

Practical Considerations for Discrete-Time Implementations of Continuous-Time Control Barrier Function-Based Safety Filters

Lukas Brunke, Siqi Zhou, Mingxuan Che, and Angela P. Schoellig

Abstract—Safety filters based on control barrier functions (CBFs) have become a popular method to guarantee safety for uncertified control policies, e.g., as resulting from reinforcement learning. Here, safety is defined as staying in a pre-defined set, the safe set, that adheres to the system’s state constraints, e.g., as given by lane boundaries for a self-driving vehicle. In this paper, we examine one commonly overlooked problem that arises in practical implementations of continuous-time CBF-based safety filters. In particular, we look at the issues caused by discrete-time implementations of the continuous-time CBF-based safety filter, especially for cases where the magnitude of the Lie derivative of the CBF with respect to the control input is zero or close to zero. When overlooked, this filter can result in undesirable chattering effects or constraint violations. In this work, we propose three mitigation strategies that allow us to use a continuous-time safety filter in a discrete-time implementation with a local relative degree. Using these strategies in augmented CBF-based safety filters, we achieve safety for all states in the safe set by either using an additional penalty term in the safety filtering objective or modifying the CBF such that those undesired states are not encountered during closed-loop operation. We demonstrate the presented issue and validate our three proposed mitigation strategies in simulation and on a real-world quadrotor.

I. INTRODUCTION

Safety filters have recently gained interest with the rise of learning-based control and reinforcement learning approaches. While such learning-based approaches can improve the controller’s performance based on interaction data, they do not provide safety guarantees (e.g., a self-driving vehicle that should stay inside lane boundaries). A safety filter tries to find the minimal modification to a potentially arbitrary control input proposed by an uncertified control policy that still achieves safety [1]. Safety filters typically rely on model knowledge of the system for accurate predictions [2]–[4].

One popular safety filtering method relies on control barrier functions (CBF). Control barrier functions can be used to encode control invariant sets. These sets guarantee that if the system’s state is initialized in this set, there always exists a feasible control input to keep the system inside the set for all future time. If a control invariant set satisfies the state constraints, it is typically called a safe set, and the system is referred to as being safe inside this set [5]. The advantage of CBFs is that determining control invariance amounts to checking a scalar condition. For control-affine systems, this scalar condition can be used as an affine constraint in a quadratic program (QP) to find the closest feasible control

The authors are with the [Learning Systems and Robotics Lab](#) at the Technical University of Munich, Germany and the University of Toronto, Canada. The authors are also affiliated with the Munich Institute of Robotics and Machine Intelligence (MIRMI), the University of Toronto Robotics Institute, and the Vector Institute for Artificial Intelligence. Emails: {lukas.brunke, siqi.zhou, mingxuan.che, angela.schoellig}@tum.de

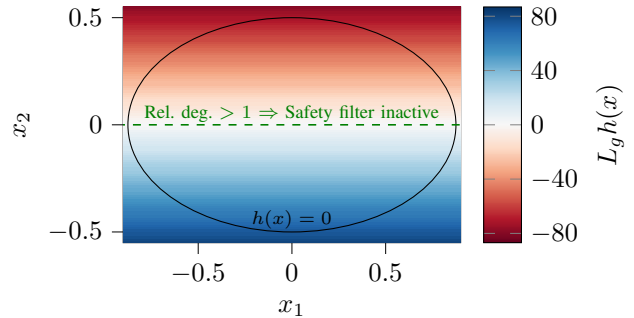


Fig. 1: Visualization of the case study in Sec. VI for a linear system with an ellipsoidal CBF. The states for which $\|L_g h(x)\|$ is small allow the CBF-based safety filter to apply control inputs close to the uncertified control input, rendering the safety filter close to being inactive. Moreover, along the green line, the Lie derivative term $L_g h(x)$ is zero, and the relative degree s does not equal one. In these states, the safety filter is completely inactive. In a discrete-time implementation, the instantaneous inactivity of the safety filter can result in undesirable inputs that cannot be corrected at the subsequent discrete-time step. In both cases, this can lead to chattering and/or safe set violations.

input to a proposed uncertified control input. This QP yields the control barrier function-based safety filter.

CBFs are typically defined for continuous-time systems. To preserve the CBF-based safety filter’s guarantees, the aforementioned QP has to be solved infinitely fast [6]. This is impossible in practice, such that the CBF-based safety filter only approximately provides safety guarantees by solving the QP at discrete timesteps. Nevertheless, applying this approximate strategy in practice has led to many successful real-world implementations, especially when the sampling time is small [7]–[12].

A few works have systematically addressed this discrete-time implementation by analyzing the safety between discrete timesteps or providing an event-triggered formulation [6], [13]. However, these approaches can lead to conservative closed-loop performance or rely on knowing the global relative degree. Checking the global relative degree is not always straightforward, and misspecification can lead to undesired behaviours, especially in discrete-time implementations.

There also exist discrete-time CBFs that have been specifically designed for discrete-time implementations [14]–[16]. However, even for control-affine systems, the resulting safety filter optimization problem is typically a nonlinear program, which reduces the real-time feasibility for real-world systems. Therefore, discrete-time implementations of continuous-time CBF-based safety filters continue to be a popular method for approximately guaranteeing the safety of real-world systems.

Our contributions in this work are three-fold: (i) we draw attention to practical issues of using standard continuous-time CBF-based safety filters in discrete-time implementations, (ii) we propose three practical strategies to mitigate the issues, and (iii) we verify our proposed strategies in simulation and real-world quadrotor experiments.

II. PROBLEM FORMULATION

In this work, we consider the control architecture shown in Fig. 2 and a continuous-time nonlinear control system in the following control-affine form:

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

where $x \in \mathbb{X} \subset \mathbb{R}^n$ is the state of the system with \mathbb{X} being the set of admissible states, $u \in \mathbb{R}^m$ is the input of the system, and $f: \mathbb{R}^n \mapsto \mathbb{R}^n$ and $g: \mathbb{R}^n \mapsto \mathbb{R}^{n \times m}$ are locally Lipschitz continuous functions. We assume that \mathbb{X} is a known compact set, and f and g are known functions.

The safe set $\mathbb{C} \subseteq \mathbb{X}$ is assumed to be given and is defined as the zero-superlevel set of a smooth function $h: \mathbb{R}^n \rightarrow \mathbb{R}$: $\mathbb{C} = \{x \in \mathbb{X} \mid h(x) \geq 0\}$ where the boundary of the safe set is $\partial\mathbb{C} = \{x \in \mathbb{X} \mid h(x) = 0\}$ with $\partial h(x)/\partial x \neq 0$ for all $x \in \partial\mathbb{C}$. Our goal is to modify a given, potentially uncertified control policy $\pi(x)$ with a safety filter based on a continuous-time CBF-based safety filter such that the system is safe (i.e., the system's state x stays inside a safe set \mathbb{C} if it starts inside of \mathbb{C}).

III. BACKGROUND

In this section, we introduce the necessary definitions and the relevant background on CBF-based safety filters (see Fig. 2).

Definition 1 (Extended class- \mathcal{K} function [5]): A function $\gamma: \mathbb{R} \rightarrow \mathbb{R}$ is said to be of class- \mathcal{K}_e if it is continuous, $\gamma(0) = 0$, and strictly increasing.

Definition 2 (Positively control invariant set): Let \mathcal{U} be the set of all bounded controllers $\nu: \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}^m$. A set $\mathbb{C} \subseteq \mathbb{X}$ is a positively control invariant set for the control system in (1) if $\forall x_0 \in \mathbb{C}, \exists \nu \in \mathcal{U}, \forall t \in \mathbb{T}_{x_0}^+, \phi(t, x_0, \nu) \in \mathbb{C}$, where $\phi(t, x_0, \nu)$ is the system's phase flow starting at x_0 under the controller ν , and $\mathbb{T}_{x_0}^+$ is the maximum time interval.

Definition 3 (Relative degree [17]): The system consisting of the dynamics equation in (1) and the output equation $y = h(x)$ has relative degree $s \in \{1, \dots, n\}$ in $\mathbb{X}_s \subseteq \mathbb{R}^n$ if it is s -th order differentiable and $L_g L_f^{i-1} h(x) = 0$ for $i \in \{1, \dots, s-1\}$ and $L_g L_f^{s-1} h(x) \neq 0$ for all $x \in \mathbb{X}_s$.

Intuitively, the relative degree specifies how often we have to differentiate h along the dynamics (1) until the control input appears. If $\mathbb{X}_s = \mathbb{X}$, then s is the global relative degree. Consequently, s is a local relative degree if $\mathbb{X}_s \subset \mathbb{X}$. We use the global relative degree in the following two definitions for CBFs and higher-order CBFs.

Definition 4 (CBF [5]): Let $\mathbb{C} \subseteq \mathbb{X}$ be the superlevel set of a continuously differentiable function $h: \mathbb{X} \rightarrow \mathbb{R}$, then h is a CBF if there exists a class- \mathcal{K}_e function γ such that for all $x \in \mathbb{X}$ the control system in (1) has a global relative degree of $s = 1$ and satisfies

$$\max_{u \in \mathbb{R}^m} [L_f h(x) + L_g h(x)u] \geq -\gamma(h(x)), \quad (2)$$

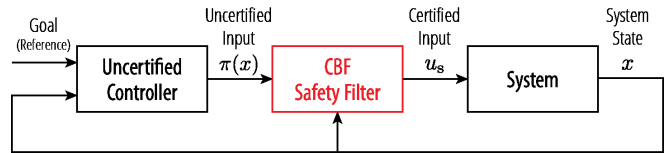


Fig. 2: A block diagram of a typical CBF-based safety filter framework. A CBF-based safety filter is augmented to an uncertified controller $\pi(x)$ and modifies the input of the uncertified controller if it is deemed unsafe.

where $L_f h(x)$ and $L_g h(x)$ are the Lie derivatives of h along f and g , respectively. In the following, we will also write $\dot{h}(x, u) = L_f h(x) + L_g h(x)u$ for simplicity.

Definition 5 (Higher-Order CBF [18]): Consider an s -th order continuously differentiable function $h: \mathbb{X} \mapsto \mathbb{R}$. Let $h_i(x), i \in \{1, \dots, s\}$ be defined as

$$h_i(x) = \dot{h}_{i-1}(x) + \gamma_i(h_i(x)), i \in \{1, \dots, s\} \quad (3)$$

where $h_0 = h$, and $\gamma_i(x), i \in \{1, \dots, s\}$ denote class \mathcal{K}_e functions and $\mathbb{C}_i(t), i \in \{1, \dots, s\}$ be defined by the superlevel set of $h_{i-1}, i \in \{1, \dots, s\}$. Then h is a Higher-Order CBF of relative degree s if there exists differentiable class \mathcal{K}_e functions $\gamma_i(x), i \in \{1, \dots, s\}$ such that for all $x \in \mathbb{X}$ the control system in (1) has a global relative degree of s and satisfies

$$\max_{u \in \mathbb{R}^m} [L_f^s h(x) + L_g L_f^{s-1} h(x)u + O(h(x))] \geq -\gamma_s(h_{s-1}(x)) \quad (4)$$

for all $x \in \mathbb{C}_1 \cap \dots \cap \mathbb{C}_s$, where $O(h)$ is given by

$$O(h) = \sum_{i=1}^{s-1} L_f^i (\gamma_{s-i} \circ h_{s-i-1})(x). \quad (5)$$

Using a CBF with relative degree $s = 1$, we can define an input set

$$\mathbb{U}_{\text{cbf}}(x) = \{u \in \mathbb{R}^m \mid \dot{h}(x, u) \geq -\gamma(h(x))\} \quad (6)$$

that renders the system positively control invariant, which we will refer to as safe [5]. This requires the selection of a class- \mathcal{K}_e function γ that yields a non-empty $\mathbb{U}_{\text{cbf}}(x)$ for all $x \in \mathbb{C}$. For an uncertified controller $\pi(x)$ that may not be designed to be safe, one can formulate a QP to augment the control input such that the system satisfies the CBF condition in (2) [19]:

$$u_s(x) = \underset{u \in \mathbb{R}^m}{\text{argmin}} \frac{1}{2} \|u - \pi(x)\|^2 \quad (7a)$$

$$\text{s.t. } \dot{h}(x, u) \geq -\gamma(h(x)). \quad (7b)$$

This is the safety filtering optimization that is solved inside the safety filter block in Fig. 2. Intuitively, the optimization problem in (7) finds an input in $\mathbb{U}_{\text{cbf}}(x)$ that is as close as possible to $\pi(x)$, where the closeness of the inputs is specified with respect to a chosen distance measure (e.g., the Euclidean norm in (7a)). For generic cases with a higher relative degree, we can define (6) and (7) analogously.

In practice, a continuous-time implementation of (7) is often infeasible. This is partially due to the fact that the system state and input commands are only processed at discrete times. Moreover, practical systems are subject to delays (e.g.,

the time taken for solving the QP and communication delays in the control system), which hinders an ideal realization of a continuous-time CBF safety filter. In most implementations, the safety filter (7) is solved and applied to a system in a discrete-time fashion.

IV. UNDESIRABLE EFFECTS OF DISCRETE-TIME IMPLEMENTATIONS OF CBF-BASED SAFETY FILTERS

In this section, we discuss an undesirable effect of discrete-time implementations of continuous-time CBF-based safety filters.

We consider the case, where (7) is solved at a sampling time $\Delta t > 0$ to approximate the continuous-time safety filter policy. Consider the system (1) at state $x(t_0)$ with $t_0 > 0$. Then, the certified control input is $u_s(x(t_0))$. The certified control input is applied over the time interval $t \in [t_0, t_0 + \Delta t)$. However, only the initial time step t_0 has been certified by the safety filter, such that safety for the open time interval $t \in (t_0, t_0 + \Delta t)$ is typically not guaranteed in discrete-time implementations of continuous-time CBF-based safety filters. Poor performance in discrete-time implementations can especially arise when $\|L_g L_f^{s-1} h(x)\| \rightarrow 0$. In these cases, relatively large inputs are permissible by the CBF condition. However, when the system is close to the boundary, the large input applied over the finite time interval could result in subsequent chattering effects and even constraint violations.

Another related issue is when $L_g L_f^{s-1} h(x) = 0$. This happens when the CBF has a local relative degree higher than s , which can often be non-trivial to determine in practice. For a simple ellipsoidal CBF candidate $h(x) = 1 - x^T P x$ with $P \succ 0$, one would need to check $x^T P g(x)$ is nonzero for all $x \in \mathbb{X}$ to determine if the relative degree is one to satisfy the CBF definition. Fig. 1 shows an example where the relative degree at certain $x \in \mathbb{X}$ is greater than one. At these states, the safety filter (7) becomes inactive (as $L_g h(x) = 0$). When $x \in \mathbb{C}$ and $L_g h(x) = 0$, the lower bound on the Lie derivative (7b) is trivially satisfied for any control input $u \in \mathbb{R}^m$. Therefore, the control input can be chosen to minimize (7a), such that $u_s(x) = \pi(x)$. While the Lie derivative $\dot{h}(x, u) = L_f h(x)$ is a constant for all $u \in \mathbb{R}^m$ at such a state x , we can have $g(x) \neq 0$ such that $\dot{x} = f(x) + g(x)u$ is nonzero for $u \in \mathbb{R}^m$. Although the Lie derivative is unaffected by the control input, the control input may still affect the closed-loop system dynamics. Therefore, at states x where $L_g h(x) = 0$, the safety filter allows the application of the unsafe control input $\pi(x)$ for at least the time interval $[t_0, t_0 + \Delta t)$. Since $\pi(x)$ may be any arbitrary control policy, this can lead to safe set violations, poor performance, or chattering (frequent switching between an active and inactive safety filter [20], [21]).

To avoid this undesirable closed-loop behaviour, we encourage practitioners to implement additional verification methods, either online or offline. We propose possible mitigation strategies in the next section.

V. MITIGATING UNDESIRABLE BEHAVIORS FOR DISCRETE-TIME IMPLEMENTATIONS

In this section, we discuss potential practical strategies for handling cases when $\|L_g L_f^{s-1} h(x)\| \rightarrow 0$ and even

$L_g L_f^{s-1} h(x) = 0$ (which by Def. 3 indicates a misspecified global relative degree). This list shall lend practitioners a set of methods to improve real-world implementations.

One way to mitigate the issues resulting from $\|L_g L_f^{s-1} h(x)\| \rightarrow 0$ is to let the sampling time $\Delta t \rightarrow 0$. In the limit, this results in a continuous-time implementation. However, as discussed above, this is not feasible in practice. Alternatively, a prediction model can be used to determine if a future time step is inside the safe set, e.g., $x(t + \Delta t) \in \mathbb{C}$. However, this can result in nonlinear constraints in the safety filter optimization problem, e.g., using Euler integration and a nonlinear CBF. Therefore, we aim to provide mitigation strategies that minimally increase the online computation.

A. Penalty Term

One method to handle the case of $\|L_g L_f^{s-1} h(x)\| \rightarrow 0$ is by modifying the safety filter objective function by adding a term that explicitly accounts for $L_g L_f^{s-1} h(x)$ becoming close to 0. Our new proposed safety filtering objective is

$$J(x) = \frac{1}{2} \|u - \pi(x)\|^2 + \frac{r}{2 \|L_g L_f^{s-1} h(x)\|^2} \|u - \pi_{\text{safe}}(x)\|^2, \quad (8)$$

where π_{safe} is a known safe backup control policy (e.g., a stabilizing controller that renders \mathbb{C} control invariant). The new objective (8) replaces the standard CBF safety filtering objective for all $\|L_g L_f^{s-1} h(x)\| > \epsilon$ where ϵ is a small positive number and $r > 0$ is a weighting parameter. The closer $\|L_g L_f^{s-1} h(x)\|$ gets to 0, the greater the impact of the second term in the safety filtering objective. In this case, the safety filter will track the safe backup control policy instead of the uncertified control policy $\pi(x)$. The weighting parameter r determines the balancing between the two terms when $\|L_g L_f^{s-1} h(x)\|$ is far from 0. To avoid numerical instabilities, we set $u_s(x) = \pi_{\text{safe}}(x)$ when we are in a state x such that $\|L_g L_f^{s-1} h(x)\| \leq \epsilon$.

This strategy requires almost no added computational effort. In practice, the design of the safe backup control policy will require some attention, such that the backup policy can return the system to states where $\|L_g L_f^{s-1} h(x)\| > \epsilon$. Otherwise, the system will continue using the backup control policy π_{safe} for all future time.

B. Modified Safe Set Design

The undesired behaviors can also be mitigated by accounting for the issue of $\|L_g L_f^{s-1} h(x)\| \rightarrow 0$ during the design of the CBF for a system. In the following, we introduce two practical modification strategies.

1) *Safe Set Transformation*: One method to avoid the effects of $\|L_g L_f^{s-1} h(x)\| \rightarrow 0$ during the design is by applying a transformation to the CBF, e.g., a translation or a rotation, which yields $\tilde{h}(x)$ and the new safe set $\tilde{\mathbb{C}}$, respectively. This transformation has to be chosen such that the closed-loop system from a specific initial condition x_0 is not affected by $\|L_g L_f^{s-1} \tilde{h}(x)\| \rightarrow 0$ for the transformed CBF. For example, if the underlying control policy π is known to be stabilizing for a certain subset of the safe set, we can safely use π in those states, e.g., the inactive safety filter does not jeopardize safety. Therefore, the goal of this strategy is to find a transformation, such that the states x

for which $\|L_g L_f^{s-1} \tilde{h}(x)\| \rightarrow 0$ are the states for which π is safe. The transformed CBF $\tilde{h}(x)$ is given by

$$\tilde{h}(x) = h(R(x - \delta)), \quad (9)$$

where $\delta \in \mathbb{R}^n$ represents the translation and $R \in \text{SO}(n) \subset \mathbb{R}^{n \times n}$ represents the rotation matrix with the special orthogonal group $\text{SO}(n)$ such that R satisfies $RR^\top = I$ with I being the identity matrix and $\det R = 1$. Following the transformation, it is not guaranteed that $\tilde{h}(x)$ is a valid CBF anymore and has to be verified before using it online [12].

Since it may be impractical to show that π is stabilizing for states x for which $\|L_g L_f^{s-1} \tilde{h}(x)\| \rightarrow 0$, there is also the option to combine the transformation with the penalty term as introduced in the previous subsection.

2) *Safe Set Approximation*: Our final proposed method to handle $\|L_g L_f^{s-1} h(x)\| \rightarrow 0$ is based on finding an alternative safe set $\tilde{\mathbb{C}}$ that can be represented by a set of valid CBFs $\{h_i(x)\}_{i=0}^q$, where $q \in \mathbb{N}$ is the number of CBFs. The functions h_i should be chosen in such a way that $\|L_g L_f^{s-1} h_i(x)\| > \epsilon$ for all $x \in \tilde{\mathbb{C}}$, where $\epsilon > 0$ is a parameter.

One possible choice for such functions h_i are affine functions of the form

$$h_i(x) = p_i^\top x + b_i, \quad (10)$$

where $p_i \in \mathbb{R}^n$ and $b_i \in \mathbb{R}$. This has the advantage that $\frac{\partial h_i}{\partial x}$ is independent of the state x and constant. This can simplify the choice of p_i : pick p_i such that $L_g h_i(x) = p_i^\top g(x) \neq 0$ for all $x \in \tilde{\mathbb{C}}$ such that a relative degree of 1 is achieved. However, a disadvantage is that this results in a convex set $\tilde{\mathbb{C}}$, which can be restrictive. One may also let the alternative safe set be an inner approximation of the original safe set, i.e., $\tilde{\mathbb{C}} \subseteq \mathbb{C}$ to satisfy the initial safety requirements. An improved approximation can be achieved by adding more constraints; the accuracy of the approximation and the computational demand have to be traded off in real-world applications.

VI. CASE STUDY: LTI SYSTEM WITH ELLIPSOIDAL CBF

In this section, we validate our proposed mitigation strategies in simulation and on a real-world quadrotor system. All of our CBF-based safety filters have been implemented using CasADi [22].

To demonstrate the undesirable effect, we first set up two examples (in simulation and the real world) where the issues discussed in Sec. IV occur. For both examples, we consider an ellipsoidal CBF of the form

$$h(x) = \beta - (x - c)^\top P(x - c), \quad (11)$$

where $\beta > 0$ defines the superlevel set, $c \in \mathbb{R}^n$ is the ellipsoid's center, and $P \in \mathbb{R}^{n \times n}$ is a positive definite matrix. Here we select $\beta = 1$, $P = \text{diag}(1, 31, 4, 00)$, $c = [0 \ 0]^\top$ for simulation and $c = [1.125 \ 0]^\top$ for real-world experiments.

A. Simulation Results

The system of interest for the simulation is the following continuous-time LTI system

$$\dot{x} = \underbrace{\begin{bmatrix} 0.00 & 1.00 \\ -0.09 & 0.10 \end{bmatrix}}_A x + \underbrace{\begin{bmatrix} 0 \\ 18.09 \end{bmatrix}}_B u, \quad (12)$$

which was identified from offline data collected on a real-world quadrotor system, with $x \in \mathbb{R}^2$ and $u \in \mathbb{R}$. This yields

$$L_g h(x) = -2x^\top P B. \quad (13)$$

Note that, for all states $x \in \mathbb{X}_{s \neq 1} = \{x \in \mathbb{R}^2 \mid x^\top P B = 0\}$, we have $L_g h(x) = 0$ with a relative degree $s \neq 1$; elsewhere we have $s = 1$. For any state in the neighbourhood of $\mathbb{X}_{s \neq 1}$, we have that $\|L_g h(x)\|$ is zero or close to zero.

Here, we select the class- \mathcal{K}_e function γ as the identity map. The uncertified control policy is $\pi(x) = -0.1$. We simulate the system using a sampling time $\Delta t = 0.001$ s for 15 s and always initialize the system to $x_0 = [0.5 \ -0.1]^\top$.

The closed-loop state and input trajectories for the uncertified controller and the standard discrete-time implementation of a continuous-time CBF-based safety filter are shown in Fig. 3a. The system leaves the safe set \mathbb{C} in both cases. For the certified control policy, this is caused by the closed-loop trajectory x_{u_s} entering the set of states where $L_g h(x) = 0$ (indicated by $\mathbb{X}_{s \neq 1}$). The CBF condition in (7b) certifies arbitrary control inputs for states $x \in \mathbb{X}_{s \neq 1}$, rendering the safety filter inactive. This results in the system starting to chatter (u_s ranging from -3.20 to 2.04 , see (a, right)) and eventually violating the safe set constraint \mathbb{C} . This is a result of the discrete-time implementation of the continuous-time safety filter.

In the following, we apply our mitigation strategies to the simulation example. These strategies pursue one of two goals: (i) prevent $\|L_g h(x)\| \rightarrow 0$ by modifying the CBF, or (ii) by using a safe control policy π_{safe} when $\|L_g h(x)\|$ is close to 0.

First, we use the additional penalty term. Here, we select the new objective function to substitute (7a) with our new objective in (8) and select $r = 1$ and $\epsilon = 10^{-8}$. The closed-loop state and input trajectories for the CBF-based safety filter with the modified objective function are shown in Fig. 3b. Initially, the closed-loop system behaves similarly to the standard CBF safety filter. Unlike the standard CBF safety filter, our modified safety filtering objective results in a switch to the backup control policy $\pi_{\text{safe}} = 0$ when it enters a neighbourhood of $\mathbb{X}_{s \neq 1}$. Therefore, no chattering or safe set violations occur.

The next mitigation strategy is achieved by transforming the safe set. For this example, we choose $\delta = 0$ and a two-dimensional rotation matrix R that rotates every state x with an angle $\theta = \frac{\pi}{6}$ around an axis normal to the x_1 - x_2 plane. Fig. 3c shows the closed-loop state and input trajectories for the CBF-based safety filter with the transformed CBF. The transformed CBF $\tilde{\mathbb{C}}$ also modifies the set of states for which $L_g h(x) = 0$ (see dashed green line in Fig. 3c). This allows the closed-loop system to safely pass through $\mathbb{X}_{s \neq 1}$ and its neighbourhood while applying $\pi(x)$ during the first couple of time steps and then stabilize at a state $x(T)$ that is not close to $\mathbb{X}_{s \neq 1}$, where $T > 0$ is the final time in the simulation. To achieve a useful transformation of the CBF, knowledge of which states yield $L_g h(x) = 0$ is required and necessitates an additional step in the offline design. As mentioned above, this strategy can also be used with the additional penalty term.

Finally, we find an alternative CBF to mitigate the undesirable impact of $L_g h(x) = 0$ on the closed-loop safety filtering

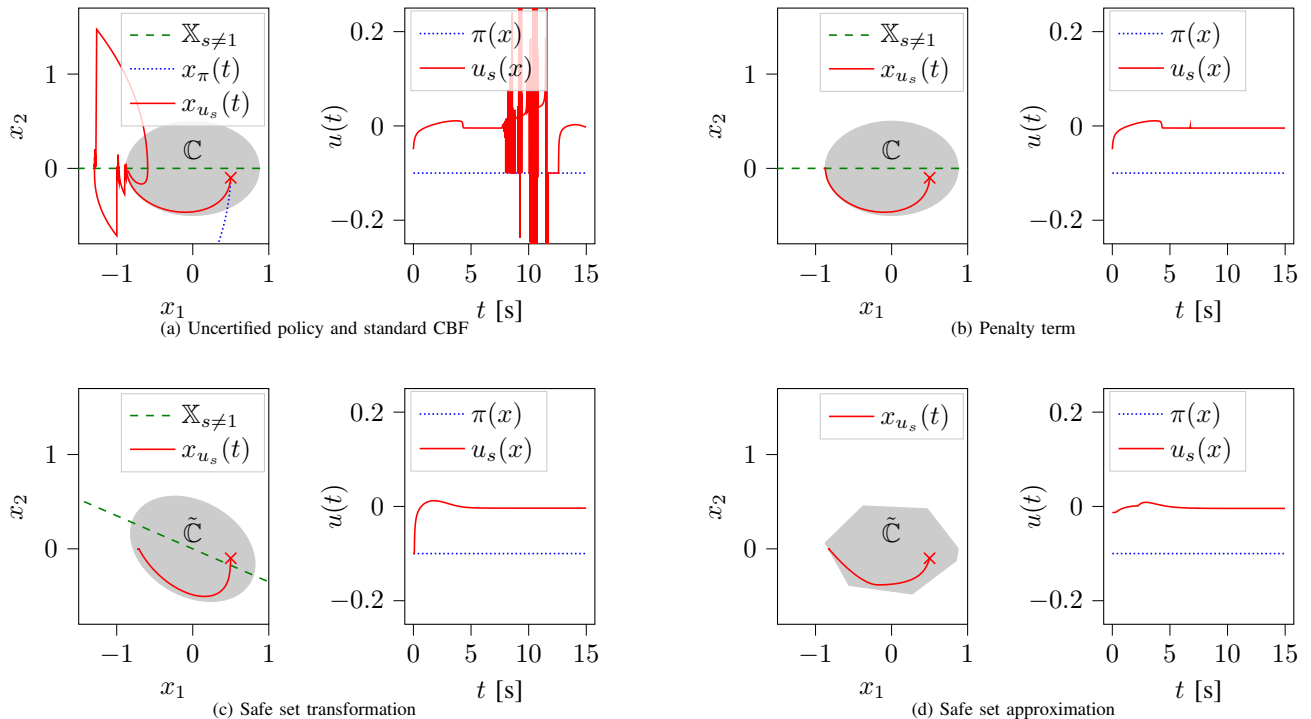


Fig. 3: Demonstration of the undesired closed-loop behaviour when $\|L_g h(x)\| \rightarrow 0$ (see (a)) and our proposed mitigation strategies (see (b), (c), and (d)) in simulation. The closed-loop trajectories in (a, left) show the state trajectories using the uncertified control policy $\pi(x)$ and the certified control policy $u_s(x)$, respectively. In both cases, the system leaves the safe set \mathcal{C} . For the certified control policy, this is caused by the closed-loop trajectory x_{u_s} entering the neighbourhood of states where $L_g h(x) = 0$ (indicated by $\mathbb{X}_{s \neq 1}$). This results in the system starting to chatter (see (a, right)) and eventually violating the safe set constraint \mathcal{C} . In (b), we prevent safety violations by adding our proposed penalty term to the safety filtering objective. Initially, the closed-loop system behaves similarly to the standard CBF-based safety filter. Then, the system switches to a backup control policy $\pi_{\text{safe}} = 0$ when it enters a neighbourhood of $\mathbb{X}_{s \neq 1}$. In (c), we successfully prevent chattering and safety violations by using a transformed safe set $\tilde{\mathcal{C}}$. This allows the closed-loop system to safely pass through the set $\mathbb{X}_{s \neq 1}$ during the first couple of time steps and then stabilize at the final state in the simulation far from $\mathbb{X}_{s \neq 1}$. Finally, in (d), we demonstrate the mitigation of $\|L_g h(x)\| \rightarrow 0$ by using an alternative safe set $\tilde{\mathcal{C}}$. Again, no safe set violations occur, as none of the affine constraints given by h_i are parallel to the input matrix B . Therefore, $\mathbb{X}_{s \neq 1}$ is empty.

behavior, see Fig. 3d. Again, no safe set violations occur, as none of the affine constraints given by h_i are orthogonal to the set $B^\perp = \{x \in \mathbb{R}^2, x^\top B = 0\}$. Therefore, this yields $L_g h_i(x) \neq 0, i \in \{1, \dots, 7\}$ for all states x in the modified safe set $\in \tilde{\mathcal{C}}$. We highlight that all the h_i have to be specifically chosen to avoid $p_i^\top B = 0$ (for the affine case).

The presented simulation results encourage determination for which states $L_g L_f^{s-1} h(x) = 0$ for the problem at hand. This enables finding an effective modification to the safe set \mathcal{C} . For high-dimensional systems or online adapted safe sets or systems (e.g., as in learning-based methods), the determination of $L_g L_f^{s-1} h(x) = 0$ can be computationally infeasible. In such cases, the additional penalty term is a better option.

B. Quadrotor Experiments

We perform physical experiments on a miniature quadrotor, the Crazyflie 2.1 [23], to verify the proposed mitigation strategies. A video of the experiments can be seen at this link: <http://tiny.cc/practicalCBF>. In the experiment, a CBF-based safety filter is supposed to prevent a falling quadrotor from colliding with the floor. We stabilize the attitude of the quadrotor such that the motion of the quadrotor is limited to its z -axis. Then the system's state reduces to $x = [z_{\text{pos}} \ z_{\text{vel}}]^\top \in \mathbb{R}^2$ and the applied input

is the collective thrust $u_{\text{total}} \in \mathbb{R}$. The elements of the state vector, z_{pos} and z_{vel} , are determined by a motion capture system and numerical differentiation, respectively. Instead of the collective thrust, we use the delta thrust as our control input $u = u_{\text{total}} - u_0$ where u_0 is the collective thrust for hovering with mass $m = 0.033$ kg and gravitational constant $g = 9.81 \frac{\text{m}}{\text{s}^2}$. This yields the following continuous-time LTI system:

$$\dot{x} = \underbrace{\begin{bmatrix} 0.00 & 1.00 \\ 0.00 & 0.00 \end{bmatrix}}_A x + \underbrace{\begin{bmatrix} 0 \\ 30.30 \end{bmatrix}}_B u \quad (14)$$

The set $\mathbb{X}_{s \neq 1}$ with relative degree $s \neq 1$ can be readily determined, and we note that the issue introduced in section Sec. IV also exists in our real-world system. The sampling time for the real-world experiments is $\Delta t = 0.167$ s. We highlight that in our real-world system, the sampling time is much larger than in our simulation. Recall that with a discrete-time implementation of a continuous-time CBF-based safety filter, safety is typically not guaranteed between consecutive sampling time instants. Therefore, more conservative, i.e., less steep, class- \mathcal{K}_e functions have to be chosen to prevent the system from approaching the safe set boundary too fast.

We begin our experiments by hovering at an approximated

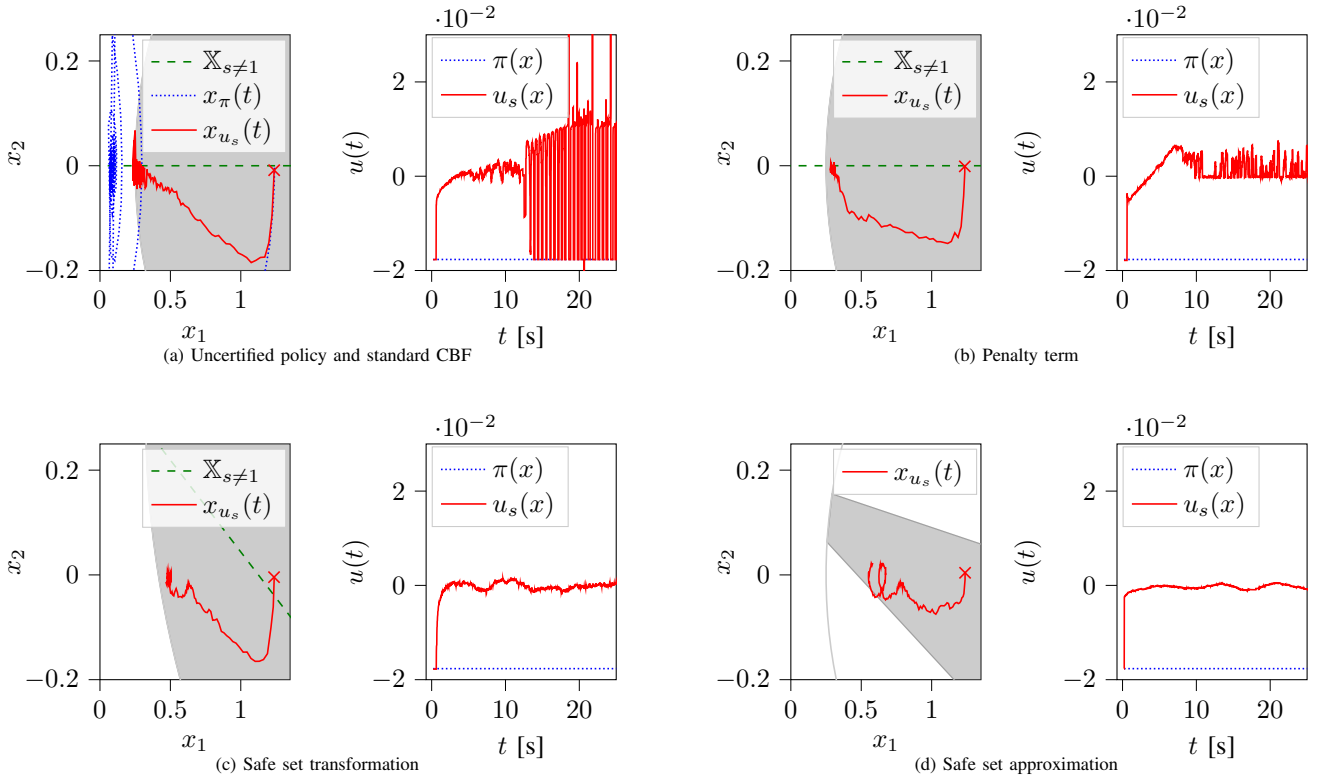


Fig. 4: Demonstration of the undesired closed-loop behavior when $\|L_g h(x)\| \rightarrow 0$ (see (a)) and our proposed mitigation strategies (see (b), (c), and (d)) on a real-world quadrotor system. The closed-loop trajectories in (a, left) show the state trajectories using the uncertified control policy $\pi(x)$ and the certified control policy $u_s(x)$, respectively. The quadrotor violates the safe set \mathbb{C} for both scenarios. For the standard CBF-based safety filter, chattering happens when the system enters the vicinity of set $\mathbb{X}_{s \neq 1}$, which causes the quadrotor to leave the safe set. In (b), with our proposed penalty formulation, the system switches to the backup control policy $\pi_{\text{safe}} = 0$ when it enters the neighbourhood of $\mathbb{X}_{s \neq 1}$ and chattering is significantly reduced. Most importantly, the quadrotor strictly stays inside the safe set in this experiment. Then, in (c), when using a transformed safe set $\tilde{\mathbb{C}}$, no chattering and safety violations can be observed. The quadrotor ends up hovering inside of the transformed safe set $\tilde{\mathbb{C}}$. In (d), we apply an alternative safe set $\hat{\mathbb{C}}$ to achieve $L_g h_i(x) \neq 0$ for all $x \in \hat{\mathbb{C}}$. The quadrotor violates the new smaller safe set for a few states. However, the system stays inside the original safe set \mathbb{C} throughout the experiment. No chattering is observed with this alternative safe set.

initial condition $x_0 = [1.25 \ 0]^T$. Then, the uncertified control policy $\pi(x) = -0.05mg$ is applied. In the different experiments on the real-world system, we demonstrate the closed-loop behavior of the system with no safety filter, the standard CBF-based safety filter, and the augmented CBF-based safety filters using each of our three proposed mitigation strategies.

The experimental results of the closed-loop state and input trajectories for the uncertified controller and the standard CBF safety filter are presented in Fig. 4a. Both cases result in the quadrotor leaving the safe set. For the standard CBF safety filter, when the system enters the neighbourhood of set $\mathbb{X}_{s \neq 1}$ where arbitrary control inputs are certified by (7b), chattering can be observed with u_s varying from $-0.14mg$ to $0.37mg$ at a high frequency.

The proposed mitigation strategies are validated in the following experiments. We first add the penalty term to our objective function, which results in the new objective (8). In this experiment, we select $r = 75$ and $\epsilon = 10^{-8}$. When the system enters the neighbourhood of $\mathbb{X}_{s \neq 1}$, a backup control policy $\pi_{\text{safe}} = 0$ is applied. This is a backup control policy, as all the states $x \in \mathbb{X}_{s \neq 1}$ have $x_2 = 0$, so x_1 will be constant for all future time. The state and input

trajectories with this modified objective function are shown in Fig. 4b. While chattering is not entirely prevented, its magnitude is significantly decreased with the certified control input $u_s \in [-0.0016mg, 0.02mg]$ during the timesteps when chattering occurs. Furthermore, the safety constraints are not violated during the closed-loop operation. We note that for real-world safety-critical applications, the parameter r can be picked conservatively so that the safety filter will tend to rather adopt the given safe backup control policy to avoid dangerous behaviors.

Secondly, we apply the transformed safe set with the same design parameters as in the simulation subsection to reduce the unfavorable effect caused by the system's state entering the neighbourhood of $\mathbb{X}_{s \neq 1}$. The real-world results are shown in Fig. 4c. The quadrotor safely passes through the vicinity of set $\mathbb{X}_{s \neq 1}$ and eventually, the augmented safety filter successfully achieves hover inside the transformed safe set $\tilde{\mathbb{C}}$. Additionally, no chattering is observed because of the transformed safe set $\tilde{\mathbb{C}}$.

Finally, we leverage an alternative CBF in our last experiment. We select a new set of five affine constraints that satisfy $p_i^T B \neq 0$ for $i = \{1, \dots, 5\}$ to accommodate the sampling effect of the discrete-time implementation on our real-world

systems. The new safe set and system trajectories are shown in Fig. 4d. We observe no chattering but minor violations of the new safe set for a few states. After the violation happens, the quadrotor is brought back into the safe set by the safety filter and remains there for the rest of the experiment. The model mismatch between the real-world system and our LTI model or the large sampling time could cause this violation. We also emphasize that, by choosing this alternative CBF, we sacrifice a large part of the original safe set, which is still rendered safe by this strategy.

VII. CONCLUSION

In this work, we highlight the issue of CBF-based safety filters becoming inactive in states where the norm of the Lie derivative of the CBF with respect to the input dynamics $\|L_g L_f^{s-1} h(x)\|$ is close to zero (or zero). In such states, the CBF-based safety filter certifies any control input as safe, including the uncertified control policy. For discrete-time implementations, this inactivity can lead to chattering and/or safe set violations as an uncertified control input can be applied for the sampling duration.

To prevent this issue, we propose three strategies to modify a standard CBF-based safety filter. Our first strategy aims at switching to a safe backup control policy close to states that cause the discussed issue using an additional term in the safety filtering objective. The second strategy transforms the safe set such that the undesired states are avoided in closed-loop operation with the safety filter. Our last strategy leverages an alternative safe set that is carefully designed such that undesired states do not exist in the new safe set. Finally, we demonstrate the presented issue and validate our three proposed mitigation strategies in simulation and on a real-world quadrotor.

REFERENCES

- [1] L. Brunke, M. Greeff, A. W. Hall, Z. Yuan, S. Zhou, J. Panerati, and A. P. Schoellig, "Safe learning in robotics: From learning-based control to safe reinforcement learning," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 5, pp. 411–444, 2022.
- [2] J. F. Fisac, A. K. Akametalu, M. N. Zeilinger, S. Kaynama, J. Gillula, and C. J. Tomlin, "A general safety framework for learning-based control in uncertain robotic systems," *IEEE Trans. on Automatic Control*, vol. 64(7), pp. 2737–2752, 2019.
- [3] K. P. Wabersich and M. N. Zeilinger, "Linear model predictive safety certification for learning-based control," in *Proc. of the IEEE Conf. on Decision and Control (CDC)*, 2018, pp. 7130–7135.
- [4] L. Brunke, S. Zhou, and A. P. Schoellig, "Barrier Bayesian linear regression: Online learning of control barrier conditions for safety-critical control of uncertain systems," in *Proc. of the Learning for Dynamics and Control Conf. (LADC)*, vol. 168, 2022, pp. 881–892.
- [5] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *Proc. of the European Control Conf. (ECC)*, 2019, pp. 3420–3431.
- [6] M. J. Khojasteh, V. Dhiman, M. Franceschetti, and N. Atanasov, "Probabilistic safety constraints for learned high relative degree system dynamics," in *Proc. of the Learning for Dynamics and Control Conf. (LADC)*, vol. 120, 2020, pp. 781–792.
- [7] T. Gurriet, A. Singletary, J. Reher, L. Ciarletta, E. Feron, and A. Ames, "Towards a framework for realizable safety critical control through active set invariance," in *2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCP)*, 2018, pp. 98–106.
- [8] M. Ohnishi, L. Wang, G. Notomista, and M. Egerstedt, "Barrier-certified adaptive reinforcement learning with applications to brushbot navigation," *IEEE Trans. on Rob.*, vol. 35(5), pp. 1186–1205, 2019.
- [9] A. Taylor, A. Singletary, Y. Yue, and A. Ames, "Learning for safety-critical control with control barrier functions," in *Proc. of the Learning for Dynamics and Control Conf. (LADC)*, vol. 120, 2020, pp. 708–717.
- [10] A. Singletary, W. Guffey, T. G. Molnar, R. Sinnet, and A. D. Ames, "Safety-critical manipulation for collision-free food preparation," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10954–10961, 2022.
- [11] Z. Jian, Z. Yan, X. Lei, Z. Lu, B. Lan, X. Wang, and B. Liang, "Dynamic control barrier function-based model predictive control to safety-critical obstacle-avoidance of mobile robot," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 3679–3685.
- [12] L. Brunke, S. Zhou, M. Che, and A. P. Schoellig, "Optimized control invariance conditions for uncertain input-constrained nonlinear control systems," *IEEE Control Systems Letters*, vol. 8, pp. 157–162, 2024.
- [13] W. Xiao, C. Belta, and C. G. Cassandras, "Event-triggered control for safety-critical systems with unknown dynamics," *IEEE Transactions on Automatic Control*, vol. 68, no. 7, pp. 4143–4158, 2023.
- [14] J. Zeng, B. Zhang, and K. Sreenath, "Safety-critical model predictive control with discrete-time control barrier function," in *Proc. of the IEEE American Control Conference (ACC)*, 2021, pp. 3882–3889.
- [15] R. K. Cosner, P. Culbertson, A. J. Taylor, and A. D. Ames, "Robust safety under stochastic uncertainty with discrete-time control barrier functions," in *2023 Robotics: Science and Systems (RSS)*, 2023.
- [16] A. J. Taylor, V. D. Dorobantu, R. K. Cosner, Y. Yue, and A. D. Ames, "Safety of sampled-data systems with control barrier functions via approximate discrete time models," in *2022 IEEE 61st Conference on Decision and Control (CDC)*, 2022, pp. 7127–7134.
- [17] H. K. Khalil, *Nonlinear Systems*, ser. Pearson Education. Prentice Hall, 2002.
- [18] W. Xiao and C. Belta, "High-order control barrier functions," *IEEE Transactions on Automatic Control*, vol. 67, no. 7, pp. 3655–3662, 2021.
- [19] A. D. Ames, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs with application to adaptive cruise control," in *Proc. of the IEEE Conf. on Decision and Control (CDC)*, 2014, pp. 6271–6278.
- [20] T. Koller, F. Berkenkamp, M. Turchetta, J. Boedecker, and A. Krause, "Learning-based model predictive control for safe exploration and reinforcement learning," 2019. [Online]. Available: <https://arxiv.org/abs/1906.12189>
- [21] F. P. Bejarano, L. Brunke, and A. P. Schoellig, "Multi-step model predictive safety filters: Reducing chattering by increasing the prediction horizon," in *2023 62nd IEEE Conference on Decision and Control (CDC)*, 2023, pp. 4723–4730.
- [22] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi – A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.
- [23] W. Giernacki, M. Skwierczyński, W. Witwicki, P. Wroński, and P. Kozierski, "Crazyflie 2.0 quadrotor as a platform for research and education in robotics and control engineering," in *2017 22nd International Conference on Methods and Models in Automation and Robotics (MMAR)*. IEEE, 2017, pp. 37–42.