Safety-Critical Control under Multiple State and Input Constraints and Application to Fixed-Wing UAV

Donggeon David Oh*, Dongjae Lee*, and H. Jin Kim

***Abstract*— This study presents a framework to guarantee safety for a class of second-order nonlinear systems under multiple state and input constraints. To facilitate real-world applications, a safety-critical controller must consider multiple constraints simultaneously, while being able to impose general forms of constraints designed for various tasks (e.g., obstacle avoidance). With this in mind, we first devise a zeroing control barrier function (ZCBF) using a newly proposed nominal evading maneuver. By designing the nominal evading maneuver to 1) be continuously differentiable, 2) satisfy input constraints, and 3) be capable of handling other state constraints, we deduce an ultimate invariant set, a subset of the safe set that can be rendered forward invariant with admissible control inputs. Thanks to the development of the ultimate invariant set, we then propose a safety-critical controller, which is a computationally tractable one-step model predictive controller (MPC) with guaranteed recursive feasibility. We validate the proposed framework in simulation, where a fixed-wing UAV tracks a circular trajectory while satisfying multiple safety constraints including collision avoidance, bounds on flight speed and flight path angle, and input constraints.**

I. Introduction

Safety is an essential factor to be considered in controller design. Safety in general encompasses various keywords including collision avoidance, input saturation, or task-space constraints [1] which frequently appear in control/robotics applications. When a *single* state constraint is imposed on the given system with actuation limits, one widely adopted method of safety-critical controller design is to find a zeroing control barrier function (ZCBF) by which a subset of the safe set, herein called the ZCBF set, is guaranteed to be rendered forward invariant by an admissible control law [2]–[6].

However, in many cases, *multiple* state constraints are imposed simultaneously on a system under input constraints. The most common approach for guaranteeing safety in the presence of such multiple constraints has been designing multiple ZCBFs, each of which is induced by a single state constraint, and then applying all ZCBF constraints at once in a quadratic program (QP) [7]–[9]. A major limitation of this approach is that the feasibility of the QP cannot be guaranteed; unlike each of the ZCBF sets, the intersection of the ZCBF sets may not be rendered forward invariant by any admissible control law.

Recently, a few strategies that attempt to alleviate such issue of controller infeasibility were presented. A QP with guaranteed feasibility that addresses multiple ZCBFs was formulated in [10], but the consideration of input constraints was left for future work. In [11], a controller that handles multiple ZCBFs as well as input constraints was proposed. However, the authors only considered the case of non-overlapping ZCBFs (i.e., ZCBFs with non-intersecting set boundaries), in which only one ZCBF acts at a time. Multiple ZCBFs that together ensure forward invariance of a safe set were constructed in [12], but the method is not applicable to state constraints such as obstacle avoidance constraints that cannot be written in the form of box constraints. In [13], a strategy for decoupling the design of multiple ZCBFs in the presence of input constraints was introduced, but it may result in an overly conservative viability domain (i.e., controlled invariant set) due to the idea of shrinking the set of available control inputs. To sum up, the existing methodologies either are applicable only to special cases of safety constraints, or may lead to an overly conservative invariant set.

The ability to handle multiple safety constraints in various forms is crucial for a safety-critical controller when considering its application to a real-world dynamical system, for instance, a fixed-wing unmanned aerial vehicle (UAV). In order to prevent overly aggressive maneuvers and aerodynamic stall which significantly deteriorate control performance, flight path angle and flight speed should be bounded [14], [15]. Furthermore, the UAV should avoid collision with the surrounding obstacles [16]. Lastly, control inputs should be constrained as per the actuation limits. Such constraints that naturally arise from the safety requirements of actual applications are unlikely to obey the non-overlapping assumption nor be represented in the form of box constraints.

Therefore, in this study, we present a framework to guarantee safety for a class of second-order nonlinear systems under input constraints and multiple state constraints. The presented method addresses two types of state constraints: one that can be formulated as a function of states with relative degree two, and box constraints that bound the states with relative degree one. Since the suggested method is able to handle any form of constraint function with relative degree two, it is applicable for tasks with complex workspace constraints including obstacle avoidance.

Our main contribution is the construction of the *ultimate*

* The first two authors contributed equally to this work.

This work was supported by Unmanned Vehicles Core Technology Research and Development Program through the National Research Foundation of Korea(NRF) and Unmanned Vehicle Advanced Research Center(UVARC) funded by the Ministry of Science and ICT(NRF-2020M3C1C1A010864).

D. D. Oh is with the Department of Mechanical and Aerospace Engineering, Seoul National University, Seoul 08826, South Korea donggeonoh1999@snu.ac.kr

D. Lee and H. Jin Kim are with the Department of Aerospace Engineering and the Automation and Systems Research Institute (ASRI), Seoul National University, Seoul 08826, South Korea ehdwo713@snu.ac.kr, hjinkim@snu.ac.kr

invariant set, which is a subset of the safe set that could be rendered forward invariant by an admissible control law. We do this by developing the method presented in [3], where a *nominal evading maneuver* was utilized to derive a ZCBF from a single constraint function. However, unlike the original method, we design a new continuously differentiable nominal evading maneuver that takes into account all the state constraints. The proposed nominal evading maneuver is designed specifically to render the ultimate invariant set in a non-conservative way, while being able to handle overlapping constraint functions (i.e. constraint functions whose 0-sublevel sets have intersecting boundaries). Then, we formulate a safety-critical one-step model predictive controller (MPC) with guaranteed recursive feasibility, which is suitable for real-time applications. The safety-critical controller is applied to a fixed-wing UAV, and we validate the proposed approach in simulation.

II. Preliminaries

A. Notations

The class of r-times continuously differentiable functions is denoted C^r. Let ∂S denote the boundary of a set S, and $;$ represent the empty set. Given a matrix $W \in \mathbb{R}^{n \times n}$ and a vector $x \in \mathbb{R}^n$, $\|x\|_W^2$ is equivalent to $x^\top W x$. $L_f h(x) = \frac{\partial h}{\partial x} f(x)$ denotes the Lie derivative of a function $h : \mathbb{R}^n \to \mathbb{R}$ along $f : \mathbb{R}^n \to \mathbb{R}^{n \times m}$ at point $x \in \mathbb{R}^n$. A function $\alpha : \mathbb{R} \to \mathbb{R}$ belongs to extended class-$\mathcal{K}_1$ if α is strictly increasing, $\alpha(0) = 0$, $\lim_{r \to \infty} \alpha(r) = \infty$, and $\lim_{r \to -\infty} \alpha(r) = -\infty$. $[N]$ is equivalent to the set $\{1, 2, \ldots, N\}$ for an integer $N \geq 1$. The subscript i denotes the i^{th} element of the related vector.

B. Safety and Set Invariance

Consider a nonlinear control-affine system

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

with state $x \in X \subset \mathbb{R}^n$ and control input $u \in U \subset \mathbb{R}^m$. U represents the set of admissible control inputs, and $f : X \to \mathbb{R}^n$ and $g : X \to \mathbb{R}^{n \times m}$ are locally Lipschitz.

Assume that a part of the state-space X should be avoided. We formulate such constraint using a continuously differentiable constraint function $h : X \to \mathbb{R}$ that defines the safe set S as the 0-sublevel set of h:

$$S := \{x \in X \mid h(x) \geq 0\}. \tag{2}$$

We aim to design a control law $u : X \to U$ that is guaranteed to keep the system (1) inside S. In this regard, we formally define safety using the concept of forward invariance.

Definition 1. **[17]** *Let $u = \kappa(x)$ be a feedback control law that induces the closed loop dynamics $\dot{x} = f_{cl}(x) := f(x) + g(x)\kappa(x)$ which is assumed to be locally Lipschitz. The set S is <u>forward invariant</u> if $x(t) \in S$ for all $x(0) \in S$ and $t \geq 0$. The closed-loop system $\dot{x} = f_{cl}(x)$ is <u>safe</u> with respect to the set S if S is forward invariant.*

A sufficient condition for the set S to be rendered forward invariant and thus achieve safety is h being a zeroing control barrier function, defined as follows.

Definition 2. **[17]** *Let $S \subset X$ be defined as (2) for a continuously differentiable function $h : X \to \mathbb{R}$. h is a <u>zeroing control barrier function (ZCBF)</u> if there exists an extended class-$\mathcal{K}_1$ function α such that for the control system (1) and for all $x \in X$:*

$$\inf_{u \in U} [L_f h(x) + L_g h(x) u \quad (-\alpha(h(x)))] \geq 0. \tag{3}$$

Theorem 1. **[3], [17]** *Let $S \subset X$ be defined as (2) for a continuously differentiable function $h : X \to \mathbb{R}$. If $h : X \to \mathbb{R}$ is a ZCBF on X and $\frac{\partial h}{\partial x}(x) \neq 0$ for all $x \in \partial S$, then any Lipschitz continuous controller $u : X \to U$ that satisfies*

$$L_f h(x) + L_g h(x) u(x) \quad (-\alpha(h(x))), \forall x \in S \tag{4}$$

renders S forward invariant.

If h satisfies (3) for some extended class-$\mathcal{K}_1$ function α, then h is a valid ZCBF, and its 0-sublevel set S is referred to as a *ZCBF set*. In practice, a constraint function is usually not a valid ZCBF unless it is designed specifically in consideration of (3). This is generally due to the following two issues [18]:

- **I1.** The set of admissible control inputs U restricts the available control actions.
- **I2.** If the constraint function h has a relative degree of $r \geq 2$ with respect to the system (1), no control authority exists since (3) simplifies to $L_f h(x) \quad (-\alpha(h(x))) \geq 0$.

In both cases, an extended class-$\mathcal{K}_1$ function α that satisfies (3) is unlikely to exist.

One way of addressing the issues I1 and I2 is to design a predefined *nominal evading maneuver* $u_{ev} : X \to U$ that attempts to drive the system towards the interior of S [2], [3]. The flow operator $\phi_h(t; x; u_{ev})$ represents the value $h(y(t))$ resulting from the initial value problem $\dot{y} = f(y) + g(y)u_{ev}(y)$, $y(0) = x$, and was used to define a new ZCBF candidate $H : X \to \mathbb{R}$ and the corresponding 0-sublevel set S_H in [3].

Theorem 2. **[3]** *The function $H : X \to \mathbb{R}$ defined as $H(x) := \sup_{t \geq 0} \phi_h(t; x; u_{ev})$ is a valid ZCBF, provided $S_H \not\subset \partial S$.*

Note that the ZCBF set S_H is a subset of the safe set S. A means to calculate H for general systems was also provided in [3], given $u_{ev} \in C^1$. Then, by Theorem 2, any locally Lipschitz controller $u : X \to U$ that satisfies $\dot{H}(x, u) \geq -\alpha(H(x))$ for an arbitrary extended class-$\mathcal{K}_1$ function α would render S_H forward invariant.

The method of using a nominal evading maneuver u_{ev} to define $H(x)$ alleviates the issues I1 and I2. This is because at least one admissible control law $u \in U$, namely the nominal evading maneuver u_{ev}, is assured to satisfy (3) for any extended class-$\mathcal{K}_1$ function α, and H has a relative degree of 1 with respect to the system (1).

III. Problem Formulation

A. Dynamical Model

We consider a class of second-order nonlinear systems of the form

$$\dot{x} = \begin{bmatrix} \dot{r} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} f_r(x) \\ f_v(x) \end{bmatrix} + \begin{bmatrix} 0_{n \times m} \\ g(x) \end{bmatrix} u; \tag{5}$$

with state $x := [r^\top; v^\top]^\top \in X \subset \mathbb{R}^{n+m}$ and control input $u \in U \subset \mathbb{R}^m$. Functions $f_r : X \to \mathbb{R}^n$, $f_v : X \to \mathbb{R}^m$, and $g : X \to \mathbb{R}^{m \times m}$ are at least C^2, where $g(x)$ is defined as $g(x) := \text{diag}[g_{v_1}(x); g_{v_2}(x); \ldots; g_{v_m}(x)]$ with $g_{v_i} : X \to \mathbb{R}$, $i \in [m]$. States of relative degree 2 along (5) are denoted $r \in X_r \subset \mathbb{R}^n$, and $v \in X_v \subset \mathbb{R}^m$ represents the states of relative degree 1, and $X = X_r \times X_v$. We will refer to r_i, $i \in [n]$ as *RD2 states*, and v_j, $j \in [m]$ as *RD1 states*. Dynamical models of a wide range of systems including fixed-wing UAVs [19], adaptive cruise control problems [17], [18], and spacecrafts [3], [4], [20] could be formulated in the form of (5).

B. Constraints

Three types of constraints are considered in this work: RD2 constraint, RD1 constraints, and input constraints.

The *RD2 constraint function* $h_r : X_r \to \mathbb{R}$ is formulated as a function of the RD2 states r, and assumed to be at least C^2. The associated safe set is defined as the 0-sublevel set of h_r:

$$S_r := \{ x = [r^\top; v^\top]^\top \in X \mid h_r(r) \geq 0 \}. \tag{6}$$

We will refer to S_r as the *RD2 safe set*. If a dynamical model of an Euler-Lagrange system could be represented in the form of (5), then r would be equivalent to the generalized coordinates. Therefore, an RD2 constraint can be applied ubiquitously for tasks that involve designing a safe set dictated by the position of a system, e.g., obstacle avoidance.

An RD1 constraint bounds the corresponding RD1 state in the form of a box constraint. Without loss of generality, we assume that RD1 constraints are applied to $v_1; v_2; \ldots; v_c$, where $c \leq m$. For an RD1 state v_i, $i \in [c]$, an *RD1 constraint function* is formulated as

$$h_{v_i}(v) := -\left(v_i - \frac{v_i^{min} + v_i^{max}}{2}\right)^2 + \left(\frac{v_i^{max} - v_i^{min}}{2}\right)^2; \tag{7}$$

where $v_i^{min} < v_i^{max}$ for all $i \in [c]$. The *RD1 safe set* is then defined as the 0-sublevel set of h_{v_i}:

$$S_{v_i} := \{ x = [r^\top; v^\top]^\top \in X \mid h_{v_i}(v) \geq 0 \}; \tag{8}$$

which is equivalent to $\{x \in X \mid v_i^{min} \leq v_i \leq v_i^{max}\}$. Such type of constraint is highly applicable for general automated systems [12].

The *safe set* is defined as the intersection of the RD2 safe set and all RD1 safe sets:

$$S = S_r \cap \left(\bigcap_{i=1}^{c} S_{v_i}\right). \tag{9}$$

Input constraints represent the limited actuation capabilities of real-world systems. The set of admissible control inputs is defined as follows:

$$U := \{ u \in \mathbb{R}^m \mid u_i^{min} \leq u_i \leq u_i^{max}, \forall i \in [m] \}; \tag{10}$$

where $u_i^{min} < u_i^{max}$ for all $i \in [m]$. Therefore, the problem we aim to solve can be formally stated as follows:

Problem 1. *Given the system (5), find a subset of the safe set S (9) that could be rendered forward invariant by an admissible control law $u : X \to U$. Then, design a safety-critical controller that is always feasible if the system state is inside such subset.*

As mentioned in Section I, there exist several methodologies that ensure forward invariance of a subset of the safe set by designing multiple ZCBFs. However, since the boundary of the RD2 safe set cannot be represented using box constraints and clearly intersects with the boundaries of RD1 safe sets, the strategies from [11], [12] cannot be applied to the system of interest. Moreover, the input constraints preclude the application of the methodology presented in [10]. To this end, instead of constructing multiple ZCBFs, we propose a new methodology to obtain an invariant set.

IV. Safety-Critical Controller Design

In this section, we first present issues that need to be addressed in the design of nominal evading maneuver. Then, a nominal evading maneuver which satisfies the input constraints and takes into account multiple state constraints is proposed. Using the nominal evading maneuver, an *ultimate safe set*, a subset of the safe set which can be rendered forward invariant using admissible control inputs, is defined. Finally, we construct a safety-critical one-step MPC with guaranteed feasibility that utilizes the invariance of the ultimate safe set.

A. Issues in Nominal Evading Maneuver Design

Recall the RD2 constraint function h_r and the RD2 safe set S_r. The RD2 constraint function is not likely to be a valid ZCBF because of the issues I1 and I2: the system is under input constraints (10), and the relative degree of h_r with respect to the system (5) is 2. The latter can be easily observed by computing derivatives of the RD2 constraint function $\dot{h}_r$ and $\ddot{h}_r$ as

$$\begin{aligned} \dot{h}_r(x) &= L_f h_r(x) + L_g h_r(x)\, u = L_f h_r(x); \\ \ddot{h}_r(x, u) &= L_f^2 h_r(x) + \sum_{i=1}^{m} d_i(x)\, u_i; \end{aligned}$$

where $d_i(x) := \frac{\partial h_r}{\partial v_i} g_{v_i}(x)$ for all $i \in [m]$. Therefore, we adopt the methodology presented in Theorem 2 and design a nominal evading maneuver $u_e : X \to U$ that allows us to define a valid ZCBF $H_r : X \to \mathbb{R}$. We will refer to H_r as the *RD2 ZCBF*. The corresponding *RD2 ZCBF set* S_{H_r} is a subset of S_r and is rendered forward invariant by a control law that satisfies the input constraints [3].

A nominal evading maneuver should be designed to effectively drive the system towards the interior of the RD2

ZCBF set S_{H_r}. However, since the relative degree of the RD2 constraint function is 2, both h_r and $\dot{h}_r$ cannot be manipulated directly with the control input u. Thus, one alternative approach for designing a greedy nominal evading maneuver u_{greedy} would be pointwise minimizing $\dot{h}_r$ as

$$u_{greedy}(x) := \underset{u \in U}{\operatorname{argmin}} \sum_{i=1}^{m} d_i(x)\, u_i. \tag{11}$$

Such nominal evading maneuver was shown to be effective for the task of safety-critical obstacle avoidance of a spacecraft in [3].

Unfortunately, u_{greedy} cannot be applied to the system (5) under the presence of RD1 constraints because of the following issues:

I3. Under the input constraints given as (10), u_{greedy} from (11) is equivalent to

$$u_{greedy_i}(x) = \begin{cases} u_i^{min} & \text{if } d_i(x) > 0 \\ u_i^{max} & \text{if } d_i(x) < 0 \end{cases} \tag{12}$$

for all $i \in [m]$. Therefore $u_{greedy}(x) \notin C^1$, and $\dot{h}_r$ cannot be calculated using the methodology presented in [3]. This is problematic because $\dot{h}_r$ is required to impose the ZCBF constraint in the form of (4).

I4. $u_{greedy}(x)$ does not take into account the RD1 constraints. For example, if $v_i = v_i^{min}$, $d_i(x) > 0$, and $g_{v_i}(x) > 0$ for some $i \in [c]$, then from Assumption 2 which will be presented shortly afterwards, $u_{greedy_i}(x)$ renders $\dot{v}_i < 0$. That is, u_{greedy} cannot constrain the RD1 states to be inside the corresponding RD1 safe sets.

To address the issues I3 and I4, in the next subsection, we propose a new nominal evading maneuver $\pi : X \to U$ that attempts to drive the system (5) towards the interior of S_{H_r}, while being at least C^1 and handling the RD1 constraints. Before entering the next subsection, we introduce a modified input $\tilde{u} : X \times U \to \mathbb{R}^m$ to consider nonzero $f_v(x)$ term in (5) and possibly asymmetric input constraints (i.e., $u_i^{min} \neq -u_i^{max}$).

The modified input is defined in an elementwise manner:

$$\tilde{u}_i(x;u) := \frac{f_{v_i}(x)}{g_{v_i}(x)} + u_i, \tag{13}$$

for all $i \in [m]$. Then, derivatives of the RD1 states are computed as

$$\dot{v}_i = f_{v_i}(x) + g_{v_i}(x)\, u_i = g_{v_i}(x)\, \tilde{u}_i(x;u) \tag{14}$$

for all $i \in [m]$. We see that $\dot{v}_i$ is solely dependent on a single input channel u_i. For the ease of controller design, we define functions $\underline{\alpha}_i : X \to \mathbb{R}$ and $\overline{\alpha}_i : X \to \mathbb{R}$ as

$$\underline{\alpha}_i(x) := u_i^{min} - \frac{f_{v_i}(x)}{g_{v_i}(x)}; \quad \overline{\alpha}_i(x) := u_i^{max} + \frac{f_{v_i}(x)}{g_{v_i}(x)} \tag{15}$$

for all $i \in [m]$, where $\underline{\alpha}_i(x)$ and $\overline{\alpha}_i(x)$ represent the minimum and maximum admissible values of the modified input $\tilde{u}_i(x;u)$ at $x \in X$.

We now state the two assumptions that are required to further the discussion of safety-critical controller design.

Assumption 1. $g_{v_i}(x) \neq 0$ for all $i \in [m]$ and $x \in X$.

Assumption 2. $\underline{\alpha}_i(x) < 0 < \overline{\alpha}_i(x)$ for all $i \in [m]$ and $x \in X$.

The two assumptions are essential to ensure that the system (5) maintains sufficient control authority and are also widely underlain in other existing works [3], [12]. If either of the two assumptions is violated, then the system loses control authority for v_i or the sign of $\dot{v}_i = g_{v_i}(x)(u_i + \frac{f_{v_i}(x)}{g_{v_i}(x)})$ becomes uncontrollable at some $x \in X$. Therefore, Assumptions 1 and 2 are essential for controlling the given system (5), let alone guaranteeing its safety.

Next, to consider input constraints for the modified input $\tilde{u}_i$, for all $i \in [m]$, we define $\tilde{u}_i^{max} : X \to \mathbb{R}$ as a smooth approximation of $\min(\underline{\alpha}_i(x), \overline{\alpha}_i(x))$:

$$\tilde{u}_i^{max}(x) := \frac{\underline{\alpha}_i(x) + \overline{\alpha}_i(x) - \sqrt{(\underline{\alpha}_i(x) - \overline{\alpha}_i(x))^2 + \varepsilon}}{2}; \tag{16}$$

where $\varepsilon \in \mathbb{R}_+$ is a small scalar for numerical stability. Such smooth approximation of the $\min(\cdot,\cdot)$ operator is adopted in order to design a continuously differentiable nominal evading maneuver and therefore resolve the issue I3. From Assumption 2, by taking sufficiently small ε that satisfies $\varepsilon < 4\,\underline{\alpha}_i(x)\,\overline{\alpha}_i(x)$ for all $i \in [m]$ and $x \in X$, $\tilde{u}_i^{max}(x)$ is assured to be greater than 0. Any modified input $\tilde{u}_i(x;u)$ that satisfies $-\tilde{u}_i^{max}(x) \leq \tilde{u}_i(x;u) \leq \tilde{u}_i^{max}(x)$ is admissible with respect to the input constraints (10). We utilize this property later in Lemma 2.

B. Nominal Evading Maneuver Design

We first design a nominal evading maneuver in terms of the modified input $\tilde{u}_i : X \to \mathbb{R}$ for all $i \in [m] \setminus [c]$. In this case, we need not worry about the issue I4 because for all $i \in [m] \setminus [c]$, the RD1 state v_i, which is dictated by $\tilde{u}_i$, is not constrained. However, we do need to make sure that d_i is at least C^1 to alleviate the issue I3. In this regard, we propose the following nominal evading maneuver in terms of the modified input $\tilde{u}_i$ for all $i \in [m] \setminus [c]$ that mimics the greedy control law u_{greedy} from (11):

$$\tilde{u}_i(x) := -\tilde{u}_i^{max}(x) \tanh(-k_i\, d_i(x)), \tag{17}$$

where $k_i \in \mathbb{R}_+$ is a control gain. $\tilde{u}_i(x)$ quickly approaches $\tilde{u}_i^{max}(x)$ as $d_i(x)$ increases from zero, and converges to $-\tilde{u}_i^{max}(x)$ as $d_i(x)$ decreases from zero. Such behavior is similar to that of u_{greedy_i} from (12) with the difference being that $\tilde{u}_i$ is continuously differentiable, while u_{greedy_i} is not.

On the contrary, for all $i \in [c]$, the RD1 state v_i should be constrained inside the RD1 safe set S_{v_i} defined as (8). Therefore, for all $i \in [c]$, the nominal evading maneuver in terms of the modified input $\tilde{u}_i : X \to \mathbb{R}$ should attempt to drive the system towards the interior of S_{H_r}, while addressing the issues I3 and I4. We present such nominal evading maneuver $\tilde{u}_i$ for all $i \in [c]$ in (18), where $v_i^d = v_i^{max} - v_i^{min}/2$, $v_i^s = v_i^{max} + v_i^{min}/2$, and $k_{i;1}, k_{i;2}, k_{i;3} \in \mathbb{R}_+$ are control gains.

It can be seen from (18) that the nominal evading maneuver in terms of the modified input $\tilde{u}_i$ is continuously differentiable for all $i \in [c]$, thereby alleviating the issue I3. Moreover, $\tilde{u}_i$ attempts to effectively drive the system towards

$$\tilde{u}_i(x) := \tilde{u}_i^{max}(x)\tanh\left(-k_{i;1}g_{v_i}(x)\right)\tanh\left(k_{i;2}(v_i - v_i^d)\tanh\left(-k_{i;3}g_{v_i}(x)d_i(x)\right) + (v_i^s - v_i^d)\right) \quad (18)$$

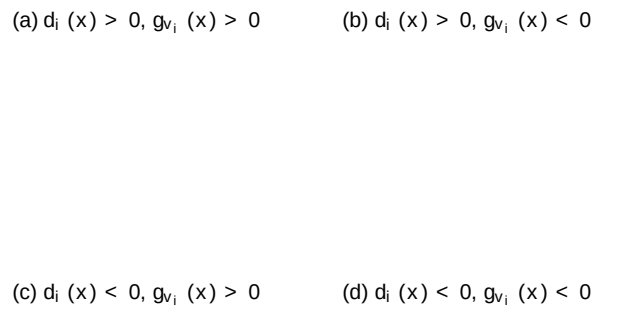


Fig. 1: Plots of the nominal evading maneuver in terms of the modified input $\tilde{u}_i(x)$ against v_i for $i \in [c]$ given different signs of $d_i(x)$ and $g_{v_i}(x)$. $\tilde{u}_i(x)$ pointwise minimizes $h_r(x,u)$ while assuring v_i to remain inside the RD1 safe set $\mathcal{S}_{v_i}$.

the interior of the RD2 ZCBF set, while resolving the issue I4 by guaranteeing v_i to remain inside the RD1 safe set $\mathcal{S}_{v_i}$ for all $i \in [c]$. We first provide a brief explanation of such property using plots of $\tilde{u}_i$ against v_i, and then present a formal proof in the next subsection.

Plot of the nominal evading maneuver in terms of the modified input $\tilde{u}_i(x)$ against v_i given $d_i(x) > 0$ and $g_{v_i}(x) > 0$ is shown in Fig. 1a. The nominal evading maneuver is designed to satisfy $\dot{v}_i(x) \geq 0$ if $v_i \leq v_i^{min}$, resulting in $\dot{v}_i \geq 0$ from (14). As v_i increases from v_i^{min}, $\tilde{u}_i(x)$ quickly converges to $\tilde{u}_i^{max}(x)$. Similar analyses can be done to Figs. 1b, 1c and 1d. Taken together, $\tilde{u}_i(x)$ attempts to pointwise minimize $\dot{h}_r$ in a similar manner as the greedy control law $u_{greedy_i}(x)$ from (12). However, unlike $u_{greedy_i}(x)$, $\tilde{u}_i(x)$ is able to guarantee v_i to remain inside the corresponding RD1 safe set $\mathcal{S}_{v_i}$. This will be formally proven in Theorem 3.

We now reformulate the nominal evading maneuver in terms of the original control input. Using the definition of the modified input from (13) and $\tilde{u}_i$ defined as (17) for $i \in [m] \setminus [c]$ and (18) for $i \in [c]$, the nominal evading maneuver in terms of the original input $u : X \to U$ is derived as follows:

$$u_i(x) = \tilde{u}_i(x) - \frac{f_{v_i}(x)}{g_{v_i}(x)}, \forall i \in [m]. \quad (19)$$

We restate the assessment of Fig. 1 in the following lemma.

Lemma 1. The nominal evading maneuver u from (19) renders the derivative of an RD1 constraint function non-positive, i.e., $\dot{h}_{v_i}(x,u) \leq 0$, for all $x \in \partial\mathcal{S}_{v_i}$, $i \in [c]$.

proof. For all $i \in [c]$, the derivative of $h_{v_i}(v)$ under the system dynamics (5) and the control law is given as

$$\dot{h}_{v_i}(x,u) = 2(v_i - \frac{v_i^{min} + v_i^{max}}{2})g_{v_i}(x)\tilde{u}_i(x), \quad (20)$$

where $\tilde{u}_i(x)$ is the nominal evading maneuver in terms of the modified input from (18).

First, we consider the case where $v_i = v_i^{max}$. Since $\tanh(-k_{i;3}g_{v_i}(x)d_i(x)) \leq 1$,

$$v_i^{max} - v_i^d\tanh(-k_{i;3}g_{v_i}(x)d_i(x)) + v_i^s - v_i^d \geq 0.$$

Consequently, the second $\tanh$ function of (18) is always greater than or equal to 0. If $g_{v_i}(x) > 0$, then the first $\tanh$ function of (18) is less than 0, thereby rendering $\dot{h}_{v_i}(x,u) \leq 0$. If $g_{v_i}(x) < 0$, then the first $\tanh$ function of (18) is greater than 0, and thus $\dot{h}_{v_i}(x,u) \leq 0$.

$\dot{h}_{v_i}(x,u)$ being rendered nonpositive for the case when $v_i = v_i^{min}$ could be shown in a similar way. Therefore, $\dot{h}_{v_i}(x,u) \leq 0$ for all $x \in \partial\mathcal{S}_{v_i}$, $i \in [c]$.

We also emphasize that the nominal evading maneuver is guaranteed to satisfy the input constraints from (10), as will be shown in the following lemma.

Lemma 2. The nominal evading maneuver defined as (19) satisfies $u(x) \in U$ for all $x \in X$.

proof. For all $i \in [m]$, the nominal evading maneuver in terms of the modified input given as (17) or (18) satisfies $|\tilde{u}_i(x)| < \tilde{u}_i^{max}(x)$ for all $x \in X$. Since $\tilde{u}_i^{max}(x) < \min(\alpha_i(x); -\beta_i(x))$ from (16), $\alpha_i(x) < \tilde{u}_i(x) < -\beta_i(x)$. Therefore, from (15) and (19), $u_i^{min} < u_i(x) < u_i^{max}$, and the nominal evading maneuver satisfies $u(x) \in U$ for all $x \in X$.

C. Ultimate Invariant Set

In this subsection, we define the ultimate invariant set $\mathcal{S}_u$ which is a subset of the safe set $\mathcal{S}$ (9), and show the existence of an admissible control law $u : X \to U$ that renders $\mathcal{S}_u$ forward invariant.

The RD2 ZCBF $H_r : X \to \mathbb{R}$ is defined using the methodology presented in Theorem 2 with the nominal evading maneuver u from (19):

$$H_r(x) := \sup_{t \geq 0} h_r(t;x;u). \quad (21)$$

The RD2 ZCBF set $\mathcal{S}_{H_r}$ is defined as the 0-sublevel set of H_r: $\mathcal{S}_{H_r} := \{x \in X \mid H_r(x) \geq 0\}$. We assume $\frac{\partial H_r}{\partial x}(x) \neq 0$ for all $x \in \partial\mathcal{S}_{H_r}$. For later brevity, we will refer to the RD2 ZCBF set $\mathcal{S}_{H_r}$ and the RD1 safe sets $\mathcal{S}_{v_1}, \mathcal{S}_{v_2}, \ldots, \mathcal{S}_{v_c}$ as the ultimate safe sets. We define the ultimate invariant set $\mathcal{S}_u$ as the intersection of all ultimate safe sets:

$$\mathcal{S}_u = \mathcal{S}_{H_r} \cap (\bigcap_{i=1}^{c} \mathcal{S}_{v_i}). \quad (22)$$

Note that since the RD2 ZCBF set $\mathcal{S}_{H_r}$ is a subset of the RD2 safe set $\mathcal{S}_r$, the ultimate invariant set $\mathcal{S}_u$ is a subset of the safe set $\mathcal{S}$ (9). In other words, all $x \in \mathcal{S}_u$ obeys the RD2 constraint as well as every RD1 constraint.

We now show that the ultimate invariant set S_u could be rendered forward invariant by a controller which satisfies the input constraints given as (10).

Theorem 3. There exists a controller $u : X \to U$ that renders the ultimate invariant set S_u defined as (22) forward invariant, while satisfying the input constraints given as (10).

proof. We prove the theorem using Nagumo's theorem [21]. Since the ultimate invariant set S_u is defined as the intersection of multiple ultimate safe sets, $x \in \partial S_u$ could be on the boundary of a single ultimate safe set, or where boundaries of multiple ultimate safe sets intersect.

We first consider the case where $x \in \partial S_{H_r}$ while $x \notin \partial S_{v_i}$ for all $i \in [c]$. The nominal evading maneuver u_r defined as (19), which always belongs to U by Lemma 2, renders $\dot{H}_r(x, u_r) \leq 0$ for all $x \in S_{H_r}$ [3]. In fact, $\dot{H}_r(x, u_r)$ would be rendered nonpositive for all $u : X \to U$ satisfying $u \in C^1$.

We then consider the case where $x \in \partial S_u$ while $x \notin \partial S_{H_r}$ (i.e., x is on the boundary of one or more RD1 safe sets). Without loss of generality, we assume $x \in \partial S_{v_i}$ for all $i \in [k]$, where $1 \leq k \leq c$. Since the derivative of an RD1 constraint function $\dot{h}_{v_i}(x, u)$ depends only on a single input channel u_i as can be seen in (20), the nominal evading maneuver u_r from (19) simultaneously renders $\dot{h}_{v_i}(x, u_r) \geq 0$ for all $i \in [k]$ by Lemma 1.

Finally, we consider the case where $x \in \partial S_{H_r}$ and $x \in \partial S_{v_i}$ for all $i \in [k]$, where $1 \leq k \leq c$. The nominal evading maneuver u_r defined as (19) yields $\dot{H}_r(x, u_r) < 0$ while simultaneously rendering $\dot{h}_{v_i}(x, u_r) \geq 0$ for all $i \in [k]$. Other forms of nominal evading maneuver that do not take into account the RD1 constraints would still render $\dot{H}_r(x, u_r) \leq 0$ as discussed earlier in this proof, but would not be able to guarantee the nonpositiveness of $\dot{h}_{v_i}(x, u_r)$ at the same time.

The above cases show that for all $x \in \partial S_u$, the derivatives of all RD2 ZCBF or RD1 constraint functions with zero value at x are rendered nonpositive by at least one admissible control law $u : X \to U$, namely the nominal evading maneuver u_r. Furthermore, $\frac{\partial \dot{h}_{v_i}}{\partial x}(x) \neq 0$ for all $x \in \partial S_{v_i}$ and $i \in [c]$, and $\frac{\partial \dot{H}_r}{\partial x}(x)$ is assumed to be nonzero for all $x \in \partial S_{H_r}$. Therefore, by Nagumo's theorem, the ultimate invariant set S_u could be rendered forward invariant by a controller that satisfies the input constraints given as (10).

Remark 1. It is crucial to compute a sufficiently large ultimate invariant set S_u inside the safe set S [3]. Unlike the prior works that design multiple ZCBFs from the state constraints [10]–[13], we did not construct additional ZCBFs for the RD1 constraints. Instead, in Theorem 3, we have shown that the intersection of the RD2 ZCBF set and the RD1 safe sets could be rendered forward invariant with an admissible control law. This enables us to fully utilize the RD1 safe sets, not the ZCBF sets that are subsets of the RD1 safe sets, in the construction of S_u, which results in a less conservative invariant set.

Remark 2. Since the greedy control law u_{greedy} from (11) pointwise minimizes $\dot{H}_r$ and therefore attempts to minimize $\dot{H}_r$, it may help to obtain a less conservative RD2 ZCBF set S_{H_r} when compared to other designs of nominal evading maneuver that does not minimize $\dot{H}_r$. Therefore, the nominal evading maneuver u_r from (19), which is specifically designed to mimic u_{greedy}, may as well result in a less conservative ultimate invariant set S_u.

D. Safety-Critical One-Step MPC

Here, we present a safety-critical controller with guaranteed feasibility that utilizes the results from Theorem 3. Similar to existing safety-critical controllers, we first impose the RD2 constraint in the form of ZCBF constraint (4). On the contrary, since we did not design ZCBFs for the RD1 constraints to obtain a less conservative invariant set, RD1 constraints cannot be written as input affine constraints (4). Instead, they should be formulated as $\dot{h}_{v_i}(v) \geq 0$ for all $i \in [c]$, which are not applicable to a QP. This motivates the formulation of a safety-critical one-step MPC that constrains the system to be inside the ultimate invariant set S_u:

$$u_{t;\,safe} = \underset{u_t \in U}{\operatorname{argmin}} \; \|u_t - u_0\|^2_{R_1} + \|u_t - u_{t-1}\|^2_{R_2}; \tag{23a}$$

$$\text{s.t. } x_{t+1} = F(x_t, u_t); \tag{23b}$$

$$\dot{H}_r(x_t, u_t) \leq -\gamma(-H_r(x_t)); \tag{23c}$$

$$h_{v_i}(v_{t+1}) \geq 0, \forall i \in [c]; \tag{23d}$$

where subscript t denotes the value of the related vector at time t. The first term of the cost function (23a) is similar to that of CBF-QP [17] except that it is now weighted by a positive definite matrix $R_1 \in \mathbb{R}^{m \times m}$, where $u_0 : X \to U$ is a nominal feedback controller that achieves some performance goal (e.g., trajectory tracking), but without consideration of state constraints. The second term of (23a), which is weighted by a positive semi-definite matrix $R_2 \in \mathbb{R}^{m \times m}$, is added to decrease chattering, and does not hinder safety guarantee. The condition imposed on u_t in (23a) represents the input constraints from (10), and (23b) describes the system dynamics (5) in the discrete-time domain. (23c) represents the RD2 ZCBF constraint (4), and the RD1 constraints are implemented as (23d). Together, (23c) and (23d) assure the system state to remain inside the ultimate safe set S_u.

Proposition 1. The safety-critical one-step MPC (23) is always feasible, given $x_t \in S_u$.

proof. The nominal evading maneuver u_r from (19) renders $\dot{H}_r(x_t, u_r) \leq -\gamma(-H_r(x_t))$ for all $x_t \in S_u \subseteq S_{H_r}$ [3]. Moreover, as shown in the proof of Theorem 3, the derivatives of all RD1 constraint functions with zero value at x_t are rendered nonpositive by u_r, and thus assures $h_{v_i}(v_{t+1}) \geq 0$ for all $i \in [c]$. Therefore, at least one admissible control input, namely u_r, simultaneously satisfies the constraints (23c) and (23d).

The proposed framework is guaranteed to be recursively feasible since the controller (23) is feasible given $x_t \in S_u$ and the resulting optimal input yields $x_{t+1} \in S_u$. Thus, we formulate a one-step MPC to enhance real-time applicability.

V. Application to Fixed-Wing UAV

In this section, we demonstrate the safety-critical controller (23) in simulation on a fixed-wing UAV.

A. Dynamical Model and Constraints

The dynamical model of a fixed-wing UAV is given as follows [19], [22]:

$$\begin{bmatrix} \dot{P}_x \\ \dot{P}_y \\ \dot{P}_z \\ \dot{V} \\ \dot{\chi} \\ \dot{\gamma} \end{bmatrix} = \begin{bmatrix} V\cos\gamma\cos\chi \\ V\cos\gamma\sin\chi \\ V\sin\gamma \\ u_V \\ (u_\chi - g\cos\gamma)/V \\ -u_\chi(V\cos\gamma) \end{bmatrix}, \tag{24}$$

where the system state and control input are defined as $x = [P_x; P_y; P_z; V; \chi; \gamma]^\top \in \mathbb{R}^6$ and $u = [u_V; u_\chi; u_\gamma]^\top \in \mathbb{R}^3$. The RD2 states $P := [P_x; P_y; P_z]^\top$ represent the position of the UAV in the inertial coordinate frame. Flight speed, flight path angle, and heading angle of the UAV, which are denoted V, γ, and χ, correspond to the RD1 states. Note that (24) could be written in the form of (5).

The UAV is subject to multiple safety constraints. First, we consider an RD2 constraint function $h_{obs} : \mathbb{R}^3 \to \mathbb{R}$ which is designed for the task of avoiding a spherical obstacle:

$$h_{obs}(P) = (R + R_{obs} + R_{min})^2 - \|P - P_{obs}\|^2, \tag{25}$$

where R, R_{obs}, and R_{min} represent radius of the sphere that circumscribes the UAV, radius of the spherical obstacle, and the minimum distance to be maintained between the UAV and the obstacle. P_{obs} is the position of the center of the obstacle. The corresponding RD2 ZCBF is obtained from (21). Next, the RD1 states V and γ should be bounded in the form of box constraints in order to prevent aggressive maneuvers and aerodynamic stall [14], [15]. Such constraints are formulated using the RD1 constraint function (7). Finally, the system (24) is under input constraints that arise from the actuator limits of real-world dynamical systems. We represent the set of admissible inputs U in the form of (10).

B. Simulation Results

As for the nominal feedback controller π_0, we utilize an MPC that tracks a circular reference trajectory. To avoid confusion with the safety-critical one-step MPC, we will refer to the nominal feedback controller as the nominal MPC. The control horizon of the nominal MPC is set as 0.5 s and its cost function is formulated as a sum of quadratic terms that are designed for trajectory tracking and input regulation. The nominal MPC takes into account the input constraints, however the RD2 and RD1 constraints are not considered. To demonstrate the validity of the proposed safety-critical controller, a circular reference trajectory that penetrates the obstacle as can be seen in Fig. 4 is provided. Collision is unavoidable without an additional safety-critical controller.

We compare two types of safety-critical controllers: the proposed safety-critical one-step MPC and the ZCBF-based controller presented in [3]. The two resultant features of the proposed controller compared to the other are that 1) additional RD1 constraints can be satisfied and 2) input chattering can be reduced. In implementing the ZCBF-based controller, owing to the originally adopted nominal evading maneuver in [3] not being continuously differentiable, we slightly modify the method by using the continuously differentiable nominal evading maneuver (17) instead. Such controller will be referred to as "modified [3]" in the following figures.

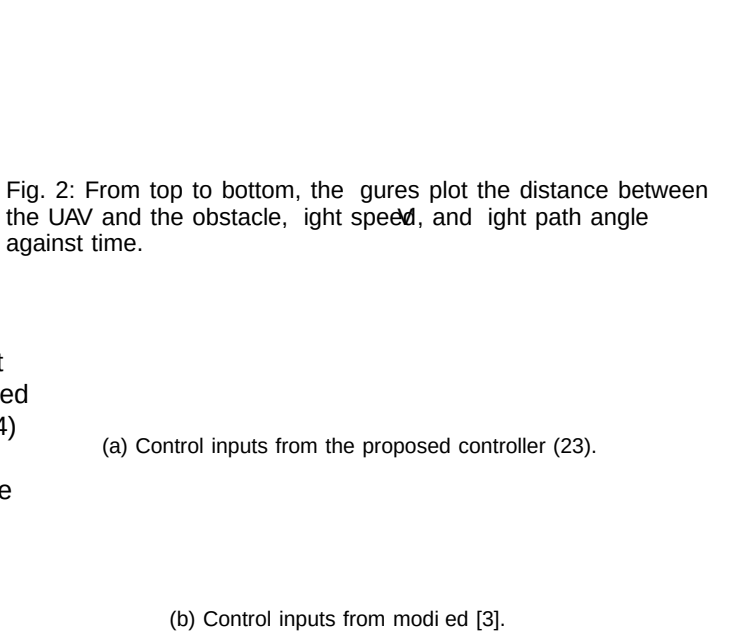
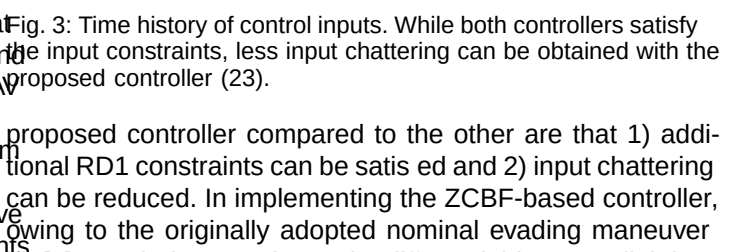
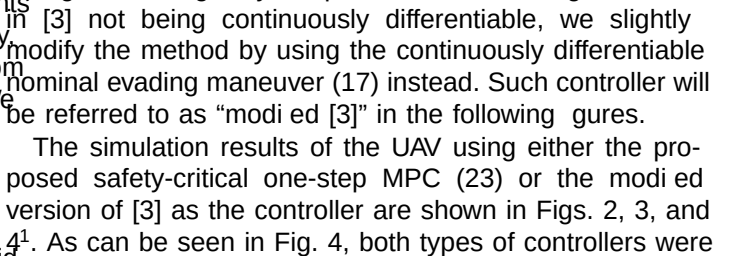


Fig. 2: From top to bottom, the figures plot the distance between the UAV and the obstacle, flight speed V, and flight path angle γ against time.



(a) Control inputs from the proposed controller (23).



(b) Control inputs from modified [3].

Fig. 3: Time history of control inputs. While both controllers satisfy the input constraints, less input chattering can be obtained with the proposed controller (23).

The simulation results of the UAV using either the proposed safety-critical one-step MPC (23) or the modified version of [3] as the controller are shown in Figs. 2, 3, and 4[1]. As can be seen in Fig. 4, both types of controllers were able to guarantee safety regarding the RD2 constraint which represents collision avoidance. The same conclusion can be made from Fig. 2, where the distance between the UAV and the obstacle was kept positive for both types of controllers.

However, notice that the modified version of [3] was unable to restrict the RD1 states V and γ between their maximum and minimum allowable values, which are represented as black dotted lines in Fig. 2. In contrast, the proposed one-step MPC successfully bounded V and γ to remain inside the corresponding RD1 safe sets. Moreover, the proposed controller was able to find a safe and feasible control input even when the UAV was simultaneously on the boundary

[1] Implementation details including the parameteres and codes can be found at https://github.com/DonggeonDavidOh/LARR2023.git