehicle $i-1$ is under unconstrained optimal control [34], and vehicle i takes the unconstrained optimal control as a reference. Vehicle i does not know its own dynamics, as well as those of $i-1$. In order to implement the continous version of the safe merging constraint (89), we define a CBF $b(\boldsymbol{x}_i, \boldsymbol{x}_{i-1}) = x_{i-1}(t) - x_i(t) - \varphi \frac{x_i(t)}{L} v_i(t)$. The relative degree of this CBF is only one with respect to (86). We choose $\alpha_1(b(\boldsymbol{x}_i, \boldsymbol{x}_{i-1})) = b(\boldsymbol{x}_i, \boldsymbol{x}_{i-1})$ in Def. 4. The CBF constraint (5) which in this case is (with respect to the real dynamics (86)): $\dot{b}(\boldsymbol{x}_i, \boldsymbol{x}_{i-1}) + b(\boldsymbol{x}_i, \boldsymbol{x}_{i-1}) \geq 0$. Combining (13), (87) and this constraint, we have

$$\begin{aligned} \bar{v}_{i-1} + h_{i-1,1} + \dot{e}_{x_{i-1}} - \bar{v}_i - h_{i,1} - \dot{e}_{x_i} \\ -\frac{\varphi}{L}(\bar{v}_i + h_{i,1} + \dot{e}_{x_i})(\bar{v}_i + e_{v_i}) - \frac{\varphi}{L}(\bar{x}_i + e_{x_i})(u_i + h_{i,2} + \dot{e}_{v_i}) \\ +\bar{x}_{i-1} + e_{x_{i-1}} - \bar{x}_i - e_{x_i} - \frac{\varphi}{L}(\bar{x}_i + e_{x_i})(\bar{v}_i + e_{v_i}) \geq 0, \end{aligned} \quad (90)$$

where $e_{x_{i-1}} = x_{i-1} - \bar{x}_{i-1}, e_{x_i} = x_i - \bar{x}_i, e_{v_i} = v_i - \bar{v}_i$.

The vehicles may arrive at the origins at the same time, i.e., $b(\boldsymbol{x}_i, \boldsymbol{x}_{i-1})$ may be initially close to 0. Thus, we take $\boldsymbol{s}(\cdot), \boldsymbol{S}_i(\cdot)$ in (23) to be some constant vectors. Similar to (23) and (45), (46), we consider the state and bound the errors at step $t_k, k = 1, 2 \dots$ for the above CBF constraint in the form:

$$\begin{aligned} &\bar{x}_{i-1}(t_k) - S_1 \leq \bar{x}_{i-1} \leq \bar{x}_{i-1}(t_k) + S_1, \\ &\bar{v}_{i-1}(t_k) - S_2 \leq \bar{v}_{i-1} \leq \bar{v}_{i-1}(t_k) + S_2, \\ &\bar{x}_i(t_k) - s_1 \leq \bar{x}_i \leq \bar{x}_i(t_k) + s_1, \bar{v}_i(t_k) - s_2 \leq \bar{v}_i \leq \bar{v}_i(t_k) + s_2, \\ &|e_{x_{i-1}}| \leq W, \quad |\dot{e}_{x_{i-1}}| \leq V, \\ &|e_{x_i}| \leq w_1, \quad |\dot{e}_{x_i}| \leq \nu_1, \quad |e_{v_i}| \leq w_2, \quad |\dot{e}_{v_i}| \leq \nu_2 \end{aligned} \quad (91)$$

where $S_1 > 0, S_2 > 0, s_1 > 0, s_2 > 0, W > 0, w_1 > 0, w_2 > 0, V > 0, \nu_1 > 0, \nu_2 > 0$.

Motivated by (38)-(41), we synchronize the state and update the adaptive dynamics (87) at step $t_k, k = 1, 2 \dots$ in the form:

$$\bar{x}_{i-1}(t_k) = x_{i-1}(t_k),\ \bar{v}_{i-1}(t_k) = v_{i-1}(t_k),\ \bar{x}_i(t_k) = x_i(t_k),$$
$$\bar{v}_i(t_k) = v_i(t_k), \quad h_{i-1,1}(t^+) = h_{i-1,1}(t^-) + \sum_{i=0}^{k} \dot{e}_{x_{i-1}}(t_i),$$
$$h_{i,1}(t^+) = h_{i,1}(t^-) + \sum_{i=0}^{k} \dot{e}_{x_i}(t_i), h_{i,2}(t^+) = h_{i,2}(t^-) + \sum_{i=0}^{k} \dot{e}_{v_i}(t_i), \tag{92}$$

where $\dot{e}_{x_{i-1}}(t_k) = \dot{x}_{i-1}(t_k) - (\bar{v}_{i-1}(t_k) + h_{i-1,1}(t_k))$, $\dot{e}_{x_i}(t_k) = \dot{x}_i(t_k) - (\bar{v}_i(t_k) + h_{i,1}(t_k))$, $\dot{e}_{v_i}(t_k) = \dot{v}_i(t_k) - (u_i(t_k^-) + h_{i,2}(t_k))$, $u(t_k^-) = u(t_{k-1})$ and $u(t_0) = 0$. $\dot{x}_{i-1}(t_k), \dot{x}_i(t_k), \dot{v}_i(t_k)$ are estimated by a sensor that measures the real dynamics (86) of $i-1, i$ at time t_k.

Then, we can find the limit values as in (27)-(32), solve the QP (34) at each time step $t_k, k = 1, 2 \dots$, and evaluate the next time step t_{k+1} by (35) afterwards. In the evaluation of t_{k+1}, we have $e_{x_{i-1}} = x_{i-1} - \bar{x}_{i-1}, e_{x_i} = x_i - \bar{x}_i, e_{v_i} = v_i - \bar{v}_i, \dot{e}_{x_{i-1}} = \dot{x}_{i-1} - (\bar{v}_{i-1} + h_{i-1,1}), \dot{e}_{x_i} = \dot{x}_i - (\bar{v}_i + h_{i,1}), \dot{e}_{v_i} = \dot{v}_i - (u_i + h_{i,2})$, where $x_{i-1}, x_i, v_i, \dot{x}_{i-1}, \dot{x}_i, \dot{v}_i$ are estimated by a sensor that measures the real dynamics of $i-1, i$, and $u(t_k)$ is already obtained by solving the QP (34) and is held as a constant until we find t_{k+1}. The optimizations (27)-(32) are NLPs due to the nonlinearity of the CBF $b(\boldsymbol{x}_i, \boldsymbol{x}_{i-1})$. Each NLP can be solved with a computational time of about $0.03s$ using fmincon in MATLAB, and each QP can be solved within $0.01s$ using quadprog in MATLAB (Intel(R) Core(TM) i7-8700 CPU @ 3.2GHz×2).

In the simulation, the initial speeds of vehicles $i-1, i$ are $18m/s, 20m/s$ with arrival times $0s, 1s$ at the origin O or O', respectively. Other simulation parameters are $\beta = 2.666, \varphi = 1.8s, L = 400m, S_1 = 0.5m, S_2 = 0.2m/s, s_1 = 0.5m, s_2 = 0.2m/s, W = 0.6m, V = 0.3m/s, w_1 = 0.6m, w_2 = 0.3m/s, \nu_1 = 0.3m/s, \nu_2 = 0.2m/s^2$.

The pdf's of $\sigma_1(t), \sigma_2(t), \sigma_3(t)$ are uniform over the intervals $[-2, 2]m/s, [-0.2, 0.2]m/s^2, [0.9, 1.1]$, respectively. The sensor sampling rate is 100Hz. We compare the proposed event-driven framework with the time-driven approach ($\Delta t = 0.02s$) that takes double integrator as vehicle dynamics.

The simulation results are shown in Fig. 9. Note that, in order to improve the computation efficiency while staying close to optimal solutions, we employed the joint optimal control and barrier function method. As expected, the safe merging constraint between i and $i-1$ is not satisfied with the time-driven method (blue curves shown in Fig. 9) due to the unknown dynamics of both i and $i-1$. The safe merging constraint for i and $i-1$ is guaranteed when using the event-driven approach, as the red curves shown in Fig. 9, but vehicle i tends to be conservative when it approaches the merging point. In order to alleviate this conservativeness, we consider a small-bound case in which the state and error bound values are 20% of the default values, as the green curves shown in Fig. 9.

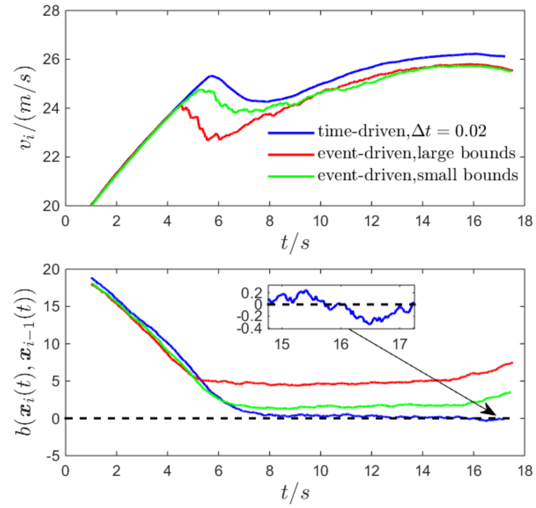


Fig. 9. Comparision between the proposed event-driven method and the time-driven method in guaranteeing the satisfaction of the safe merging constraint for vehicles $i, i-1$. $b(\boldsymbol{x}_i, \boldsymbol{x}_{i-1}) \geq 0$ denotes the forward invariance of C_1, i.e., the satisfaction of the safe merging constraint (under the event driven method). In the small-bounds case, all the state and error bound values are 20% of the ones in the large-bounds case (default values).

VII. Conclusion & Future Work

This paper proposes an event-triggered framework for safety-critical control of systems with unknown dynamics. This framework is based on defining adaptive affine dynamics to estimate the real system state, and an event-triggering mechanism for solving the problem using a condition we determine that guarantees safety or convergence between events. We have demonstrated the effectiveness of the proposed framework by applying it to three separate applications. In the future, we will study the issue of guaranteeing the feasibility of all QPs, as well as investigate an extension of the proposed framework to include receding horizon control.

References

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

[2] E. V. Denardo, *Dynamic Programming: Models and Applications.* Dover Publications, 2003.

[3] J. B. Rawlings, D. Q. Mayne, and M. M. Diehl, *Model Predictive Control: Theory, Computation, and Design.* Nob Hill Publishing, 2018.

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

[5] P. Glotfelter, J. Cortes, and M. Egerstedt, “Nonsmooth barrier functions with applications to multi-robot systems,” *IEEE control systems letters*, vol. 1, no. 2, pp. 310–315, 2017.

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

[7] K. P. Tee, S. S. Ge, and E. H. Tay, “Barrier lyapunov functions for the control of output-constrained nonlinear systems,” *Automatica*, vol. 45, no. 4, pp. 918–927, 2009.

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

[9] S. P. Boyd and L. Vandenberghe, *Convex optimization.* New York: Cambridge university press, 2004.

[10] J. P. Aubin, *Viability theory.* Springer, 2009.

[11] S. Prajna, A. Jadbabaie, and G. J. Pappas, “A framework for worst-case and stochastic safety verification using barrier certificates,” *IEEE Trans. on Automatic Control*, vol. 52, no. 8, pp. 1415–1428, 2007.

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

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

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

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

[16] K. Galloway, K. Sreenath, A. D. Ames, and J. Grizzle, "Torque saturation in bipedal robotic walking through control lyapunov function based quadratic programs," *preprint arXiv:1302.7314*, 2013.

[17] G. Yang, C. Belta, and R. Tron, "Self-triggered control for safety critical systems using control barrier functions," in *Proc. of the American Control Conference*, 2019, pp. 4454–4459.

[18] A. J. Taylor, P. Ong, J. Cortes, and A. D. Ames, "Safety-critical event triggered control via input-to-state safe barrier functions," *IEEE Control Systems Letters*, vol. 5, no. 3, pp. 749–754, 2021.

[19] P. Tabuada, "Event-triggered real-time scheduling of stabilizing control tasks," *IEEE Transactions on Automatic Control*, vol. 52, no. 9, pp. 1680–1685, 2007.

[20] P. Ong and J. Cortes, "Performance-barrier-based event-triggered control with applications to network systems," *preprint arXiv:2108.12702*, 2021.

[21] A. J. Taylor, A. Singletary, Y. Yue, and A. D. Ames, "Learning for safety-critical control with control barrier functions," in *Proc. of Conf. on Learning for Dynamics and Control*, 2020, pp. 708–717.

[22] M. J. Khojasteh, V. Dhiman, M. Franceschetti, and N. Atanasov, "Probabilistic safety constraints for learned high relative degree system dynamics," in *Proc. of Conf. on Learning for Dynamics and Control*, 2020, pp. 781–792.

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

[24] U. Borrmann, L. Wang, A. D. Ames, and M. Egerstedt, "Control barrier certificates for safe swarm behavior," *IFAC-PapersOnLine*, vol. 48, no. 27, pp. 68–73, 2015.

[25] R. Cheng, M. J. Khojasteh, A. D. Ames, and J. W. Burdick, "Safe multi-agent interaction through robust control barrier functions with learned uncertainties," in *Proc. of 59rd IEEE Conference on Decision and Control*, 2020, pp. 777–783.

[26] W. Xiao, C. Belta, and C. G. Cassandras, "Event-triggered safety-critical control for systems with unknown dynamics," in *Proc. of 60th IEEE Conference on Decision and Control*, 2021, pp. 540–545.

[27] H. K. Khalil, *Nonlinear Systems*. Prentice Hall, third edition, 2002.

[28] L. Lindemann and D. V. Dimarogonas, "Control barrier functions for signal temporal logic tasks," *IEEE Control Systems Letters*, vol. 3, no. 1, pp. 96–101, 2019.

[29] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2017.

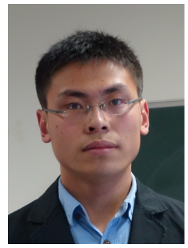
[30] W. Xiao, C. Belta, and C. G. Cassandras, "High order control lyapunov-barrier functions for temporal logic specifications," in *Proc. of the American Control Conference*, 2021, pp. 4886–4891.

[31] X. Tan, W. Shaw Cortez, and D. V. Dimarogonas, "High-order barrier functions: Robustness, safety and performance-critical control," *IEEE Transactions on Automatic Control*, pp. 1–1, 2021.

[32] W. Xiao, C. Belta, and C. G. Cassandras, "Sufficient conditions for feasibility of optimal control problems using control barrier functions," *Automatica*, vol. 135, p. 109960, 2022.

[33] B. J. Morris, M. J. Powell, and A. D. Ames, "Continuity and smoothness properties of nonlinear optimization-based feedback controllers," in *2015 54th IEEE Conference on Decision and Control*, 2015, pp. 151–158.

[34] W. Xiao and C. G. Cassandras, "Decentralized optimal merging control for connected and automated vehicles with safety constraint guarantees," *Automatica*, vol. 123.109333, 2021.



Wei Xiao (S'19) is currently a postdoctoral associate at Massachusetts Institute of Technology. He received a B.Sc. degree from the University of Science and Technology Beijing, China in 2013, a M.Sc. degree from the Chinese Academy of Sciences (Institute of Automation), China in 2016, and a Ph.D. degree from the Boston University, Brookline, MA, USA in 2021. His research interests include control theory and machine learning, with particular emphasis on robotics and traffic control. He received an Outstanding Student Paper Award at the 2020 IEEE Conference on Decision and Control.

Calin Belta (F'17) received B.Sc. and M.Sc, degrees from the Technical University of Iasi, Romania in 1995 and 1997 and M.Sc. and Ph.D. degrees from the University of Pennsylvania, Philadelphia, USA in 2001 and 2003. He is a Professor in the Department of Mechanical Engineering at Boston University, where he holds the Tegan family Distinguished Faculty Fellowship. He is the Director of the BU Robotics Lab and of Center for Autonomous and Robotic Systems (CARS). His research focuses on dynamics and control theory, with particular emphasis on hybrid and cyber-physical systems, formal synthesis and verification, and applications in robotics and systems biology. He received the Air Force Office of Scientific Research Young Investigator Award and the National Science Foundation CAREER Award. He is a fellow of IEEE.

Christos G. Cassandras (F'96) is Distinguished Professor of Engineering at Boston University. He is Head of the Division of Systems Engineering, Professor of Electrical and Computer Engineering, and co-founder of Boston Universitys Center for Information and Systems Engineering (CISE). Hereceived degrees from Yale University, Stanford University, and Harvard University. In 1982-1984 he was with ITP Boston, Inc. where he worked on the design of automated manufacturing systems. In 1984-1996 he was a faculty member at the Department of Electrical and Computer Engineering, University of Massachusetts/Amherst. He specializes in the areas of discrete event and hybrid systems, cooperative control, stochastic optimization, and computer simulation, with applications to computer and sensor networks, manufacturing systems, and transportation systems. He has published over 450 refereed papers in these areas, and six books. He has guest-edited several technical journal issues and currently serves on several journal Editorial Boards, including Editor of Automatica. In addition to his academic activities, he has worked extensively with industrial organizations on various systems integration projects and the development of decision support software. He has most recently collaborated with The MathWorks, Inc. in the development of the discrete event and hybrid system simulator SimEvents.

Dr. Cassandras was Editor-in-Chief of the IEEE Transactions on Automatic Control from 1998 through 2009 and has also served as Editor for Technical Notes and Correspondence and Associate Editor. He was the 2012 President of the IEEE Control Systems Society (CSS). He has also served as Vice President for Publications and on the Board of Governors of the CSS, as well as on several IEEE committees, and has chaired several conferences. He has been a plenary/keynote speaker at numerous international conferences, including the 2017 IFAC World Congress, the American Control Conference in 2001 and the IEEE Conference on Decision and Control in 2002 and 2016, and has also been an IEEE Distinguished Lecturer.

He is the recipient of several awards, including the 2011 IEEE Control Systems Technology Award, the Distinguished Member Award of the IEEE Control Systems Society (2006), the 1999 Harold Chestnut Prize (IFAC Best Control Engineering Textbook) for Discrete Event Systems: Modeling and Performance Analysis, a 2011 prize and a 2014 prize for the IBM/IEEE Smarter Planet Challenge competition, the 2014 Engineering Distinguished Scholar Award at Boston University, several honorary professorships, a 1991 Lilly Fellowship and a 2012 Kern Fellowship. He is a member of Phi Beta Kappa and Tau Beta Pi. He is also a Fellow of the IEEE and a Fellow of the IFAC.