# Collision Avoidance and Geofencing for Fixed-Wing Aircraft With Control Barrier Functions

Tamas G. Molnar, *Member, IEEE*, Suresh K. Kannan, James Cunningham, Kyle Dunlap, Kerianne L. Hobbs, and Aaron D. Ames, *Fellow, IEEE*

***Abstract*— Safety-critical failures often have fatal consequences in aerospace control. Control systems on aircraft, therefore, must ensure the strict satisfaction of safety constraints, preferably with formal guarantees of safe behavior. This article establishes the safety-critical control of fixed-wing aircraft in collision avoidance and geofencing tasks. A control framework is developed wherein a run-time assurance (RTA) system modulates the nominal flight controller of the aircraft whenever necessary to prevent it from colliding with other aircraft or crossing a boundary (geofence) in space. The RTA is formulated as a safety filter using control barrier functions (CBFs) with formal guarantees of safe behavior. CBFs are constructed and compared for a nonlinear kinematic fixed-wing aircraft model. The proposed CBF-based controllers showcase the capability of safely executing simultaneous collision avoidance and geofencing, as demonstrated by simulations on the kinematic model and a high-fidelity dynamical model.**

***Index Terms*— Aerospace control, aircraft navigation, collision avoidance, control barrier function (CBF), geofencing.**

## I. Introduction

SAFE behavior is of utmost importance for aerial vehicles due to the severity of consequences in the case of an incidental failure. Thus, on-board control systems must satisfy strict safety constraints while operating aircraft during flight. Safety constraints may span a wide range of criteria, including bounds on the aircraft's flight envelope and control surfaces (envelope protection), altitude, speed, acceleration, load factors, orientation, angular rates, angle of attack, control surface deflections, and their rates, as well as bounds on an aircraft's trajectory (safe navigation) that may include geofences (keep-in or keep-out zones) and collision avoidance with the ground or other air vehicles. Such strict safety specifications necessitate control systems that are designed in a safety-critical fashion, preferably with guarantees or certificates of safe behavior under certain operating conditions. Recently, this has led to the idea of run-time assurance (RTA) systems [1] that serve as an added module to the aircraft's nominal flight controller and intervene whenever necessary to avoid the violation of safety. RTA is often used to supervise complex control systems whose safe behavior is difficult to guarantee or verify, such as learning-based controllers [2], [3]. RTA systems showed promising results in indoor flight with quadrotors [4], navigation of unmanned fixed-wing aircraft along safe corridors [5], neural network-based aircraft taxiing application [6], and aerial refueling task in naval aviation [7]. How to design safe control laws for use in RTA, however, is still an open problem.

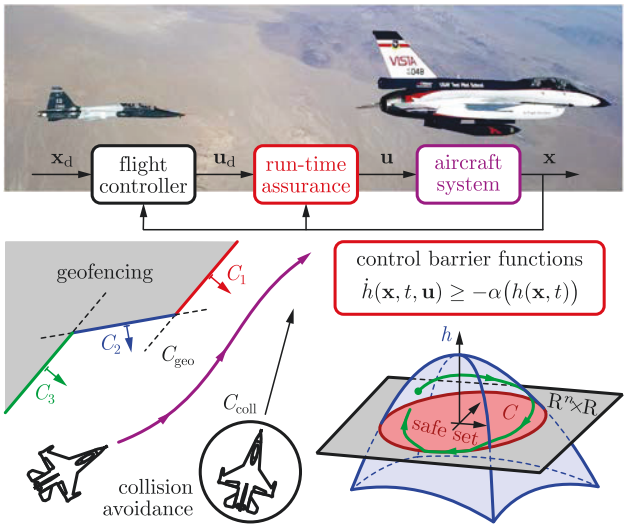


Fig. 1. Overview of the safety-critical control framework for fixed-wing aircraft. The motion of a single aircraft is controlled to avoid collisions with other aircraft and prevent crossing a geofence in 3-D space. To this end, the proposed CBF-based RTA system intervenes into the nominal flight controller whenever necessary to avoid dangers.

Safety has long been of interest in the aerospace control literature. Many studies have focused on safe aircraft

Received 18 July 2024; revised 17 November 2024; accepted 26 January 2025. Approved for public release: distribution is unlimited. Case Number AFRL-2024-0947 and AFRL-2024-2450. The views expressed are those of the authors and do not reflect the official guidance or position of the United States Government, the Department of Defense, or the United States Air Force. This work was supported in part by the National Science Foundation (CPS Award) under Grant 1932091; in part by AFOSR; and in part by Nodein Inc., through USAF Award, under Grant FA864922P0787. Recommended by Associate Editor I. Prodan. *(Corresponding author: Tamas G. Molnar.)*

Tamas G. Molnar is with the Department of Mechanical Engineering, Wichita State University, Wichita, KS 67260 USA (e-mail: tamas.molnar@wichita.edu).

Suresh K. Kannan is with Nodein Autonomy Corporation, Farmington, CT 06085 USA (e-mail: kannan@nodein.com).

James Cunningham and Kyle Dunlap are with Parallax Advanced Research, Beavercreek, OH 45431 USA (e-mail: james.cunningham@parallaxresearch.org; kyle.dunlap@parallaxresearch.org).

Kerianne L. Hobbs is with the Autonomy Capability Team (ACT3), Air Force Research Laboratory, Wright-Patterson AFB, OH 45433 USA (e-mail: kerianne.hobbs@us.af.mil).

Aaron D. Ames is with the Department of Mechanical and Civil Engineering, California Institute of Technology, Pasadena, CA 91125 USA (e-mail: ames@caltech.edu).

Digital Object Identifier 10.1109/TCST.2025.3536215

navigation, where the overall motion is controlled in scenarios like collision avoidance, trajectory tracking, and geofencing. Approaches include the use of potential fields [8], [9], velocity obstacles [10], control barrier functions (CBFs) [11], controlled invariant sets [12], and reinforcement learning [13], [14]. Notably, CBFs provide provable safety guarantees as opposed to most learning-based methods. Furthermore, some CBF formulations can be used to generate control invariant sets [15], while CBFs can even be constructed from artificial potential fields [16] and velocity obstacles [17]. Most CBF-based controllers respond to the current state of the system without a preview of future trajectories. This may lead to suboptimal behavior compared to other methods with preview, such as motion planning algorithms or model predictive control, but it also makes CBF-based controllers easy to compute.

CBFs demonstrated success in collision avoidance on quadrotors in simulation [18], [19] and experiments [16], as well as on fixed-wing aircraft in the context of probabilistic safety certificates [20], learning-based [21] and data-driven CBFs [22], and multiaircraft control [23]. Trajectory tracking by fixed-wing aircraft also leveraged CBF theory to keep path following errors within prescribed bounds, via high-order CBFs [24], barrier Lyapunov functions [25], and robust CBFs [26]. Moreover, geofencing—where the aircraft is navigated to avoid a restricted territory beyond a "geofence" like the airspace of a country or private property—was also addressed by CBFs with success on quadrotors [27], [28]. A CBF-based fixed-wing aircraft RTA system for simultaneous collision avoidance and geofencing is yet to be developed.

Notably, Corraro et al. [29] addressed simultaneous collision avoidance and geofencing without CBFs, for tactical unmanned aircraft considering multiple intruder aircraft, fixed obstacles, no-fly zones, and bad weather areas. An optimal control problem was formulated with risk assessment and probabilistic safety constraints. In this work, we instead focus on deterministic safety constraints and develop a CBF-based RTA system for simultaneous collision avoidance and geofencing. We use a reduced-order model of the aircraft dynamics to formulate the CBF and synthesize a controller in closed form that provides formal guarantees of safety. The approach of developing safety-critical controllers via reduced-order models and CBFs has been successful on a variety of autonomous systems, including legged, wheeled, and flying robots, manipulators, and heavy-duty trucks [30]. Now we seek to extend this approach to establish RTA for fixed-wing aircraft.

In this article, we propose a safety-critical RTA system for fixed-wing aircraft using CBFs, as summarized in Fig. 1. Specifically, we accomplish simultaneous collision avoidance and geofencing in 3-D space on a nonlinear kinematic model of an aircraft. To the best of authors' knowledge, this is the first application of CBFs to simultaneous collision avoidance and geofencing for fixed-wing aircraft. First, we encode multiple safety constraints (related to collision avoidance and multiple geofence boundaries) into a single CBF candidate. Second, we establish how to construct CBFs for nonlinear aircraft dynamics. We introduce and compare three approaches: high-order CBFs, backstepping-based CBFs, and model-free CBFs. We highlight that a careful CBF construction is required to leverage all control inputs and make the aircraft both accelerate, pitch, and roll for safe collision avoidance and geofencing. This challenge can be addressed, for example, via backstepping, and this work is the first application of CBF backstepping to fixed-wing aircraft. Finally, we demonstrate the behavior of the proposed RTA by simulations of the kinematic model and also a high-fidelity dynamical model.

This article is structured as follows. Section II describes the nonlinear model of fixed-wing aircraft kinematics. Section III provides background on CBFs. Section IV introduces the main contributions by establishing and simulating the proposed CBF-based RTA system. Section V closes with conclusions.

## II. Modeling the Aircraft's Motion

First, we introduce a kinematic model, called *3-D Dubins model*, that describes the motion of fixed-wing aircraft. Kinematic models are popular in navigation and path planning [31], [32]. This model is derived in Appendix A from six-degrees-of-freedom (6-DoF) dynamics using certain assumptions (see Assumption 3). The main assumption is that the angles of attack and sideslip are negligible. We use this model for safety-critical controller synthesis, to design acceleration and angular velocity commands for the aircraft in a provably safe fashion.

Consider the fixed-wing aircraft in Fig. 2. We describe its motion by using its: position along North, East, and down directions, $n$, $e$, and $d$; roll, pitch, and yaw Euler angles, $\phi$, $\theta$, and $\psi$; speed, $V_{\mathrm{T}}$; angular velocities about the front, right, and down axes, $P$, $Q$, and $R$; and longitudinal acceleration, $A_{\mathrm{T}}$. Specifically, we use the *3-D Dubins model* detailed in Appendix A as governing equation

$$\begin{aligned}
\dot{n} &= V_{\mathrm{T}} c_\psi c_\theta \\
\dot{e} &= V_{\mathrm{T}} s_\psi c_\theta \\
\dot{d} &= -V_{\mathrm{T}} s_\theta \\
\dot{\phi} &= P + s_\phi t_\theta Q + c_\phi t_\theta R \\
\dot{\theta} &= c_\phi Q - s_\phi R \\
\dot{\psi} &= \frac{s_\phi}{c_\theta} Q + \frac{c_\phi}{c_\theta} R \\
\dot{V}_{\mathrm{T}} &= A_{\mathrm{T}}
\end{aligned} \tag{1}$$

where $s_{(\cdot)}$, $c_{(\cdot)}$, and $t_{(\cdot)}$ stand for $\sin(\cdot)$, $\cos(\cdot)$, $\tan(\cdot)$, respectively, and

$$R = \frac{g_D}{V_{\mathrm{T}}} s_\phi c_\theta \tag{2}$$

with $g_D$ being the gravitational acceleration.

The top three rows of (1) give the velocity of the center of mass. The next three rows of (1) are the typical 3–2–1 Euler angle kinematics [33] that relate orientation to angular velocity. The last row of (1) associates speed with acceleration. Note that (1) is true for the motion of any rigid body in an inertial frame. Meanwhile, (2) makes this an aircraft model by stating that the airplane must roll in order to turn left or right ($R \neq 0$ requires $\phi \neq 0$). The model has seven states (three positions, three orientation angles, and speed), while we consider three control inputs (acceleration and angular

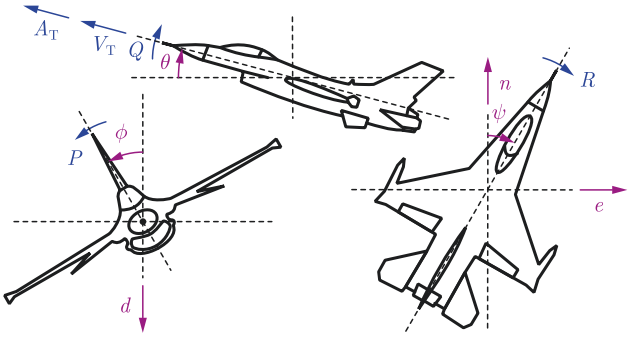


Fig. 2. Illustration of aircraft kinematics: position coordinates $n$, $e$, and $d$, Euler angles $\phi$, $\theta$, and $\psi$, speed $V_T$, angular velocities $P$, $Q$, and $R$, and acceleration $A_T$.

velocities related to rolling and pitching). The state $\mathbf{x}$ and input $\mathbf{u}$ read

$$\mathbf{x} = \begin{bmatrix} n & e & d & \phi & \theta & \psi & V_T \end{bmatrix}^\top$$
$$\mathbf{u} = \begin{bmatrix} A_T & P & Q \end{bmatrix}^\top. \tag{3}$$

To compress notations, we introduce the position vector $\mathbf{r}$, the velocity-related states $\boldsymbol{\zeta}$, and the velocity vector $\mathbf{v}$

$$\mathbf{r} = \begin{bmatrix} n \\ e \\ d \end{bmatrix}, \quad \boldsymbol{\zeta} = \begin{bmatrix} V_T \\ \theta \\ \psi \end{bmatrix}, \quad \mathbf{v}(\boldsymbol{\zeta}) = \begin{bmatrix} V_T c_\theta c_\psi \\ V_T c_\theta s_\psi \\ -V_T s_\theta \end{bmatrix}. \tag{4}$$

With these, the dynamics (1) and (2) become

$$\begin{aligned} \dot{\mathbf{r}} &= \mathbf{v}(\boldsymbol{\zeta}) \\ \dot{\boldsymbol{\zeta}} &= \mathbf{f}_\zeta(\boldsymbol{\zeta}, \phi, R, A_T, Q), \quad R = \rho(\boldsymbol{\zeta}, \phi) \\ \dot{\phi} &= f_\phi(\boldsymbol{\zeta}, \phi, R, Q, P). \end{aligned} \tag{5}$$

These dynamics form a cascaded structure, where the evolution of higher level states depends on lower level states. Specifically, the evolution of the position $\mathbf{r}$ is given by the velocity-related states $\boldsymbol{\zeta}$. The evolution of $\boldsymbol{\zeta}$ depends on the remaining state $\phi$, the state-dependent turning rate $R$, and two control inputs $A_T$ and $Q$. Finally, the evolution of the last state $\phi$ involves the third control input $P$. Overall, the dynamics have a three-level cascaded structure, where the control inputs enter at the second and third levels (with an auxiliary algebraic formula for $R$). Importantly, the right-hand side expressions

$$\mathbf{f}_\zeta(\boldsymbol{\zeta}, \phi, R, A_T, Q) = \begin{bmatrix} A_T \\ c_\phi Q - s_\phi R \\ \dfrac{s_\phi}{c_\theta} Q + \dfrac{c_\phi}{c_\theta} R \end{bmatrix}, \quad \rho(\boldsymbol{\zeta}, \phi) = \frac{g_D}{V_T} s_\phi c_\theta$$
$$f_\phi(\boldsymbol{\zeta}, \phi, R, Q, P) = P + s_\phi t_\theta Q + c_\phi t_\theta R \tag{6}$$

are affine in the turning rate $R$ and the control inputs $A_T$, $P$, and $Q$. We will leverage this structure when designing controllers.

Furthermore, we also write (1) and (2) into the compact form

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u} \tag{7}$$

where $\mathbf{x}$ and $\mathbf{u}$ are given in (3) and

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} \mathbf{v}(\boldsymbol{\zeta}) \\ \mathbf{f}_\xi(\mathbf{x}) \\ 0 \end{bmatrix}, \quad \mathbf{g}(\mathbf{x}) = \begin{bmatrix} \mathbf{0}_{3\times 3} \\ \mathbf{g}_\xi(\mathbf{x}) \\ 1 \quad 0 \quad 0 \end{bmatrix} \tag{8}$$

with

$$\mathbf{f}_\xi(\mathbf{x}) = \frac{g_D}{V_T} \begin{bmatrix} s_\phi c_\phi s_\theta \\ -s_\phi^2 c_\theta \\ s_\phi c_\phi \end{bmatrix}, \quad \mathbf{g}_\xi(\mathbf{x}) = \begin{bmatrix} 0 & 1 & s_\phi t_\theta \\ 0 & 0 & c_\phi \\ 0 & 0 & \dfrac{s_\phi}{c_\theta} \end{bmatrix}. \tag{9}$$

In what follows, we synthesize controllers for the aircraft model given by (1) and (2), with the end goal of formal safety guarantees with respect to collision avoidance and geofencing. In particular, we design safe controllers for the control-affine form (7) of the model by using CBFs constructed based on the cascaded structure (5) of the dynamics. When designing the controller, we consider that the position $\mathbf{r}$ of the aircraft is safe at time $t$ if the pair $(\mathbf{r}, t)$ is inside a safe set $\mathcal{S} \subset \mathbb{R}^3 \times \mathbb{R}_{\geq 0}$ defined by the collision avoidance and geofencing constraints (see details in Section IV).

*Problem Statement:* Consider system (7) and design a controller $\mathbf{k}{:}\mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^m$, $\mathbf{u} = \mathbf{k}(\mathbf{x}, t)$ that guarantees safety with respect to collision avoidance and geofencing. Specifically, the controller must ensure that the trajectory $\mathbf{r}(t)$ of the closed-loop system is maintained within a prescribed set $\mathcal{S}$: $(\mathbf{r}(t), t) \in \mathcal{S}$, $\forall t \geq 0$ for initial positions satisfying $(\mathbf{r}_0, 0) \in \mathcal{S}$.

## III. Control Barrier Functions

We utilize CBFs [34] to guarantee safety on fixed-wing aircraft with respect to collision avoidance and geofencing. In particular, we consider time-dependent CBFs because safety depends on the time when other aircraft move or the geofence is updated.

*Notations:* If $\alpha{:}\mathbb{R} \to \mathbb{R}$ is continuous and strictly increasing with $\alpha(0) = 0$ and $\lim_{r\to\pm\infty} \alpha(r) = \pm\infty$, then $\alpha$ is of extended class-$\mathcal{K}_\infty$ ($\alpha \in \mathcal{K}_\infty^e$). Norms are denoted by $\|\mathbf{u}\|^2 = \mathbf{u}^\top\mathbf{u}$ and $\|\mathbf{u}\|_\Gamma^2 = \mathbf{u}^\top\boldsymbol{\Gamma}\mathbf{u}$ for $\mathbf{u} \in \mathbb{R}^m$ and positive definite $\boldsymbol{\Gamma} \in \mathbb{R}^{m\times m}$. The zero vector in $\mathbb{R}^3$ is $\mathbf{0}$, the zero matrix in $\mathbb{R}^{m\times n}$ is $\mathbf{0}_{m\times n}$, and the identity matrix in $\mathbb{R}^{3\times 3}$ is $\mathbf{I}$.

### A. Theoretical Background

Motivated by the aircraft model (7), consider control systems with state $\mathbf{x} \in \mathbb{R}^n$ and control input $\mathbf{u} \in \mathbb{R}^m$

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u}. \tag{10}$$

Let $\mathbf{f}{:}\mathbb{R}^n \to \mathbb{R}^n$ and $\mathbf{g}{:}\mathbb{R}^n \to \mathbb{R}^{m\times n}$ be locally Lipschitz continuous. Given these dynamics, our goal is to design a locally Lipschitz continuous controller $\mathbf{k}{:}\mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^m$, $\mathbf{u} = \mathbf{k}(\mathbf{x}, t)$ such that the closed control loop

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{k}(\mathbf{x}, t) \tag{11}$$

exhibits safe behavior. We characterize the behavior by the solution $\mathbf{x}(t)$ of (11) with initial condition $\mathbf{x}(0) = \mathbf{x}_0 \in \mathbb{R}^n$, and we assume that $\mathbf{x}(t)$ exists for all $t \geq 0$.

We consider the system to be safe if the solution $\mathbf{x}(t)$ at any time $t$ is kept within a *safe set* $\mathcal{C} \subset \mathbb{R}^n \times \mathbb{R}$, stated as follows.

*Definition 1:* System (11) is called *safe with respect to* $\mathcal{C}$ if $(\mathbf{x}(t), t) \in \mathcal{C}$, $\forall t \geq 0$ holds for all $\mathbf{x}_0 \in \mathbb{R}^n$ that satisfies $(\mathbf{x}_0, 0) \in \mathcal{C}$. In other words, safety means that set $\mathcal{C}$ is forward invariant along (11).

Let $\mathcal{C}$ be given as the 0-superlevel set of a continuously differentiable function $h{:}\mathbb{R}^n \times \mathbb{R} \to \mathbb{R}$, with boundary $\partial\mathcal{C}$

$$\begin{aligned}\mathcal{C} &= \{(\mathbf{x}, t) \in \mathbb{R}^n \times \mathbb{R} : h(\mathbf{x}, t) \geq 0\} \\ \partial\mathcal{C} &= \{(\mathbf{x}, t) \in \mathbb{R}^n \times \mathbb{R} : h(\mathbf{x}, t) = 0\}.\end{aligned} \tag{12}$$

Then, we characterize safety using the scalar-valued function $h$, whose positive (or negative) sign indicates safe (or unsafe) behavior. In the context of aircraft collision avoidance or geofencing, $h$ is related to the signed distance of the aircraft from other aircraft or the geofence, respectively, that must kept positive for safety. The expression of $h$ will be detailed in the following. Throughout this article, we assume that the underlying set $\mathcal{C}$ is nonempty, it has no isolated points, and that $\partial h/\partial \mathbf{x}(\mathbf{x}, t)$ is nonzero for all $(\mathbf{x}, t) \in \partial\mathcal{C}$ (i.e., 0 is a regular value of $h$).

Given set $\mathcal{C}$ and function $h$, the theory of CBFs establishes a method to synthesize controllers with formal safety guarantees. CBFs are defined as follows, by using the derivative of $h$ along system (10) that quantifies the effect of the input $\mathbf{u}$ on safety

$$\dot{h}(\mathbf{x}, t, \mathbf{u}) = \frac{\partial h}{\partial t}(\mathbf{x}, t) + \frac{\partial h}{\partial \mathbf{x}}(\mathbf{x}, t)\mathbf{f}(\mathbf{x}) + \frac{\partial h}{\partial \mathbf{x}}(\mathbf{x}, t)\mathbf{g}(\mathbf{x})\mathbf{u}. \tag{13}$$

*Definition 2 [11]:* Function $h$ is a *CBF* for (10) on $\mathcal{C}$ if there exists $\alpha \in \mathcal{K}_\infty^e$ such that for all $(\mathbf{x}, t) \in \mathcal{C}$

$$\sup_{\mathbf{u}\in\mathbb{R}^m} \dot{h}(\mathbf{x}, t, \mathbf{u}) > -\alpha(h(\mathbf{x}, t)). \tag{14}$$

Given a CBF, the following theorem establishes formal safety guarantees for controllers.

*Lemma 1 [11]:* If $h$ is a CBF for (10) on $\mathcal{C}$, then any locally Lipschitz continuous controller $\mathbf{k}$ that satisfies

$$\dot{h}(\mathbf{x}, t, \mathbf{k}(\mathbf{x}, t)) \geq -\alpha(h(\mathbf{x}, t)) \tag{15}$$

for all $(\mathbf{x}, t) \in \mathcal{C}$ renders (11) safe with respect to $\mathcal{C}$.

Note that if the derivative of $h$ is independent of the control input $\mathbf{u}$, i.e., $\partial h/\partial \mathbf{x}(\mathbf{x}, t)\mathbf{g}(\mathbf{x}) \equiv \mathbf{0}$ in (13), then $h$ cannot be used directly to synthesize safe controllers. In such cases, methods like high-order CBFs [35], [36] or backstepping [37] can be applied to construct a CBF from $h$ and use it for control. These methods are discussed further in Section IV. The main idea behind high-order CBFs is introducing $\dot{h} + \alpha(h)$, which occurs in (15), as a high-order (or extended) CBF

$$h_e(\mathbf{x}, t) = \frac{\partial h}{\partial t}(\mathbf{x}, t) + \frac{\partial h}{\partial \mathbf{x}}(\mathbf{x}, t)\mathbf{f}(\mathbf{x}) + \alpha(h(\mathbf{x}, t)) \tag{16}$$

whose 0-superlevel set is $\mathcal{C}_e$ see (12). If the extended CBF $h_e$ is maintained nonnegative, then safety can be guaranteed based on (15), similar to the case when $h$ is a CBF.

*Lemma 2 [36]:* If $h_e$ is a CBF for (10) on $\mathcal{C}_e$, then any locally Lipschitz continuous controller $\mathbf{k}$ that satisfies

$$\dot{h}_e(\mathbf{x}, t, \mathbf{k}(\mathbf{x}, t)) \geq -\alpha(h_e(\mathbf{x}, t)) \tag{17}$$

for all $(\mathbf{x}, t) \in \mathcal{C} \cap \mathcal{C}_e$ renders (11) safe with respect to $\mathcal{C} \cap \mathcal{C}_e$.

### B. Safety Filters

Based on Lemma 1, controllers can be designed for (10) in a safety-critical fashion by enforcing (15) during controller synthesis. Such safety-critical control designs are often established by so-called *safety filters* that modify a desired but not necessarily safe controller $\mathbf{k}_d{:}\mathbb{R}^n \times \mathbb{R} \to \mathbb{R}^m$ to a safe controller $\mathbf{k}$. In the context of aircraft control, the desired controller $\mathbf{k}_d$ may be a flight controller that tracks a desired trajectory, such as the controller in Appendix B. This can be modified for safety, for example, by the optimization problem

$$\begin{aligned}\mathbf{k}(\mathbf{x}, t) = \underset{\mathbf{u}\in\mathbb{R}^m}{\operatorname{argmin}}\ & \|\mathbf{u} - \mathbf{k}_d(\mathbf{x}, t)\|_{\mathbf{\Gamma}}^2 \\ \text{s.t. } & \dot{h}(\mathbf{x}, t, \mathbf{u}) \geq -\alpha(h(\mathbf{x}, t))\end{aligned} \tag{18}$$

that minimizes the difference of the desired and actual inputs subject to the safety constraint (15). Note that a symmetric positive definite weight matrix $\mathbf{\Gamma} \in \mathbb{R}^{m\times m}$ is useful for scaling the various components of the control input when these have different physical meanings and orders of magnitude.

We remark that the formulation above does not include input constraints (as $\mathbf{u} \in \mathbb{R}^m$). To enforce input limits, one could saturate the controller, or add input constraints to (18) and relax the safety constraint to maintain feasibility. However, these methods lose formal safety guarantees. Formally addressing input constraints while maintaining safety requires more advanced CBF approaches, such as those in [15], [38], [39], [40], and [41]. These methods are beyond the scope of this article and will be explored as future work. Instead, the controllers proposed here are tuned so that the resulting control inputs remain within reasonable bounds in simulations.

Importantly, the optimization problem (18) is feasible if $h$ is indeed a CBF that satisfies (14), and it can be solved in closed form. By factorizing $\mathbf{\Gamma}$ as $\mathbf{\Gamma} = \mathbf{W}^{-\top}\mathbf{W}^{-1}$ with positive definite $\mathbf{W} \in \mathbb{R}^{m\times m}$, (18) transforms into

$$\begin{aligned}\mathbf{k}(\mathbf{x}, t) &= \mathbf{k}_d(\mathbf{x}, t) + \mathbf{W}\hat{\mathbf{k}}(\mathbf{x}, t) \\ \hat{\mathbf{k}}(\mathbf{x}, t) &= \underset{\hat{\mathbf{u}}\in\mathbb{R}^m}{\operatorname{argmin}}\ \|\hat{\mathbf{u}}\|^2 \\ &\qquad \text{s.t. } a(\mathbf{x}, t) + \mathbf{b}(\mathbf{x}, t)\hat{\mathbf{u}} \geq 0\end{aligned} \tag{19}$$

where

$$\begin{aligned}a(\mathbf{x}, t) &= \dot{h}(\mathbf{x}, t, \mathbf{k}_d(\mathbf{x}, t)) + \alpha(h(\mathbf{x}, t)) \\ \mathbf{b}(\mathbf{x}, t) &= \frac{\partial h}{\partial \mathbf{x}}(\mathbf{x}, t)\mathbf{g}(\mathbf{x})\mathbf{W}.\end{aligned} \tag{20}$$

Then, (19) can be solved as follows [42]:

$$\mathbf{k}(\mathbf{x}, t) = \mathbf{k}_d(\mathbf{x}, t) + \Lambda(a(\mathbf{x}, t), \|\mathbf{b}(\mathbf{x}, t)\|)\mathbf{W}\mathbf{b}(\mathbf{x}, t)^\top \tag{21}$$

with

$$\Lambda(a, b) = \begin{cases} 0, & \text{if } b = 0 \\ \dfrac{1}{b}\max\left\{0, -\dfrac{a}{b}\right\}, & \text{if } b \neq 0. \end{cases} \tag{22}$$

Since the safety filter (21) is given explicitly, it is fast and easy to evaluate it, which is an advantage of using CBFs.

Finally, we remark that the controller given by (21) and (22) is continuous but not necessarily differentiable. Yet, differentiable controllers may be preferable in certain scenarios.

As established by Cohen et al. [42], a smooth over-approximation of $\Lambda$ in (22), such as

$$\Lambda(a,b)=\begin{cases}0, & \text{if } b=0\\ \dfrac{1}{\nu b}\ln\left(1+e^{-\nu\frac{a}{b}}\right), & \text{if } b\neq 0\end{cases} \tag{23}$$

with parameter $\nu>0$, leads to a smooth controller in (21) that still satisfies (15) (and it approaches (22) for $\nu\to\infty$). Henceforth, we will use the safety filter [see (21) and (22)] and the smooth safety filter [see (21)–(23)] for safety-critical control.

### C. Compositions of Safety Constraints

In practice, control systems may need to satisfy more than one safety constraints simultaneously. In such cases, the safe set $\mathcal{C}$ is composed of multiple sets $\mathcal{C}_i$ related to each safety constraint. For example, as illustrated in Fig. 1, an aircraft may execute both collision avoidance and geofencing at the same time, associated with sets $\mathcal{C}_{\text{coll}}$ and $\mathcal{C}_{\text{geo}}$. This leads to the overall safe set $\mathcal{C}=\mathcal{C}_{\text{coll}}\cap\mathcal{C}_{\text{geo}}$. With this as motivation, now we briefly revisit a method established in [43] to combine multiple safety constraints and express them by a single CBF.

First, consider the scenario when safety must be maintained against $N_{\text{c}}$ constraints simultaneously, associated with sets $\mathcal{C}_i$, functions $h_i$, and index $i\in I=\{1,\dots,N_{\text{c}}\}$. Safety is interpreted with respect to the intersection $\mathcal{C}=\cap_{i\in I}\mathcal{C}_i$ of the sets

$$\bigcap_{i\in I}\mathcal{C}_i=\left\{(\mathbf{x},t)\in\mathbb{R}^n\times\mathbb{R}:\min_{i\in I}h_i(\mathbf{x},t)\geq 0\right\}. \tag{24}$$

Then, a single CBF candidate can be constructed by a smooth approximation of the min function [43], [44]

$$h(\mathbf{x},t)=-\frac{1}{\kappa}\ln\left(\sum_{i\in I}e^{-\kappa h_i(\mathbf{x},t)}\right) \tag{25}$$

with a smoothing parameter $\kappa>0$, where the approximation error is bounded by

$$-\frac{\ln N_{\text{c}}}{\kappa}\leq h(\mathbf{x},t)-\min_{i\in I}h_i(\mathbf{x},t)\leq 0 \tag{26}$$

and diminishes as $\kappa\to\infty$. Formula (25) establishes a simple way to combine multiple barriers, ultimately providing a single CBF candidate for use in the safety filter (21), with derivative

$$\dot{h}(\mathbf{x},t,\mathbf{u})=\sum_{i=1}^{N_{\text{c}}}e^{-\kappa(h_i(\mathbf{x},t)-h(\mathbf{x},t))}\dot{h}_i(\mathbf{x},t,\mathbf{u}). \tag{27}$$

Formula (25) includes the evaluation of $N_{\text{c}}$ exponential functions; hence, its computation time increases with the number $N_{\text{c}}$ of constraints. Yet, a significant benefit of constructing a single CBF by (25) is that it enables the use of closed-form safety filters like (21) that are easy to compute. Without such composition, safety filters would require solving optimization problems like (18) with $N_{\text{c}}$ the number of constraints, which would ultimately lead to more computation especially if $N_{\text{c}}$ is large. Further details on the properties of the CBF composition method in (25) can be found in [43].

The framework in [43] can also be used for more complex compositions of safe sets, as combinations of set intersections and unions, which correspond to AND and OR logic between safety constraints, respectively. For example, the geofence in Fig. 1 can also be expressed by unions and intersections: $\mathcal{C}_{\text{geo}}=\mathcal{C}_1\cup\mathcal{C}_2\cap\mathcal{C}_3$. As such, the union of individual sets

$$\bigcup_{i\in I}\mathcal{C}_i=\left\{(\mathbf{x},t)\in\mathbb{R}^n\times\mathbb{R}:\max_{i\in I}h_i(\mathbf{x},t)\geq 0\right\} \tag{28}$$

can be captured by a single CBF candidate analogously to (25)

$$h(\mathbf{x},t)=\frac{1}{\kappa}\ln\left(\sum_{i\in I}e^{\kappa h_i(\mathbf{x},t)}\right). \tag{29}$$

Combinations of set unions and intersections can be addressed by the recursive applications of (25) and (29). Note that (14) must hold for $h$ to be a valid CBF. This requires a control-sharing property [45], [46] that the individual barriers $h_i$ are compatible and do not lead to contradicting constraints for the control input. Establishing the conditions for obtaining a valid CBF after a general composition is still an open problem, and initial results can be found in [43]. Overall, (25) and (29) provide a systematic way of encoding complex compositions of safety constraints into a single CBF candidate. This will be leveraged below in the context of simultaneous collision avoidance and geofencing where the geofence is composed of multiple boundaries.

## IV. RTA ON FIXED-WING AIRCRAFT

Now we present our main contributions, wherein we use CBFs to formally guarantee safety on fixed-wing aircraft in collision avoidance and geofencing tasks. We establish safety-critical controllers for the system (7) by utilizing the safety filter (21) as RTA system with an appropriate choice of CBF. We address the nontrivial problem of CBF synthesis for aircraft dynamics, discuss various choices of CBFs, and demonstrate the performance of the resulting controllers by numerical simulations.

### A. Position-Based CBF Candidates

We seek to execute simultaneous collision avoidance and geofencing on the aircraft in a provably safe fashion via CBFs. Both collision avoidance and geofencing correspond to safety constraints on the position $\mathbf{r}$ of the aircraft. Accordingly, we have a *safe set* $\mathcal{S}$ defined over the position space and time

$$\mathcal{S}=\left\{(\mathbf{r},t)\in\mathbb{R}^3\times\mathbb{R}_{\geq 0}:h_{\text{p}}(\mathbf{r},t)\geq 0\right\} \tag{30}$$

where $h_{\text{p}}$ is a position-based CBF candidate described in the following. We seek to ensure $(\mathbf{r}(t),t)\in\mathcal{S}$, $\forall t\geq 0$ for all $(\mathbf{r}_0,0)\in\mathcal{S}$. Note that while we focus on position constraints only, attitude constraints could be addressed in a similar fashion by constructing CBFs that depend on the Euler angles.

*1) Collision Avoidance:* To avoid collisions, we enforce that the signed distance between the controlled aircraft and another aircraft is nonnegative for all time. That is, we require $h_{\text{p},i}(\mathbf{r}(t),t)\geq 0$, $\forall t\geq 0$, with

$$h_{\text{p},i}(\mathbf{r},t)=\|\mathbf{r}-\mathbf{r}_i(t)\|-\rho_i \tag{31}$$

where $\mathbf{r}_i(t)$ is the other aircraft's position, and $\rho_i>0$ is a collision radius containing the combined size of both aircraft

and an additional buffer distance if desired. If there are multiple other aircraft, we get multiple collision constraints with index $i$. The corresponding derivatives are

$$\dot{h}_{\mathrm{p},i}(\mathbf{r},t,\mathbf{v}) = \mathbf{n}_i(\mathbf{r},t)^\top(\mathbf{v}-\mathbf{v}_i(t)) \tag{32}$$

with $\dot{\mathbf{r}}_i(t) = \mathbf{v}_i(t)$ and

$$\mathbf{n}_i(\mathbf{r},t) = \frac{\mathbf{r}-\mathbf{r}_i(t)}{\|\mathbf{r}-\mathbf{r}_i(t)\|}. \tag{33}$$

*2) Geofencing:* For geofencing, we keep the aircraft on one side of a "fence," modeled as a plane at $\mathbf{r}_i$ with unit normal vector $\mathbf{n}_i$. That is, we require $h_{\mathrm{p},i}(\mathbf{r}(t),t) \geq 0$, $\forall t \geq 0$, where

$$h_{\mathrm{p},i}(\mathbf{r},t) = \mathbf{n}_i^\top(\mathbf{r}-\mathbf{r}_i) - \rho_i \tag{34}$$

and $\rho_i \geq 0$ is the distance to be kept from the geofence. Again, index $i$ indicates the possibility of multiple geofence constraints, that is, geofences with more complex geometry. The derivatives of the CBF candidates are

$$\dot{h}_{\mathrm{p},i}(\mathbf{r},t,\mathbf{v}) = \mathbf{n}_i^\top\mathbf{v}. \tag{35}$$

Note that zero is a regular value of $h_{\mathrm{p},i}$ for both (31) and (34), i.e., the gradient of $h_{\mathrm{p},i}$ is nonzero when $h_{\mathrm{p},i}(\mathbf{r},t) = 0$.

*3) Simultaneous Collision Avoidance and Geofencing:* Given $N_{\mathrm{c}}$ number of safety constraints, these can be combined into a single CBF candidate $h_{\mathrm{p}}(\mathbf{r}(t),t)$ based on (25) and (29). For simplicity, we make the following assumption.

*Assumption 1:* Safety constraints are linked with AND logic: one must avoid collision with aircraft 1 AND aircraft 2, etc., AND obey geofence 1 AND geofence 2, and so on.

This assumption is made only to keep the exposition simple since AND logic only requires (25) for CBF composition. Combination of AND and OR logic could also be handled straightforwardly via combining (25) with (29). Considering Assumption 1 and using (25) leads to the CBF candidate

$$h_{\mathrm{p}}(\mathbf{r},t) = -\frac{1}{\kappa}\ln\left(\sum_{i=1}^{N_{\mathrm{c}}} e^{-\kappa h_{\mathrm{p},i}(\mathbf{r},t)}\right) \tag{36}$$

with $\kappa > 0$. Note that $h_{\mathrm{p}}(\mathbf{r},t) = h_{\mathrm{p},i}(\mathbf{r},t)$ if $N_{\mathrm{c}} = 1$. We remark that (36) approximates a convex polytope for geofencing when $h_{\mathrm{p},i}$ are given by (34). Nonconvex regions could be obtained as the union of convex polytopes, i.e., by using (29).

Ideally, we would use the position-based CBF candidate $h_{\mathrm{p}}$ directly as CBF: $h(\mathbf{x},t) = h_{\mathrm{p}}(\mathbf{r},t)$. However, the corresponding safety constraint in (15) with $\alpha_{\mathrm{p}} \in \mathcal{K}_\infty^e$

$$\dot{h}_{\mathrm{p}}(\mathbf{r},t,\mathbf{v}(\boldsymbol{\zeta})) \geq -\alpha_{\mathrm{p}}\big(h_{\mathrm{p}}(\mathbf{r},t)\big) \tag{37}$$

is independent of the controller and may not always hold. As a result, (14) in Definition 2 is not satisfied, $h_{\mathrm{p}}$ is not a CBF, and $h_{\mathrm{p}}$ cannot be used directly to synthesize safe controllers for (7). We rather call $h_{\mathrm{p}}$ as *CBF candidate*, and in the following, we use it to construct a CBF $h$ for controller synthesis.

TABLE I
PARAMETER VALUES FOR NUMERICAL SIMULATIONS (WITH SI UNITS)

| | | | |
|---|---|---|---|
| model | $g_{\mathrm{D}} = 9.81$ | | |
| safety filter | $\alpha(r) = \gamma r$ | $\gamma = 0.1$ | $\mathbf{W} = \begin{bmatrix} 6 & 0 & 0 \\ 0 & 0.6 & 0 \\ 0 & 0 & 0.1 \end{bmatrix}$ |
| coll. avoid | $\mathbf{r}_1(t) = \mathbf{c} + \mathbf{v}_1 t$ | $\mathbf{c} = \begin{bmatrix} -3048 \\ 0 \\ 0 \end{bmatrix}$ | $\mathbf{v}_1 = \begin{bmatrix} 121.92 \\ 161.32 \\ 0 \end{bmatrix}$ |
| geofence | $\mathbf{r}_{2,3} = \begin{bmatrix} 0 \\ 11901 \\ 0 \end{bmatrix}$ | $\mathbf{n}_2 = \frac{1}{\sqrt{17}}\begin{bmatrix} -4 \\ -1 \\ 0 \end{bmatrix}$ | $\mathbf{n}_3 = \frac{1}{\sqrt{5}}\begin{bmatrix} -2 \\ -1 \\ 0 \end{bmatrix}$ |
| pos. CBF | $\rho_1 = 30$ | $\rho_{2,3} = 15$ | $\kappa = 0.007$ |
| ext. CBF | $\gamma_{\mathrm{p}} = 0.1$ | | |
| backstepping CBF | $\alpha_{\mathrm{e}}(r) = \gamma_{\mathrm{e}} r$ | $\gamma_{\mathrm{e}} = 0.1$ | $\mathbf{W}_{\mathrm{e}} = \mathbf{I}$ |
| | $\nu_{\mathrm{e}} = 1$ | $\mu_{\mathrm{e}} = 10^{-4}$ | |
| model-free RTA | $\alpha_{\mathrm{p}}(r) = \gamma_{\mathrm{p}} r$ | $\gamma_{\mathrm{p}} = 0.1$ | $\sigma = 3$ |
| | $\Gamma_{\mathbf{v}} = 4$ | $\nu_{\mathbf{v}} = 0.007$ | |
| traj. tracking | $\mathbf{r}_{\mathrm{g}}(t) = \mathbf{v}_{\mathrm{g}} t$ | $\mathbf{v}_{\mathrm{g}} = \begin{bmatrix} 0 \\ 161.32 \\ 0 \end{bmatrix}$ | $\mathbf{K}_{\mathbf{r}} = 0.05\,\mathbf{I}$ |
| | $\mathbf{K}_{\mathbf{v}} = 0.3\,\mathbf{I}$ | $\mu = 10^{-5}$ | $\lambda = 0.2$ |

### B. RTA With Velocity-Based Extended CBF

First, we employ so-called *extended CBFs* (or high-order CBFs) proposed in [35] and [36]. Extended CBFs, defined as follows, depend on the velocity $\mathbf{v}$:

$$h_{\mathrm{e},i}(\mathbf{r},\mathbf{v},t) = h_{\mathrm{p},i}(\mathbf{r},t) + \frac{1}{\gamma_{\mathrm{p}}}\dot{h}_{\mathrm{p},i}(\mathbf{r},t,\mathbf{v}) \tag{38}$$

with $\gamma_{\mathrm{p}} > 0$. For collision avoidance, the extended CBF is

$$h_{\mathrm{e},i}(\mathbf{r},\mathbf{v},t) = \|\mathbf{r}-\mathbf{r}_i(t)\| - \rho_i + \frac{1}{\gamma_{\mathrm{p}}}\mathbf{n}_i(\mathbf{r},t)^\top(\mathbf{v}-\mathbf{v}_i(t)) \tag{39}$$

see (31). For geofencing, the extended CBF becomes

$$h_{\mathrm{e},i}(\mathbf{r},\mathbf{v},t) = \mathbf{n}_i^\top(\mathbf{r}-\mathbf{r}_i) - \rho_i + \frac{1}{\gamma_{\mathrm{p}}}\mathbf{n}_i^\top\mathbf{v} \tag{40}$$

see (34). It can be shown that zero is a regular value of $h_{\mathrm{e},i}$ in (39) and (40), i.e., the gradient of $h_{\mathrm{e},i}$ is nonzero when $h_{\mathrm{e},i}(\mathbf{r},\mathbf{v},t) = 0$. For simultaneous collision avoidance and geofencing, the extended CBFs can be combined as in (25)

$$h_{\mathrm{e}}(\mathbf{r},\mathbf{v},t) = -\frac{1}{\kappa}\ln\left(\sum_{i=1}^{N_{\mathrm{c}}} e^{-\kappa h_{\mathrm{e},i}(\mathbf{r},\mathbf{v},t)}\right). \tag{41}$$

With these definitions, we use the extended CBFs directly

$$h(\mathbf{x},t) = h_{\mathrm{e}}(\mathbf{r},\mathbf{v}(\boldsymbol{\zeta}),t). \tag{42}$$

Importantly, enforcing $h(\mathbf{x},t) \geq 0$ implies that (37) holds with $\alpha_{\mathrm{p}}(r) = \gamma_{\mathrm{p}} r$. Therefore, based on applying Lemma 2, the safety filter (21) guarantees $h(\mathbf{x}(t),t) \geq 0$, $t \geq 0$ and $h_{\mathrm{p}}(\mathbf{r}(t),t) \geq 0$, $\forall t \geq 0$ (if $h(\mathbf{x}_0,0) \geq 0$ and $h_{\mathrm{p}}(\mathbf{r}_0,0) \geq 0$ hold). This means guaranteed safety in collision avoidance and geofencing. Details and proof of the safety guarantees provided by extended CBFs are in [36].

The performance of the proposed safety-critical controller is demonstrated by simulating the aircraft model (7) with the safety filter [see (21) and (22)] and the extended CBF [see (38)–(42)]. The simulation parameters used

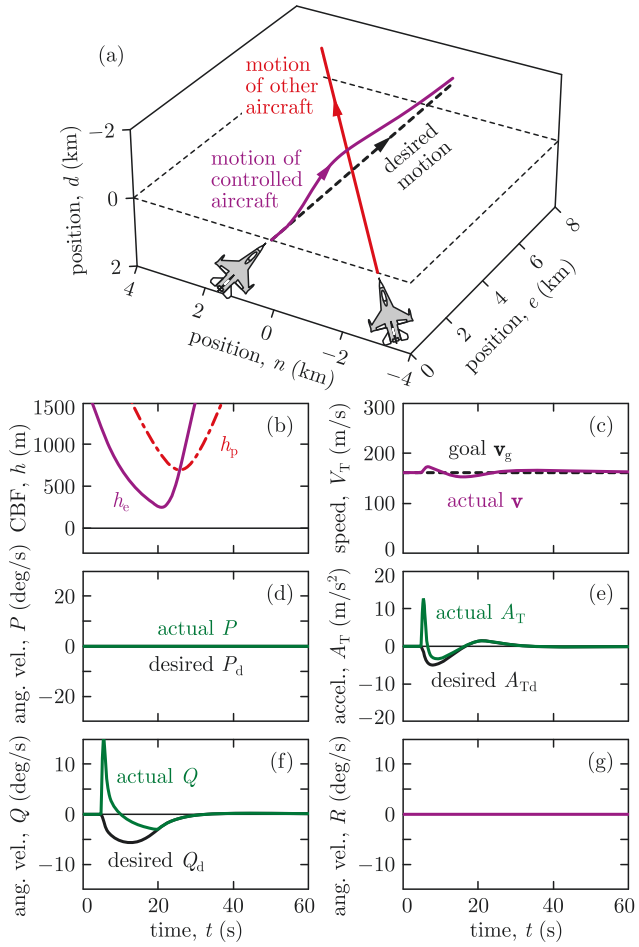


Fig. 3. Simulation of RTA in collision avoidance using extended CBF. The aircraft successfully maintains safety. (a) Trajectory. (b) CBF. (c)–(g) Relevant states and inputs.

throughout this article are listed in Table I, and the desired controller $\mathbf{k}_d$ is considered to be the trajectory tracking controller detailed in Appendix B. Note that safety guarantees do not depend on the choice of desired controller, and other flight controllers could also be used in RTA.

Fig. 3 illustrates a collision avoidance scenario. First, the controlled aircraft is tracking a straight-line trajectory at a constant speed with its desired controller $\mathbf{k}_d$. Then, another aircraft approaches from the right, which would result in a collision if the controlled aircraft did not respond. Thus, the safety filter intervenes and the controller $\mathbf{k}$ starts to deviate from the desired one $\mathbf{k}_d$. As a result, the controlled aircraft accelerates while pitching and moving up to safely avoid the other aircraft. Then, the safety filter stops intervening, and the aircraft starts to use its desired controller $\mathbf{k}_d$ to track its original course once again. Remarkably, the collision avoidance maneuver is generated automatically by the CBF.

Observe in Fig. 3(d)–(g), however, that the aircraft only leverages acceleration and pitching for collision avoidance, while it refrains from turning left or right via rolling and yawing. The safety filter only modifies two of the control inputs, $A_T$ and $Q$, that cause acceleration and pitching, while it does not affect the third input, $P$, that would induce rolling and consequently turning. This behavior is caused by the cascaded structure (5) of the dynamics and the construction of the extended CBF. Namely, the inputs $A_T$ and $Q$ enter the cascaded dynamics at the second level through the evolution of the velocity-related states $\boldsymbol{\zeta}$, while the input $P$ shows up at the third level in the equation of the roll angle $\phi$. Since the velocity-dependent extended CBF $h_e(\mathbf{r}, \mathbf{v}(\boldsymbol{\zeta}), t)$ includes the states $\boldsymbol{\zeta}$ but not the roll angle $\phi$, the resulting safety filter includes $A_T$ and $Q$ but not $P$. Hence, the safety filter cannot make the plane roll, which prevents it from turning left or right; see (2).

While the collision avoidance example was successful, lacking the ability to turn can be detrimental for RTA. Fig. 4 highlights this by showcasing a geofencing scenario with a vertical plane as geofence. Since the extended CBF $h_e$ cannot induce turning, the safety filter forces the aircraft to slow down and stop in front of the geofence. Although this behavior is safe from geofencing point of view, it is obviously infeasible in practice to make the aircraft stop. The geofencing task could only be accomplished with the ability to turn. This motivates us to construct a better CBF, which respects the cascaded structure of the dynamics and incorporates all states (including the roll angle $\phi$) so that the safety filter leverages all control inputs (including the roll motion $P$).

### C. RTA With Backstepping-Based CBF

While the extended CBF-based RTA does not leverage all possible behaviors (acceleration, pitching, rolling, and yawing), this can be done by other CBF choices [47]. In particular, the method of *backstepping* [37] offers a systematic procedure to synthesize valid CBFs for cascaded systems like (5). Here, we use backstepping to construct a CBF for use in RTA rather than construct the control law itself. Importantly, backstepping can provide a valid CBF whose derivative is affected by all control inputs; hence, the underlying RTA can leverage all aforementioned behaviors for safety—even considering 3-D motions and nontrivial cascaded dynamics.

We proceed with backstepping to design a CBF in two steps.

1) We apply the extended CBF to design a safe acceleration $\mathbf{a}_s$ and a corresponding safe angular velocity $R_s$ (related to turning). Note, however, that the aircraft cannot be commanded directly to turn, as $R$ is not an input.
2) We use the safe angular velocity $R_s$ to construct a CBF based on backstepping. This will allow us to synthesize the remaining input, the angular velocity $P$ (related to rolling), along with the other two inputs $A_T$ and $Q$.

To synthesize the safe angular velocity $R_s$, we first consider the extended CBF-based safety constraint

$$\dot{h}_e(\mathbf{r}, \mathbf{v}, t, \mathbf{a}) \geq -\alpha_e(h_e(\mathbf{r}, \mathbf{v}, t)) \tag{43}$$

and the corresponding smooth safety filter to obtain the safe acceleration $\mathbf{a}_s$ (associated with zero desired acceleration)

$$\begin{aligned} \mathbf{a}_s(\mathbf{x}, t) &= \Lambda(a_e(\mathbf{x}, t), \|\mathbf{b}_e(\mathbf{x}, t)\|)\mathbf{W}_e\mathbf{b}_e(\mathbf{x}, t)^\top \\ a_e(\mathbf{x}, t) &= \dot{h}_e(\mathbf{r}, \mathbf{v}(\boldsymbol{\zeta}), t, \mathbf{0}) + \alpha_e(h_e(\mathbf{r}, \mathbf{v}(\boldsymbol{\zeta}), t)) \\ \mathbf{b}_e(\mathbf{x}, t) &= \frac{\partial h_e}{\partial \mathbf{v}}(\mathbf{r}, \mathbf{v}(\boldsymbol{\zeta}), t)\mathbf{W}_e \end{aligned} \tag{44}$$

with $\Lambda$ in (23), a smoothing parameter $\nu_e > 0$, a weight $\mathbf{W}_e$ on the various acceleration components, and $\alpha_e \in \mathcal{K}_\infty^e$.

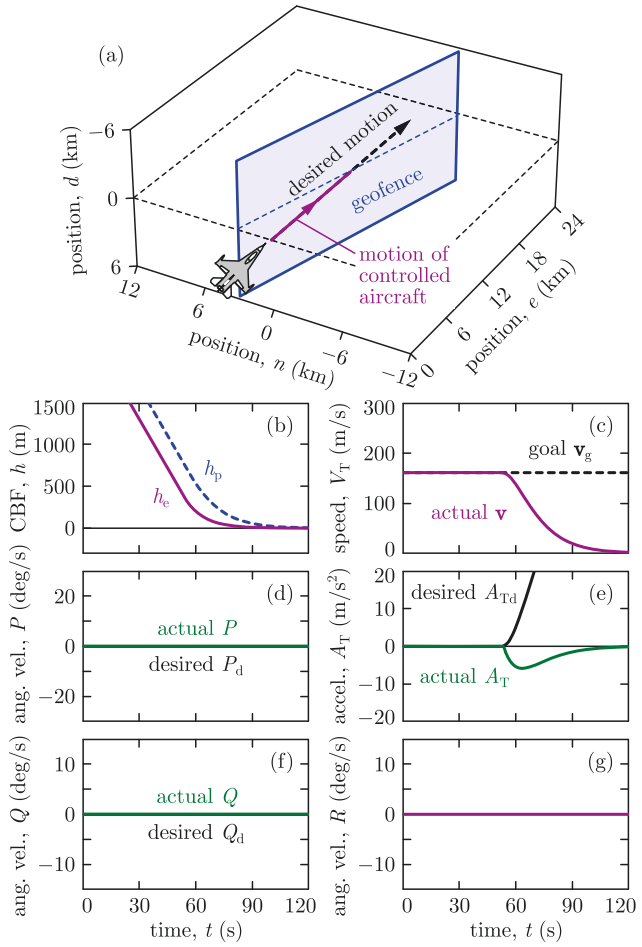


Fig. 4. Simulation of RTA in geofencing using extended CBF. The controller fails this task as it would command the aircraft to stop in front of the geofence. This CBF choice lacks the ability to make the aircraft turn left or right. (a) Trajectory. (b) CBF. (c)–(g) Relevant states and inputs.

We convert the safe acceleration $\mathbf{a}_s$ to the angular velocity $R_s$ based on (63) listed in Appendix A

$$R_s = \mathbf{W}_R(\boldsymbol{\zeta}, \phi)\mathbf{a}_s(\mathbf{x}, t) \tag{45}$$

where $\mathbf{W}_R$ is the last row of the inverse of $\mathbf{M}_\mathbf{a}$ in (64). Note that the use of a smooth safety filter makes $R_s$ differentiable.

Finally, with the safe angular velocity $R_s$, we construct the following *backstepping-based CBF* proposed by [37]:

$$h_\mathrm{b}(\mathbf{x}, t) = h_\mathrm{e}(\mathbf{r}, \mathbf{v}(\boldsymbol{\zeta}), t) - \frac{1}{2\mu_\mathrm{e}}(R_s - R)^2 \tag{46}$$

with a scaling constant $\mu_\mathrm{e} > 0$. Here, $R$ and $R_s$ are given by (2) and (45), respectively. Notice that $h_\mathrm{b}(\mathbf{x}, t) \leq h_\mathrm{e}(\mathbf{r}, \mathbf{v}(\boldsymbol{\zeta}), t)$ holds; hence, safety with respect to $h_\mathrm{b}$ implies safety with respect to $h_\mathrm{e}$. The CBF $h(\mathbf{x}, t) = h_\mathrm{b}(\mathbf{x}, t)$ can be used to execute the safety filter (21) for RTA, where the derivative of $h_\mathrm{b}$ can be obtained through lengthy calculation by differentiating the expression of $R_s$. Incorporating the turning rate $R$ into the CBF $h_\mathrm{b}$ allows us to leverage all control inputs for safety, including the angular velocity $P$ required for making the aircraft roll and consequently turn. This process means backstepping: taking a step back from rolling ($P$) to turning ($R$).

The backstepping-based CBF can be established for collision avoidance with $h_{\mathrm{e},i}$ from (39), geofencing with $h_{\mathrm{e},i}$ from (40), and the combination thereof with $h_\mathrm{e}$ from (41). Since the backstepping calculations are nontrivial, it is advantageous to first combine multiple safety constraints into a single one before executing backstepping rather than doing backstepping multiple times and then combining. The resulting safety filter (21) is in closed form. It yields safety with respect to the backstepping-based CBF, $h_\mathrm{b}(\mathbf{x}(t), t) \geq 0$, $\forall t \geq 0$, which implies safety with respect to the extended CBF, $h_\mathrm{e}(\mathbf{r}(t), \mathbf{v}(\boldsymbol{\zeta}(t)), t) \geq 0$, $\forall t \geq 0$, which finally leads to safe behavior considering the position-based CBF candidate, $h_\mathrm{p}(\mathbf{r}(t), t) \geq 0$, $\forall t \geq 0$. Note that the backstepping-based CBF (46) contains an additional term compared to the extended CBF (38), which is obtained by adding a term to the position-based CBF candidate. Thus, these CBFs have different numbers of parameters. To compare these CBFs and the resulting controllers, we use the same values for the parameters shared by the various CBFs (see Table I), and the comparison reflects the effect of the added terms.

The behavior of the aircraft with the proposed safety-critical controller is shown in Fig. 5. Model (7) is simulated with the safety filter [see (21) and (22)], the backstepping-based CBF [see (46)], and the parameters in Table I. The simulated scenario is simultaneous collision avoidance and geofencing where the geofence consists of two planar boundaries. The aircraft's RTA system with the proposed backstepping-based CBF guarantees safety with expected behavior. First, the safety filter intervenes and makes the aircraft accelerate, pitch up, and turn left to avoid collision with the other aircraft. Then, the aircraft is forced to turn right to avoid crossing the two geofence boundaries. In this case, the safety filter never stops intervening, as the aircraft keeps moving parallel to the geofence rather than returning to the original desired trajectory. Throughout the motion, the backstepping-based CBF $h_\mathrm{b}$ is kept nonnegative, which results in maintaining the three position-based CBFs $h_{\mathrm{p},i}$ (and their smooth under-approximation $h_\mathrm{p}$) nonnegative too, as highlighted in Fig. 5(b). These indicate that the underlying maneuvers are executed with guaranteed safety. While attitude and input constraints are not enforced by the controller, its parameters were tuned so that Euler angles $\theta$ and $\phi$ and inputs $P$, $Q$, and $A_\mathrm{T}$ in Fig. 5 evolve within reasonable limits.

### D. Model-Free RTA

The main challenge of designing an RTA system is the synthesis of a CBF that respects the cascaded structure of the underlying dynamics. This difficulty would vanish if we could use the position-based CBF candidate $h_\mathrm{p}$ directly in a model-free fashion. Next, we discuss a model-free RTA idea, originally introduced in [48] for robotic systems, and describe its benefits and drawbacks relative to the cascaded model-based RTA presented above.

The proposed model-free approach relies on the ability of the aircraft to track a commanded velocity. Velocity tracking (or trajectory tracking in general) is well-established for many aircraft, for example, the desired controller $\mathbf{k}_\mathrm{d}$ itself may often be a tracking controller. Such tracking controllers solve a stabilization problem (associated with Lyapunov theory) that is closely related to ensuring safety (associated with CBF theory) [48]. As a matter of fact, the tracking controller $\mathbf{k}_\mathrm{d}$ in

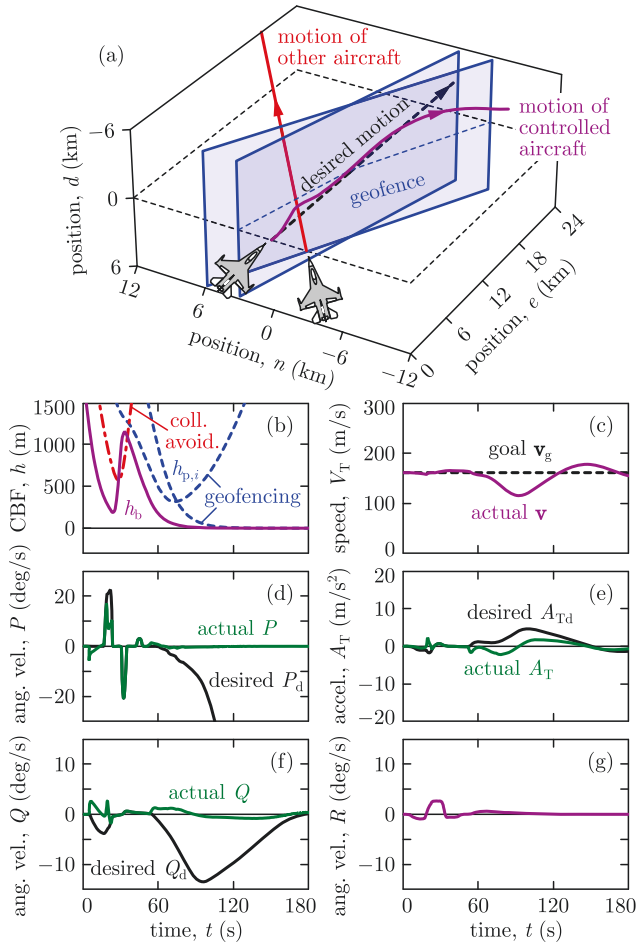


Fig. 5. Simulation of RTA in simultaneous collision avoidance and geofencing using backstepping-based CBF. Safety is successfully maintained by leveraging both acceleration, pitching, rolling, and turning. (a) Trajectory. (b) CBF. (c)–(g) Relevant states and inputs.

Appendix B is constructed through a backstepping procedure like the safety filter above. Thus, we intend to leverage the tracking controller (that already accounts for the cascaded dynamics) to address safety in a model-free fashion without going through a complicated control design procedure again.

To ensure safety, we design a safe velocity $\mathbf{v}_s(\mathbf{r},t)$ to be tracked, by considering the simplified model-independent kinematics $\dot{\mathbf{r}}=\mathbf{v}$ and regarding $\mathbf{v}$ as input. In particular, we synthesize the safe velocity $\mathbf{v}_s(\mathbf{r},t)$ so that it satisfies

$$\dot{h}_\mathrm{p}(\mathbf{r},t,\mathbf{v}_s(\mathbf{r},t)) \geq -\alpha_\mathrm{p}\big(h_\mathrm{p}(\mathbf{r},t)\big)+\sigma\left\|\frac{\partial h_\mathrm{p}}{\partial \mathbf{r}}(\mathbf{r},t)\right\|^2 \tag{47}$$

with some $\sigma>0$; see (37). The additional term with $\sigma$ originates from the theory of input-to-state safety [49], [50], and it is incorporated in order to provide robustness against tracking errors when the safe velocity $\mathbf{v}_s$ is being tracked.

We use constraint (47) to synthesize the safe velocity $\mathbf{v}_s$ from a desired velocity command $\mathbf{v}_\mathrm{d}$ via a smooth safety filter

$$\begin{aligned}
\mathbf{v}_s(\mathbf{r},t)&=\mathbf{v}_\mathrm{d}(\mathbf{r},t)+\Lambda(a_\mathbf{v}(\mathbf{r},t),\|\mathbf{b}_\mathbf{v}(\mathbf{r},t)\|)\mathbf{W}_\mathbf{v}\mathbf{b}_\mathbf{v}(\mathbf{r},t)^\top\\
a_\mathbf{v}(\mathbf{r},t)&=\dot{h}_\mathrm{p}(\mathbf{r},t,\mathbf{v}_\mathrm{d}(\mathbf{r},t))+\alpha_\mathrm{p}\big(h_\mathrm{p}(\mathbf{r},t)\big)-\sigma\left\|\frac{\partial h_\mathrm{p}}{\partial \mathbf{r}}(\mathbf{r},t)\right\|^2\\
\mathbf{b}_\mathbf{v}(\mathbf{r},t)&=\frac{\partial h_\mathrm{p}}{\partial \mathbf{r}}(\mathbf{r},t)\mathbf{W}_\mathbf{v}
\end{aligned}\tag{48}$$

with $\Lambda$ in (23) and a smoothing parameter $\nu_\mathbf{v}>0$; see (21). Matrix $\mathbf{W}_\mathbf{v}$ and $\mathbf{\Gamma}_\mathbf{v}=\mathbf{W}_\mathbf{v}^{-\top}\mathbf{W}_\mathbf{v}^{-1}$ weigh different velocity components when considering the deviation between $\mathbf{v}_s$ and $\mathbf{v}_\mathrm{d}$. For example, if we penalize the deviation of $\mathbf{v}_s$ from $\mathbf{v}_\mathrm{d}$ in the direction parallel to $\mathbf{v}_\mathrm{d}$ and in the perpendicular direction, respectively, with weights 1 and $\Gamma_\mathbf{v}$, then

$$\begin{aligned}
\mathbf{\Gamma}_\mathbf{v}&=\mathbf{P}_\mathbf{v}+\Gamma_\mathbf{v}(\mathbf{I}-\mathbf{P}_\mathbf{v})\\
\mathbf{W}_\mathbf{v}&=\mathbf{P}_\mathbf{v}+\frac{1}{\sqrt{\Gamma_\mathbf{v}}}(\mathbf{I}-\mathbf{P}_\mathbf{v}),\quad \mathbf{P}_\mathbf{v}=\frac{\mathbf{v}_\mathrm{d}\mathbf{v}_\mathrm{d}^\top}{\|\mathbf{v}_\mathrm{d}\|^2}.
\end{aligned}\tag{49}$$

Equation (48) represents model-free RTA in the sense that it does not use expressions from model (7) but only the position-based CBF candidate $h_\mathrm{p}$ in (31)–(36). As opposed, controllers with the extended CBF (38)–(42) and the backstepping-based CBF (46) depended on the model. Tracking the safe velocity $\mathbf{v}_s(\mathbf{r},t)$ obtained from the model-free RTA yields safe behavior under certain assumptions about the tracking controller, as established in the following. Note that making $\mathbf{v}_s$ to be smooth with the smooth safety filter formula facilitates velocity tracking.

*Assumption 2:* Controller $\mathbf{u}=\mathbf{k}(\mathbf{x},t)$ yields exponentially stable tracking of the safe velocity $\mathbf{v}_s(\mathbf{r},t)$. That is, there exist a Lyapunov function $V{:}\mathbb{R}^n\times\mathbb{R}_{\geq 0}\to\mathbb{R}_{\geq 0}$ and $\lambda>0$ such that

$$V(\mathbf{x},t)\geq\frac{1}{2}\|\mathbf{v}_s(\mathbf{r},t)-\mathbf{v}(\boldsymbol{\zeta})\|^2 \tag{50}$$

$$\dot{V}(\mathbf{x},t,\mathbf{k}(\mathbf{x},t))\leq-\lambda V(\mathbf{x},t) \tag{51}$$

where the derivative of $V$ is taken along (7).

For example, the controller in Appendix B satisfies Assumption 2. Then, it can be established that the system that tracks $\mathbf{v}_s$ is safe if $\alpha_\mathrm{p}$ is chosen such that $\alpha_\mathrm{p}(r)=\gamma_\mathrm{p}r$ and $\gamma_\mathrm{p}<\lambda$.

*Theorem 1:* If the safe velocity $\mathbf{v}_s(\mathbf{r},t)$ in (48) with $\alpha_\mathrm{p}(r)=\gamma_\mathrm{p}r$ is tracked by (7) with a controller $\mathbf{u}=\mathbf{k}(\mathbf{x},t)$ such that (50) and (51) hold with $\gamma_\mathrm{p}<\lambda$, then the following set $\mathcal{S}_V$ is forward invariant along the closed-loop dynamics (11):

$$\begin{aligned}
\mathcal{S}_V&=\big\{(\mathbf{x},t)\in\mathbb{R}^n\times\mathbb{R}_{\geq 0}:h_V(\mathbf{x},t)\geq 0\big\}\\
h_V(\mathbf{x},t)&=h_\mathrm{p}(\mathbf{r},t)-\frac{V(\mathbf{x},t)}{2\sigma\left(\lambda-\gamma_\mathrm{p}\right)}.
\end{aligned}\tag{52}$$

This guarantees that $(\mathbf{x}_0,0)\in\mathcal{S}_V\implies(\mathbf{x}(t),t)\in\mathcal{S}_V,\ \forall t\geq 0$, which leads to $(\mathbf{r}(t),t)\in\mathcal{S},\ \forall t\geq 0$.

Note that the restriction $(\mathbf{x}_0,0)\in\mathcal{S}_V$ on the initial condition is stricter than $(\mathbf{r}_0,0)\in\mathcal{S}$ due to the term $V(\mathbf{x}_0,0)/(2\sigma(\lambda-\gamma_\mathrm{p}))$ in the expression of $h_V(\mathbf{x}_0,0)$. However, the magnitude of this term can be reduced by increasing $\sigma$.

*Proof:* The forward invariance of $\mathcal{S}_V$ follows from Lemma 1, by showing that the following inequality holds:

$$\begin{aligned}
&\dot{h}_V(\mathbf{x},t,\mathbf{k}(\mathbf{x},t))+\gamma_\mathrm{p}h_V(\mathbf{x},t)\\
&=\dot{h}_\mathrm{p}(\mathbf{r},t,\mathbf{v}(\boldsymbol{\zeta}))+\gamma_\mathrm{p}h_\mathrm{p}(\mathbf{r},t)-\frac{\dot{V}(\mathbf{x},t,\mathbf{k}(\mathbf{x},t))}{2\sigma\left(\lambda-\gamma_\mathrm{p}\right)}\\
&\quad-\frac{\gamma_\mathrm{p}V(\mathbf{x},t)}{2\sigma\left(\lambda-\gamma_\mathrm{p}\right)}\\
&\geq\sigma\left\|\frac{\partial h_\mathrm{p}}{\partial \mathbf{r}}(\mathbf{r},t)\right\|^2+\dot{h}_\mathrm{p}(\mathbf{r},t,\mathbf{v}(\boldsymbol{\zeta}))-\dot{h}_\mathrm{p}(\mathbf{r},t,\mathbf{v}_s(\mathbf{r},t))\\
&\quad+\frac{V(\mathbf{x},t)}{2\sigma}
\end{aligned}$$

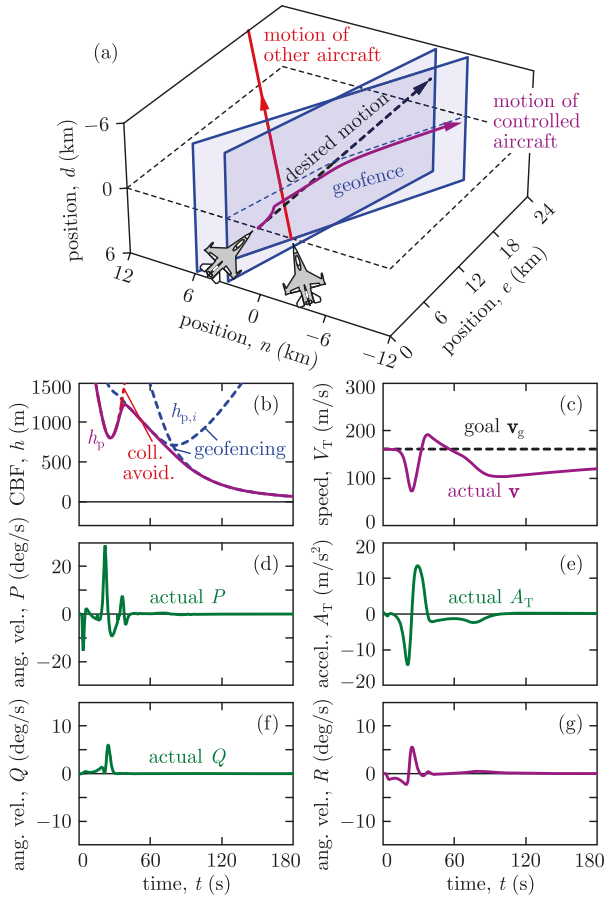


Fig. 6. Simulation of RTA in simultaneous collision avoidance and geofencing using a model-free approach with position-based CBF. Safety is maintained, while control inputs are larger than with the model-based method in Fig. 5. (a) Trajectory. (b) CBF. (c)–(g) Relevant states and inputs.

$$\begin{aligned}
&\geq \sigma \left\| \frac{\partial h_{\mathrm{p}}}{\partial \mathbf{r}}(\mathbf{r},t) \right\|^2 + \frac{\partial h_{\mathrm{p}}}{\partial \mathbf{r}}(\mathbf{r},t)(\mathbf{v}(\boldsymbol{\zeta}) - \mathbf{v}_s(\mathbf{r},t)) + \frac{V(\mathbf{x},t)}{2\sigma} \\
&\geq \sigma \left\| \frac{\partial h_{\mathrm{p}}}{\partial \mathbf{r}}(\mathbf{r},t) \right\|^2 - \left\| \frac{\partial h_{\mathrm{p}}}{\partial \mathbf{r}}(\mathbf{r},t) \right\| \sqrt{2V(\mathbf{x},t)} + \frac{V(\mathbf{x},t)}{2\sigma} \\
&\geq \left( \sqrt{\sigma} \left\| \frac{\partial h_{\mathrm{p}}}{\partial \mathbf{r}}(\mathbf{r},t) \right\| - \sqrt{\frac{V(\mathbf{x},t)}{2\sigma}} \right)^2 \\
&\geq 0
\end{aligned} \tag{53}$$

where first we used (52); second, we substituted (47) and (51); then, we expressed the difference of the $\dot{h}_{\mathrm{p}}$ terms; next, we applied the Cauchy–Schwartz inequality and (50); and finally, we completed the square. Furthermore, safety with respect to $\mathcal{S}_V$ implies safety with respect to $\mathcal{S}$, $(\mathbf{x},t) \in \mathcal{S}_V \implies (\mathbf{r},t) \in \mathcal{S}$ since $h_{\mathrm{p}}(\mathbf{r},t) \geq h_V(\mathbf{x},t)$ if $\gamma_{\mathrm{p}} < \lambda$. □

The performance of the model-free RTA system is demonstrated in Fig. 6. The same simultaneous collision avoidance and geofencing task is executed as in Fig. 5. Model (7) is simulated with the tracking controller from Appendix B, which is used to track the safe velocity resulting from the model-free smooth safety filter (48). The parameters are listed in Table I. The model-free RTA successfully maintains safety in both collision avoidance and geofencing, by keeping the smooth under-approximation $h_{\mathrm{p}}$ of the underlying CBF candidates $h_{\mathrm{p},i}$ nonnegative. Note that the aircraft with the model-free RTA only uses deceleration and turning to avoid collisions, and it does not move up or down. This is due to symmetry (and not due to the lack of capability to leverage pitching): the controlled aircraft moves in the same horizontal plane as the other aircraft since moving up or down would be indifferent because the model-free safety filter does not contain any term like gravity to break the symmetry.

The advantage of this model-free approach is that RTA becomes simpler, and it leverages the existing flight controller (i.e., the desired tracking controller). Furthermore, this approach does not modify the low-level flight control system that flies the aircraft with desired stable behavior, but only uses high-level commands about which direction and how fast the aircraft should fly. The disadvantage of the model-free approach is that it is hard to tune it in a way that the safe velocity is easy to track. This can be observed in Fig. 6(d) and (e), which highlight that tracking the safe velocity results in larger control inputs than the ones with the model-based RTA in Fig. 5(d) and (e).

In summary, the velocity-based extended CBF in Section IV-B may succeed in collision avoidance but may fail geofencing tasks because it is unable to leverage all control inputs (like roll motion) to guarantee safety. The backstepping-based CBF in Section IV-C and the model-free RTA in Section IV-D both leverage all control inputs and can successfully accomplish simultaneous collision avoidance and geofencing. The model-based RTA with backstepping may be preferable over the model-free RTA in terms of the magnitude of control inputs. Of course, incorporating model information enhances the performance of the control design, but at the price of complexity—this is a fundamental tradeoff observed in many control systems. Nevertheless, the resulting controllers enjoy the formal safety guarantees provided by CBF theory, with both the model-based and model-free RTA.

### *E. Validation With a Full-Envelope System Identified Model*

Finally, to validate that the RTA developed for model (7) could provide safety on a higher fidelity model, simulations were run using an implementation of the linearized, stitched model of the X-62 Variable-stability In-flight Simulator Test Aircraft (VISTA) (modified two-seat F-16 aircraft) shown in Fig. 7(a). This is a proprietary dynamical model that was developed based on [51], where flight test data were used to identify and combine (i.e., stitch together) linear state-space models with time delays that represent the 6-DoF dynamics at various flight conditions. The VISTA simulation was equipped with a neural network controller (NNC) as the nominal flight controller. The NNC was trained via reinforcement learning [14], [52], and it provides the necessary commands in terms of power lever angle (PLA), roll rate, and normal load factor to achieve stable wing-level flight at given altitude, heading, and speed. With the RTA system proposed in this article, we seek to modulate the NNC whenever necessary to ensure safety with respect to collision avoidance. This serves

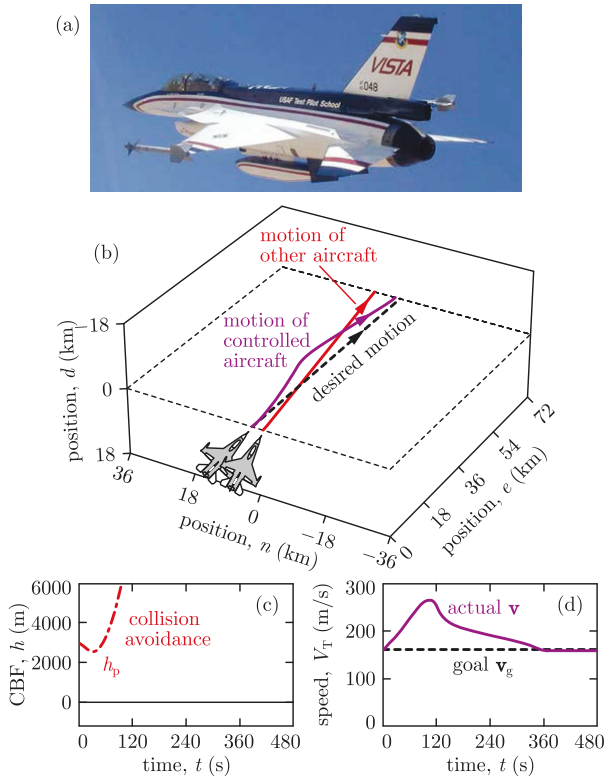


Fig. 7. Validation simulation of the proposed RTA on a high-fidelity model. (a) VISTA, whose linearized, stitched dynamical model from [51] was simulated. (b)–(d) Simulation of RTA in collision avoidance. The VISTA tracks the acceleration and angular velocity commands obtained from the RTA that uses 3-D Dubins model and backstepping-based CBF. Safety is maintained in high-fidelity simulation.

as the first step toward validating that the Dubins model-based RTA system can be executed on more realistic models, while a comprehensive validation, including experiments on hardware, is left for future work.

The high-fidelity simulation results are shown in Fig. 7. A collision avoidance scenario is considered where the VISTA is initially headed toward another aircraft and an evasive maneuver is required from the RTA. To track the nominal trajectory, the NNC provides PLA, roll rate, and normal load factor commands, which can be converted to desired acceleration and angular velocity (i.e., $A_T$, $P$, and $Q$). The RTA modifies these desired values to safe acceleration and angular velocity commands. The safe commands are converted back to PLA, roll rate, and normal load factor, and tracked by the VISTA. The RTA is based on the 3-D Dubins model (7) and the backstepping-based CBF (46), which was previously demonstrated in Fig. 5. The corresponding simulated trajectories are shown in Fig. 7(b)–(d). Remarkably, the RTA constructed from the lower fidelity 3-D Dubins model is capable of providing the required evasive maneuver, and it successfully guarantees safety even in a high-fidelity simulation—thanks to a carefully designed CBF in (46).

We remark that the high-fidelity simulation environment was also monitoring additional constraints on the aircraft, such as limits for angle of attack, angle of side slip, pilot $g$ command, roll rate, yaw rate, normal load factor, side load factor, speed, control surface deflections, and their rates. These constraints include some of the states and inputs of the Dubins model, as well as additional quantities that are not present in the model. Upon violation of these constraints, the simulation terminated. By tuning the parameters of the proposed controller, we prevented the violation of these constraints while maintaining safety with respect to collision avoidance.

Our future plan is to formally address these state and input constraints by more complex CBF formulations such as the backup set method [15], [28]. This method uses a backup controller that could maintain safe behavior with input-constrained conservative maneuvers like turning the aircraft away from unsafe regions. By forward integrating the closed-loop dynamics and computing the corresponding backup trajectory, safety filters can enforce that this trajectory leads to a backup set without violating safety. This strategy ultimately provides formal guarantees of safety while satisfying input constraints, and its safety filter may integrate a desired flight controller to achieve a higher performance than using a conservative backup controller directly. Developing the required backup controllers and sets, however, needs significant future research.

## V. Conclusion

In this article, we developed an RTA system for fixed-wing aircraft that intervenes into the operation of existing flight controllers whenever necessary for the safety of the aircraft. Specifically, we used CBFs to establish controllers with formal safety guarantees for collision avoidance, geofencing, and the simultaneous execution thereof. We established and proved the safety guarantees provided by these controllers, and we demonstrated safe operation by numerical simulation of a nonlinear kinematic aircraft model and a high-fidelity dynamical model. We highlighted that different choices of CBFs—high-order and backstepping-based CBFs that use model information, and simplified position-based CBF candidates in a model-free context—lead to qualitatively different behaviors while guaranteeing safety. In future work, we plan to enforce safety with respect to other constraints in flight envelopes such as angle of attack bounds or attitude constraints. We also plan to formally address and enforce input constraints with techniques like the backup set method [15]. Finally, we plan to validate safety-critical controllers by hardware experiments.

## Appendix A
## Equations of Motion

Here, we derive the equations of the 3-D Dubins model see (1) and (2). First, we revisit a 6-DoF model from [33] that governs the rigid-body dynamics of fixed-wing aircraft. Then, we simplify these dynamics to the 3-D Dubins model through certain assumptions. While the upcoming equations are well-known in the literature, they are required by the controllers in the main body of this article.

Consider the aircraft illustrated in Fig. 2. To describe the aircraft's motion, we rely on the North, East, and down position coordinates, $n$, $e$, and $d$; the roll, pitch, and yaw Euler

angles, $\phi$, $\theta$, and $\psi$; the speed, $V_{\mathrm{T}}$, angle of side slip, $\beta$, angle of attack, $\alpha$; and the front, right, down angular velocities, $P$, $Q$, and $R$

$$\mathbf{r} = \begin{bmatrix} n \\ e \\ d \end{bmatrix}, \quad \boldsymbol{\xi} = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}, \quad \boldsymbol{\eta} = \begin{bmatrix} V_{\mathrm{T}} \\ \beta \\ \alpha \end{bmatrix}, \quad \boldsymbol{\omega} = \begin{bmatrix} P \\ Q \\ R \end{bmatrix}. \tag{54}$$

This yields 12 states that evolve according to the 6-DoF model.

To derive the governing equations of motion, we use three reference frames: the Earth frame, $e$, to express global position; the body frame, $b$, aligned with the aircraft's body; and the wind frame, $w$, aligned with the velocity vector of the center of mass. The orientation of the body frame relative to the Earth frame is given by the Euler angles $\boldsymbol{\xi}$, whereas the orientation of the wind frame relative to the body frame is described by the angle of side slip $\beta$ and angle of attack $\alpha$. The frames are related by the transformation matrices

$$\begin{aligned} \mathbf{R}_{eb}(\boldsymbol{\xi}) &= \begin{bmatrix} c_\psi & -s_\psi & 0 \\ s_\psi & c_\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_\theta & 0 & s_\theta \\ 0 & 1 & 0 \\ -s_\theta & 0 & c_\theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi & c_\phi \end{bmatrix} \\ \mathbf{R}_{bw}(\boldsymbol{\eta}) &= \begin{bmatrix} c_\alpha & 0 & -s_\alpha \\ 0 & 1 & 0 \\ s_\alpha & 0 & c_\alpha \end{bmatrix} \begin{bmatrix} c_\beta & -s_\beta & 0 \\ s_\beta & c_\beta & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ \mathbf{R}_{be}(\boldsymbol{\xi}) &= \mathbf{R}_{eb}(\boldsymbol{\xi})^\top, \quad \mathbf{R}_{wb}(\boldsymbol{\eta}) = \mathbf{R}_{bw}(\boldsymbol{\eta})^\top \end{aligned} \tag{55}$$

where $c_{(\cdot)}$ abbreviates $\cos(\cdot)$ and $s_{(\cdot)}$ abbreviates $\sin(\cdot)$.

### A. Kinematics

First, we characterize the kinematics of the aircraft through the evolution of its position and orientation. The position $\mathbf{r}$ evolves according to the expression of the velocity $\mathbf{v}$

$$\dot{\mathbf{r}} = \mathbf{v}(\boldsymbol{\xi}, \boldsymbol{\eta}). \tag{56}$$

The velocity of the aircraft's center of mass can be given in the Earth, body, and wind frames, respectively, by

$$\begin{aligned} \mathbf{v}(\boldsymbol{\xi}, \boldsymbol{\eta}) &= \mathbf{R}_{eb}(\boldsymbol{\xi})\mathbf{v}_{\mathrm{b}}(\boldsymbol{\eta}) \\ \mathbf{v}_{\mathrm{b}}(\boldsymbol{\eta}) &= \mathbf{R}_{bw}(\boldsymbol{\eta})\mathbf{v}_w(\boldsymbol{\eta}) \\ \mathbf{v}_w(\boldsymbol{\eta}) &= \begin{bmatrix} V_{\mathrm{T}} & 0 & 0 \end{bmatrix}^\top. \end{aligned} \tag{57}$$

The Euler angles $\boldsymbol{\xi}$ are related to the angular velocity $\boldsymbol{\omega}$ by the Euler angle kinematics [33]

$$\dot{\boldsymbol{\xi}} = \mathbf{H}(\boldsymbol{\xi})\boldsymbol{\omega} \tag{58}$$

where the coefficient matrix is

$$\mathbf{H}(\boldsymbol{\xi}) = \begin{bmatrix} 1 & s_\phi t_\theta & c_\phi t_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & s_\phi / c_\theta & c_\phi / c_\theta \end{bmatrix} \tag{59}$$

and $t_\theta$ abbreviates $\tan(\theta)$. Moreover, after laborious calculation, it can be shown that the following useful identities hold:

$$\begin{aligned} \frac{\partial \mathbf{v}}{\partial \boldsymbol{\xi}}(\boldsymbol{\xi}, \boldsymbol{\eta})\mathbf{H}(\boldsymbol{\xi})\boldsymbol{\omega} &= \mathbf{R}_{eb}(\boldsymbol{\xi})(\boldsymbol{\omega} \times \mathbf{v}_{\mathrm{b}}(\boldsymbol{\eta})) \\ \frac{\partial \mathbf{v}}{\partial \boldsymbol{\eta}}(\boldsymbol{\xi}, \boldsymbol{\eta}) &= \mathbf{R}_{eb}(\boldsymbol{\xi})\mathbf{R}_{bw}(\boldsymbol{\eta}) \begin{bmatrix} 1 & 0 & 0 \\ 0 & V_{\mathrm{T}} & 0 \\ 0 & 0 & V_{\mathrm{T}} c_\beta \end{bmatrix}. \end{aligned} \tag{60}$$

These expressions define the underlying acceleration

$$\dot{\mathbf{v}} = \frac{\partial \mathbf{v}}{\partial \boldsymbol{\xi}}(\boldsymbol{\xi}, \boldsymbol{\eta})\dot{\boldsymbol{\xi}} + \frac{\partial \mathbf{v}}{\partial \eta}(\boldsymbol{\xi}, \boldsymbol{\eta})\dot{\boldsymbol{\eta}} \tag{61}$$

where the formula of $\dot{\boldsymbol{\eta}}$ is introduced in the following.

The kinematics simplify as follows in the special case $\beta \equiv 0$ and $\alpha \equiv 0$. The body–wind frame transformation reduces to $\mathbf{R}_{wb}(\boldsymbol{\eta}) = \mathbf{R}_{bw}(\boldsymbol{\eta}) = \mathbf{I}$; hence, the velocity becomes

$$\mathbf{v}(\boldsymbol{\xi}, \boldsymbol{\eta}) = \mathbf{R}_{eb}(\boldsymbol{\xi}) \begin{bmatrix} V_{\mathrm{T}} \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} V_{\mathrm{T}} c_\theta c_\psi \\ V_{\mathrm{T}} c_\theta s_\psi \\ -V_{\mathrm{T}} s_\theta \end{bmatrix}. \tag{62}$$

That is, the velocity only depends on the state $\boldsymbol{\zeta}$ defined in (4), which is emphasized by an abuse of notation in the main body of this article: $\mathbf{v}(\boldsymbol{\zeta})$ is used instead of $\mathbf{v}(\boldsymbol{\xi}, \boldsymbol{\eta})$. Accordingly, $(\partial \mathbf{v})/(\partial \phi)(\boldsymbol{\xi}, \boldsymbol{\eta}) = (\partial \mathbf{v})/(\partial \beta)(\boldsymbol{\xi}, \boldsymbol{\eta}) = (\partial \mathbf{v})/(\partial \alpha)(\boldsymbol{\xi}, \boldsymbol{\eta}) = \mathbf{0}$ holds, and the acceleration in (61) reads

$$\dot{\mathbf{v}} = \mathbf{M}_{\mathbf{a}}(\boldsymbol{\xi}, \boldsymbol{\eta}) \begin{bmatrix} A_{\mathrm{T}} \\ Q \\ R \end{bmatrix} \tag{63}$$

where $A_{\mathrm{T}} = \dot{V}_{\mathrm{T}}$ and

$$\begin{aligned} &\mathbf{M}_{\mathbf{a}}(\boldsymbol{\xi}, \boldsymbol{\eta}) \\ &= \begin{bmatrix} \dfrac{\partial \mathbf{v}}{\partial V_{\mathrm{T}}}(\boldsymbol{\xi}, \boldsymbol{\eta}) & \left[ \dfrac{\partial \mathbf{v}}{\partial \theta}(\boldsymbol{\xi}, \boldsymbol{\eta}) \quad \dfrac{\partial \mathbf{v}}{\partial \psi}(\boldsymbol{\xi}, \boldsymbol{\eta}) \right] \mathbf{H}_{\theta\psi}(\boldsymbol{\xi}) \end{bmatrix} \\ &= \begin{bmatrix} c_\theta c_\psi & -V_{\mathrm{T}}\left(c_\phi s_\theta c_\psi + s_\phi s_\psi\right) & V_{\mathrm{T}}\left(s_\phi s_\theta c_\psi - c_\phi s_\psi\right) \\ c_\theta s_\psi & V_{\mathrm{T}}\left(-c_\phi s_\theta s_\psi + s_\phi c_\psi\right) & V_{\mathrm{T}}\left(s_\phi s_\theta s_\psi + c_\phi c_\psi\right) \\ -s_\theta & -V_{\mathrm{T}} c_\phi c_\theta & V_{\mathrm{T}} s_\phi c_\theta \end{bmatrix} \end{aligned} \tag{64}$$

with $\mathbf{H}_{\theta\psi}$ being the bottom right $2 \times 2$ block of $\mathbf{H}$ in (59).

### B. Dynamics

Next, we describe the dynamics of the aircraft through the evolution of its velocity and angular velocity. These are related to the forces and moments acting on the body: the thrust force, $\mathbf{F}_{\mathrm{T}}$, assumed to be aligned with the body with magnitude $F_{\mathrm{T}}$; the aerodynamics forces, $\mathbf{F}_A$, including the lift, $L$, drag, $D$, and crosswind, $C$, components in wind frame; the gravitational force $m\mathbf{g}_D$ with the mass $m$ and the gravitational acceleration $\mathbf{g}_D$ aligned with the down axis in Earth frame with magnitude $g_D$; and the moments $\mathbf{M}$ exerted on the aircraft, with body-frame components $L$, $M$, and $N$

$$\mathbf{F}_{\mathrm{T}} = \begin{bmatrix} F_{\mathrm{T}} \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{F}_A(\boldsymbol{\eta}) = \begin{bmatrix} D(\boldsymbol{\eta}) \\ C(\boldsymbol{\eta}) \\ L(\boldsymbol{\eta}) \end{bmatrix}, \quad \mathbf{g}_D = \begin{bmatrix} 0 \\ 0 \\ g_D \end{bmatrix}$$
$$\mathbf{M} = \begin{bmatrix} L \\ M \\ N \end{bmatrix}. \tag{65}$$

The acceleration of the aircraft's center of mass can be given in the Earth frame by

$$\dot{\mathbf{v}} = \frac{1}{m}\mathbf{R}_{eb}(\boldsymbol{\xi})(\mathbf{F}_{\mathrm{T}} - \mathbf{R}_{bw}(\boldsymbol{\eta})\mathbf{F}_A(\boldsymbol{\eta})) + \mathbf{g}_D. \tag{66}$$

Expressing $\dot{\boldsymbol{\eta}}$ from (61) and substituting (58)–(60) and (66) lead to the wind-axes force equations

$$\dot{\boldsymbol{\eta}} = \mathbf{f}_\eta(\boldsymbol{\xi}, \boldsymbol{\eta}, \boldsymbol{\omega}) + \mathbf{g}_\eta(\boldsymbol{\eta}) F_\mathrm{T} \tag{67}$$

where

$$\begin{aligned}
\mathbf{f}_\eta(\boldsymbol{\xi}, \boldsymbol{\eta}, \boldsymbol{\omega}) &= \mathbf{A}(\boldsymbol{\eta})\Big(-\frac{1}{m}\mathbf{F}_A(\boldsymbol{\eta}) + \mathbf{R}_{wb}(\boldsymbol{\eta})\mathbf{R}_{be}(\boldsymbol{\xi})\mathbf{g}_D \\
&\qquad + \mathbf{v}_w(\boldsymbol{\eta}) \times (\mathbf{R}_{wb}(\boldsymbol{\eta})\boldsymbol{\omega})\Big) \\
\mathbf{g}_\eta(\boldsymbol{\eta}) &= \mathbf{A}(\boldsymbol{\eta})\frac{1}{m}\mathbf{R}_{wb}(\boldsymbol{\eta})\begin{bmatrix}1\\0\\0\end{bmatrix} \\
\mathbf{A}(\boldsymbol{\eta}) &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \dfrac{1}{V_\mathrm{T}} & 0 \\ 0 & 0 & \dfrac{1}{V_\mathrm{T} c_\beta} \end{bmatrix}.
\end{aligned} \tag{68}$$

Furthermore, the body-frame angular acceleration is

$$\dot{\boldsymbol{\omega}} = \mathbf{f}_\omega(\boldsymbol{\omega}) + \mathbf{g}_\omega \mathbf{M} \tag{69}$$

where $\mathbf{f}_\omega$ and $\mathbf{g}_\omega$ include the mass moment of inertia $\mathbf{J}$

$$\mathbf{f}_\omega(\boldsymbol{\omega}) = -\mathbf{J}^{-1}(\boldsymbol{\omega} \times (\mathbf{J}\boldsymbol{\omega})), \quad \mathbf{g}_\omega = \mathbf{J}^{-1}. \tag{70}$$

### C. Governing Equations

The kinematic and dynamic equations finally lead to the following *6-DoF model*:

$$\begin{aligned}
\dot{\mathbf{r}} &= \mathbf{v}(\boldsymbol{\xi}, \boldsymbol{\eta}) \\
\dot{\boldsymbol{\xi}} &= \mathbf{H}(\boldsymbol{\xi})\boldsymbol{\omega} \\
\dot{\boldsymbol{\eta}} &= \mathbf{f}_\eta(\boldsymbol{\xi}, \boldsymbol{\eta}, \boldsymbol{\omega}) + \mathbf{g}_\eta(\boldsymbol{\eta}) F_\mathrm{T} \\
\dot{\boldsymbol{\omega}} &= \mathbf{f}_\omega(\boldsymbol{\omega}) + \mathbf{g}_\omega \mathbf{M}.
\end{aligned} \tag{71}$$

We simplify the 6-DoF model to the 3-D Dubins model using the following assumptions.

*Assumption 3:* The angle of attack, the angle of side slip, and the crosswind force are zero: $\alpha \equiv 0$, $\beta \equiv 0$, and $C(\boldsymbol{\eta}) \equiv 0$. The dynamics of the angular velocity $\boldsymbol{\omega}$ are neglected.

Furthermore, we use the longitudinal acceleration $A_\mathrm{T} = \dot{V}_\mathrm{T}$ directly instead of the thrust force $F_\mathrm{T}$. Then, the dynamics of $\beta$, $\alpha$, and $\boldsymbol{\omega}$ are dropped, leading to the *3-D Dubins model*

$$\begin{aligned}
\dot{\mathbf{r}} &= \mathbf{v}(\boldsymbol{\xi}, \boldsymbol{\eta}) \\
\dot{\boldsymbol{\xi}} &= \mathbf{H}(\boldsymbol{\xi})\boldsymbol{\omega} \\
\dot{V}_\mathrm{T} &= A_\mathrm{T}
\end{aligned} \tag{72}$$

where $\boldsymbol{\eta} = \begin{bmatrix} V_\mathrm{T} & 0 & 0 \end{bmatrix}^\top$. By spelling out all terms, this gives (1). Furthermore, since $\alpha \equiv 0$, $\beta \equiv 0$, and $C(\boldsymbol{\eta}) \equiv 0$, we obtain $\mathbf{R}_{wb}(\boldsymbol{\eta}) = \mathbf{R}_{bw}(\boldsymbol{\eta}) = \mathbf{I}$, and the second component of the body-axes force equations (67), $\dot{\beta} \equiv 0$, simplifies to (2). Moreover, the third component of (67), $\dot{\alpha} \equiv 0$, gives the required lift force, $L = m(g_D c_\phi c_\theta + V_\mathrm{T} Q)$, that could keep $\alpha \equiv 0$. This shows that the assumption $\alpha \equiv 0$ implies a specific state-dependent lift force. Although this may not be true in practice, we use this assumption as an approximation to reduce the full dynamics model to simplified kinematics that represent the aircraft's overall motion with less complexity.

## Appendix B
## Tracking Controller

Finally, we establish a velocity tracking controller for the 3-D Dubins model (7). Note that this controller is independent of the proposed RTA, it is used in simulation examples only, and other tracking controllers could also be considered.

The velocity tracking controller is designed to track a velocity command $\mathbf{v}_\mathrm{c}$. For the model-based RTA examples in Figs. 3–5, this command is a desired velocity, $\mathbf{v}_\mathrm{c}(\mathbf{r}, t) = \mathbf{v}_\mathrm{d}(\mathbf{r}, t)$, chosen to track a goal trajectory $\mathbf{r}_g(t)$

$$\mathbf{v}_\mathrm{d}(\mathbf{r}, t) = \mathbf{v}_g(t) + \mathbf{K}_\mathbf{r}\big(\mathbf{r}_g(t) - \mathbf{r}\big) \tag{73}$$

with a symmetric positive definite gain $\mathbf{K}_\mathbf{r}$, where $\dot{\mathbf{r}}_g = \mathbf{v}_g$. For the model-free RTA example in Fig. 6, the safe velocity command $\mathbf{v}_\mathrm{c}(\mathbf{r}, t) = \mathbf{v}_s(\mathbf{r}, t)$ from (48) is tracked.

We construct an exponentially stable tracking controller for the 3-D Dubins model (7) by the help of *backstepping* with control Lyapunov functions (CLFs) [53] in two steps.

1) We design a desired acceleration $\mathbf{a}_\mathrm{d}$ that would lead to exponentially stable velocity tracking, with corresponding thrust $A_\mathrm{T}$, angular velocity $Q$ (related to pitching), and desired angular velocity $R_\mathrm{d}$ (related to turning). $A_\mathrm{T}$ and $Q$ are inputs to the 3-D Dubins model, making the aircraft accelerate and pitch. However, the aircraft cannot be commanded to turn, as $R$ is not an input, and the desired acceleration cannot be realized directly.
2) We use $R_\mathrm{d}$ in a CLF—constructed based on backstepping—to synthesize the remaining input: the angular velocity $P$ (related to rolling). Commanding $P$ drives the actual angular velocity $R$ to the desired value $R_\mathrm{d}$, and a desired turning rate is achieved through rolling. This ultimately yields exponentially stable velocity tracking.

With this construction, our goal is to establish exponentially stable velocity tracking through the CLF candidate

$$V_0(\mathbf{x}, t) = \frac{1}{2}\|\mathbf{v}_\mathrm{c}(\mathbf{r}, t) - \mathbf{v}(\boldsymbol{\zeta})\|^2 \tag{74}$$

that needs to be driven to zero. Accordingly, the desired acceleration is designed as

$$\mathbf{a}_\mathrm{d}(\mathbf{x}, t) = \mathbf{a}_\mathrm{c}(\mathbf{x}, t) + \frac{1}{2}\mathbf{K}_\mathbf{v}(\mathbf{v}_\mathrm{c}(\mathbf{r}, t) - \mathbf{v}(\boldsymbol{\zeta})) \tag{75}$$

where $\dot{\mathbf{v}}_c = \mathbf{a}_\mathrm{c}$ and $\mathbf{K}_\mathbf{v}$ is a symmetric positive definite gain. The desired acceleration ensures exponential stability such that the following holds for $\dot{\mathbf{v}} = \mathbf{a}_\mathrm{d}(\mathbf{x}, t)$ and for any $\lambda > 0$ that is smaller than or equal to the smallest eigenvalue of $\mathbf{K}_\mathbf{v}$:

$$\dot{V}_0(\mathbf{x}, t) = -\frac{1}{2}\|\mathbf{v}_\mathrm{c}(\mathbf{r}, t) - \mathbf{v}(\boldsymbol{\zeta})\|^2_{\mathbf{K}_\mathbf{v}} \leq -\lambda V_0(\mathbf{x}, t). \tag{76}$$

We convert the desired acceleration $\mathbf{a}_\mathrm{d}$ to the inputs $A_\mathrm{T}$, $Q$, and the desired angular velocity $R_\mathrm{d}$ based on (63)

$$\begin{bmatrix} A_\mathrm{T} \\ Q \\ R_\mathrm{d} \end{bmatrix} = \mathbf{M}_\mathbf{a}(\boldsymbol{\xi}, \boldsymbol{\eta})^{-1}\mathbf{a}_\mathrm{d}(\mathbf{x}, t). \tag{77}$$

The resulting $A_{\mathrm{T}}$ and $Q$ can be directly commanded as inputs, which leads to the actual acceleration

$$\dot{\mathbf{v}} = \mathbf{a}_{\mathrm{d}}(\mathbf{x}, t) + \mathbf{M}_R(\boldsymbol{\xi}, \boldsymbol{\eta})(R - R_{\mathrm{d}}) \tag{78}$$

where $\mathbf{M}_R$ is the last column of $\mathbf{M}_{\mathbf{a}}$ in (64). Meanwhile, we can use $R_{\mathrm{d}}$ to design the remaining input $P$ that does not appear in the derivative of $V_0$. This means backstepping: taking a step back from rolling ($P$) to turning ($R_{\mathrm{d}}$).

In particular, we use backstepping to construct a CLF [53]

$$V(\mathbf{x}, t) = \frac{1}{2}\|\mathbf{v}_{\mathrm{c}}(\mathbf{r}, t) - \mathbf{v}(\boldsymbol{\zeta})\|^2 + \frac{1}{2\mu}(R - R_{\mathrm{d}})^2 \tag{79}$$

with a scaling constant $\mu > 0$. Here, $R$ and $R_{\mathrm{d}}$ are given by (2) and (77), respectively, while their dynamics are of the form

$$\begin{aligned} \dot{R} &= f_R(\mathbf{x}, t) + g_R(\mathbf{x}, t)P \\ \dot{R}_{\mathrm{d}} &= f_{R_{\mathrm{d}}}(\mathbf{x}, t) + g_{R_{\mathrm{d}}}(\mathbf{x}, t)P \end{aligned} \tag{80}$$

where $f_R$, $g_R$, $f_{R_{\mathrm{d}}}$, and $g_{R_{\mathrm{d}}}$ can be obtained—through lengthy calculation—by differentiating (2) and (77).

Then, we design the remaining input $P$ by enforcing the condition of exponential stability using CLF theory [54]

$$\dot{V}(\mathbf{x}, t, \mathbf{u}) + \lambda V(\mathbf{x}, t) = a_P(\mathbf{x}, t) + b_P(\mathbf{x}, t)P \leq 0 \tag{81}$$

where, through substituting (78)–(80), the coefficients become

$$\begin{aligned} a_P(\mathbf{x}, t) = &-\frac{1}{2}\|\mathbf{v}_{\mathrm{c}}(\mathbf{r}, t) - \mathbf{v}(\boldsymbol{\zeta})\|^2_{\mathbf{K}_{\mathbf{v}}} \\ &+ (\mathbf{v}_{\mathrm{c}}(\mathbf{r}, t) - \mathbf{v}(\boldsymbol{\zeta}))^{\top}\mathbf{M}_R(\boldsymbol{\xi}, \boldsymbol{\eta})(R_{\mathrm{d}} - R) \\ &+ \frac{1}{\mu}(R_{\mathrm{d}} - R)\big(f_{R_{\mathrm{d}}}(\mathbf{x}, t) - f_R(\mathbf{x}, t)\big) \\ &+ \frac{\lambda}{2}\|\mathbf{v}_{\mathrm{c}}(\mathbf{r}, t) - \mathbf{v}(\boldsymbol{\zeta})\|^2 + \frac{\lambda}{2\mu}(R_{\mathrm{d}} - R)^2 \\ b_P(\mathbf{x}, t) = &\frac{1}{\mu}(R_{\mathrm{d}} - R)\big(g_{R_{\mathrm{d}}}(\mathbf{x}, t) - g_R(\mathbf{x}, t)\big). \end{aligned} \tag{82}$$

Here, $\lambda > 0$ is chosen such that it is smaller than or equal to the eigenvalues of $\mathbf{K}_{\mathbf{v}}$ as suggested above.

Thus, $P$ can be designed, for example, by the following CLF quadratic program:

$$\begin{aligned} P = \underset{\hat{P} \in \mathbb{R}}{\operatorname{argmin}}\ & \hat{P}^2 \\ \text{s.t. } & a_P(\mathbf{x}, t) + b_P(\mathbf{x}, t)\hat{P} \leq 0 \end{aligned} \tag{83}$$

see (18), which has the explicit solution

$$P = \begin{cases} 0, & \text{if } b_P(\mathbf{x}, t) = 0 \\ \dfrac{\min\{0, -a_P(\mathbf{x}, t)\}}{b_P(\mathbf{x}, t)}, & \text{if } b_P(\mathbf{x}, t) \neq 0 \end{cases} \tag{84}$$

see (21) and (22). Notice that backstepping ensures that this optimization problem is feasible, i.e., there exists $P$ that satisfies (81). Namely, even when $R = R_{\mathrm{d}}$, which means that $b_P(\mathbf{x}, t) = 0$ and $P$ drops from (81), the inequality still holds because $a_P(\mathbf{x}, t) \leq 0$ with the above choice of $\lambda$ based on (76) and (82). Ultimately, with the backstepping procedure, (79) and (81) provide exponential stability and also ensure that (50) and (51) hold for $\mathbf{v}_{\mathrm{c}}(\mathbf{r}, t) = \mathbf{v}_s(\mathbf{r}, t)$.

## REFERENCES


[1] K. L. Hobbs, M. L. Mote, M. C. L. Abate, S. D. Coogan, and E. M. Feron, "Runtime assurance for safety-critical systems: An introduction to safety filtering approaches for complex control systems," *IEEE Control Syst. Mag.*, vol. 43, no. 2, pp. 28–65, Apr. 2023.

[2] J. G. Fuller, "Run-time assurance: A rising technology," in *Proc. AIAA/IEEE 39th Digit. Avionics Syst. Conf. (DASC)*, Oct. 2020, pp. 1–9.

[3] P. Nagarajan, S. K. Kannan, C. Torens, M. E. Vukas, and G. F. Wilber, "ASTM F3269—An industry standard on run time assurance for aircraft systems," in *Proc. AIAA Scitech Forum*, Jan. 2021, p. 0525.

[4] S. Chaki and D. de Niz, "Certifiable runtime assurance of distributed real-time systems," in *Proc. AIAA Inf. Syst.-AIAA Infotech Aerosp.*, Jan. 2017, p. 0561.

[5] J. D. Schierman, M. D. DeVore, N. D. Richards, and M. A. Clark, "Runtime assurance for autonomous aerospace systems," *J. Guid., Control, Dyn.*, vol. 43, no. 12, pp. 2205–2217, Dec. 2020.

[6] D. Cofer et al., "Run-time assurance for learning-enabled systems," in *NASA Formal Methods*, R. Lee, S. Jha, A. Mavridou, and D. Giannakopoulou, Eds., Cham, Switzerland: Springer, 2020, pp. 361–368.

[7] D. Costello and H. Xu, "Using a run time assurance approach for certifying autonomy within naval aviation," *Syst. Eng.*, vol. 26, no. 3, pp. 271–278, May 2023.

[8] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, Jun. 1985, pp. 500–505.

[9] Y.-B. Chen, G.-C. Luo, Y.-S. Mei, J.-Q. Yu, and X.-L. Su, "UAV path planning using artificial potential field method updated by optimal control theory," *Int. J. Syst. Sci.*, vol. 47, no. 6, pp. 1407–1420, Apr. 2016.

[10] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *Int. J. Robot. Res.*, vol. 17, no. 7, pp. 760–772, Jul. 1998.

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

[12] Y. Shoukry, P. Tabuada, S. Tsuei, M. B. Milam, J. W. Grizzle, and A. D. Ames, "Closed-form controlled invariant sets for pedestrian avoidance," in *Proc. Amer. Control Conf. (ACC)*, May 2017, pp. 1622–1628.

[13] U. Ravaioli, J. Cunningham, J. McCarroll, V. Gangal, K. Dunlap, and K. Hobbs, "Safe reinforcement learning benchmark environments for aerospace control systems," in *Proc. IEEE Aerosp. Conf.*, Jul. 2022, pp. 1–20.

[14] K. L. Hobbs et al., "Systems theoretic process analysis of a run time assured neural network control system," in *Proc. AIAA SCITECH Forum*, Jan. 2023, p. 2664.

[15] T. Gurriet, M. Mote, A. Singletary, P. Nilsson, E. Feron, and A. D. Ames, "A scalable safety critical control framework for nonlinear systems," *IEEE Access*, vol. 8, pp. 187249–187275, 2020.

[16] A. Singletary, K. Klingebiel, J. R. Bourne, A. C. Browning, P. Tokumaru, and A. D. Ames, "Comparative analysis of control barrier functions and artificial potential fields for obstacle avoidance," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2021, pp. 8129–8136.

[17] A. Haraldsen, M. S. Wiig, A. D. Ames, and K. Y. Pattersen, "Safety-critical control of nonholonomic vehicles in dynamic environments using velocity obstacles," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2024, pp. 3152–3159.

[18] M. Tayal, R. Singh, J. Keshavan, and S. Kolathaya, "Control barrier functions in dynamic UAVs for kinematic obstacle avoidance: A collision cone approach," in *Proc. Amer. Control Conf. (ACC)*, vol. 62, Jul. 2024, pp. 3722–3727.

[19] C. Llanes and S. Coogan, "A ROS package for UAV run time assurance with in-the-loop reachability," in *Proc. AIAA SCITECH Forum*, Jan. 2023, p. 0879.

[20] W. Luo and A. Kapoor, "Airborne collision avoidance systems with probabilistic safety barrier certificates," in *Proc. Workshop Saf. Robustness Decis.-Making (NeurIPS)*, 2019, pp. 1–11.

[21] E. Scukins and P. Ögren, "Using reinforcement learning to create control barrier functions for explicit risk mitigation in adversarial environments," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 10734–10740.

[22] E. Squires, R. Konda, S. Coogan, and M. Egerstedt, "Model free barrier functions via implicit evading maneuvers," 2021, *arXiv:2107.12871*.

[23] E. Squires, P. Pierpaoli, R. Konda, S. Coogan, and M. Egerstedt, "Composition of safety constraints for fixed-wing collision avoidance amidst limited communications," *J. Guid., Control, Dyn.*, vol. 45, no. 4, pp. 714–725, Apr. 2022.

[24] H. Zhou, Z. Zheng, Z. Guan, and Y. Ma, "Control barrier function based nonlinear controller for automatic carrier landing," in *Proc. 16th Int. Conf. Control, Autom., Robot. Vis. (ICARCV)*, Dec. 2020, pp. 584–589.

[25] Y. Xu, R. Zhou, Z. Yu, F. Chen, and Y. Zhang, "Barrier Lyapunov function-based finite-time reliable trajectory tracking control of fixed-wing UAV with error constraints," *IFAC-PapersOnLine*, vol. 55, no. 6, pp. 597–602, 2022.

[26] Z. Zheng, J. Li, Z. Guan, and Z. Zuo, "Constrained moving path following control for UAV with robust control barrier function," *IEEE/CAA J. Autom. Sinica*, vol. 10, no. 7, pp. 1557–1570, Jul. 2023.

[27] A. Ghaffari, "Analytical design and experimental verification of geofencing control for aerial applications," *IEEE/ASME Trans. Mechatronics*, vol. 26, no. 2, pp. 1106–1117, Apr. 2021.

[28] A. Singletary, A. Swann, Y. Chen, and A. D. Ames, "Onboard safety guarantees for racing drones: High-speed geofencing with control barrier functions," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 2897–2904, Apr. 2022.

[29] F. Corraro, G. Corraro, G. Cuciniello, and L. Garbarino, "Unmanned aircraft collision detection and avoidance for dealing with multiple hazards," *Aerospace*, vol. 9, no. 4, p. 190, Apr. 2022.

[30] M. H. Cohen, T. G. Molnar, and A. D. Ames, "Safety-critical control for autonomous systems: Control barrier functions via reduced-order models," *Annu. Rev. Control*, vol. 57, Jun. 2024, Art. no. 100947.

[31] I. Lugo-Cárdenas, G. Flores, S. Salazar, and R. Lozano, "Dubins path generation for a fixed wing UAV," in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, May 2014, pp. 339–346.

[32] M. Owen, R. W. Beard, and T. W. McLain, "Implementing Dubins airplane paths on fixed-wing UAVs," in *Handbook of Unmanned Aerial Vehicles*, K. P. Valavanis and G. J. Vachtsevanos, Eds., Dordrecht, The Netherlands: Springer, 2015, pp. 1677–1701.

[33] B. L. Stevens, F. L. Lewis, and E. N. Johnson, *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*, 3rd ed., Hoboken, NJ, USA: Wiley, 2016.

[34] A. D. Ames, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs with application to adaptive cruise control," in *Proc. 53rd IEEE Conf. Decis. Control*, Jun. 2014, pp. 6271–6278.

[35] Q. Nguyen and K. Sreenath, "Exponential control barrier functions for enforcing high relative-degree safety-critical constraints," in *Proc. Amer. Control Conf. (ACC)*, 2016, pp. 322–328.

[36] W. Xiao and C. Belta, "Control barrier functions for systems with high relative degree," in *Proc. IEEE 58th Conf. Decis. Control (CDC)*, Dec. 2019, pp. 474–479.

[37] A. J. Taylor, P. Ong, T. G. Molnar, and A. D. Ames, "Safe backstepping with control barrier functions," in *Proc. Conf. Decis. Control*, 2022, pp. 5775–5782.

[38] D. R. Agrawal and D. Panagou, "Safe control synthesis via input constrained control barrier functions," in *Proc. 60th IEEE Conf. Decis. Control (CDC)*, Dec. 2021, pp. 6113–6118.

[39] A. D. Ames, G. Notomista, Y. Wardi, and M. Egerstedt, "Integral control barrier functions for dynamically defined control laws," *IEEE Control Syst. Lett.*, vol. 5, no. 3, pp. 887–892, Jul. 2021.

[40] S. Liu, J. Dolan, and C. Liu, "Safe control under input saturation with neural control barrier functions," in *Proc. 6th Annu. Conf. Robot Learn.*, 2022, pp. 1–11.

[41] W. Xiao, C. A. Belta, and C. G. Cassandras, "Sufficient conditions for feasibility of optimal control problems using control barrier functions," *Automatica*, vol. 135, Jan. 2022, Art. no. 109960.

[42] M. H. Cohen, P. Ong, G. Bahati, and A. D. Ames, "Characterizing smooth safety filters via the implicit function theorem," *IEEE Control Syst. Lett.*, vol. 7, pp. 3890–3895, 2023.

[43] T. G. Molnar and A. D. Ames, "Composing control barrier functions for complex safety specifications," *IEEE Control Syst. Lett.*, vol. 7, pp. 3615–3620, 2023.

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

[45] X. Xu, "Constrained control of input–output linearizable systems using control sharing barrier functions," *Automatica*, vol. 87, pp. 195–201, Jan. 2018.

[46] A. Katriniok, "Control-sharing control barrier functions for intersection automation under input constraints," in *Proc. Eur. Control Conf. (ECC)*, Jul. 2022, pp. 1–7.

[47] W. Xiao, C. G. Cassandras, C. A. Belta, and D. Rus, "Control barrier functions for systems with multiple control inputs," in *Proc. Amer. Control Conf. (ACC)*, Jun. 2022, pp. 2221–2226.

[48] T. G. Molnar, R. K. Cosner, A. W. Singletary, W. Ubellacker, and A. D. Ames, "Model-free safety-critical control for robotic systems," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 944–951, Apr. 2022.

[49] S. Kolathaya and A. D. Ames, "Input-to-state safety with control barrier functions," *IEEE Contr. Syst. Lett.*, vol. 3, no. 1, pp. 108–113, Jan. 2019.

[50] A. Alan, A. J. Taylor, C. R. He, G. Orosz, and A. D. Ames, "Safe controller synthesis with tunable input-to-state safe control barrier functions," *IEEE Contr. Syst. Lett.*, vol. 6, pp. 908–913, 2022.

[51] M. E. Knapp, T. Berger, M. Tischler, and M. C. Cotting, "Development of a full envelope flight identified F-16 simulation model," in *Proc. AIAA Atmos. Flight Mech. Conf.*, Jan. 2018, p. 0525.

[52] *DOD Artificial Intelligence Agents Successfully Pilot Fighter Jet*. Accessed: Feb, 26, 2024. [Online]. Available: https://www.afrl.af.mil/News/Article-Display/Article/3297364/dod-artificial-intelligence-agents-successfully-pilot-fighter-jet

[53] R. A. Freeman and P. V. Kokotović, "Backstepping design of robust controllers for a class of nonlinear systems," *IFAC Proc. Vol.*, vol. 25, no. 13, pp. 431–436, Jun. 1992.

[54] H. Khalil, *Nonlinear Systems*, 3rd ed., Upper Saddle River, NJ, USA: Prentice-Hall, 2002.

**Tamas G. Molnar** (Member, IEEE) received the B.Sc. degree in mechatronics engineering and the M.Sc. and Ph.D. degrees in mechanical engineering from Budapest University of Technology and Economics, Budapest, Hungary, in 2013, 2015, and 2018, respectively.

He held postdoctoral positions at the University of Michigan, Ann Arbor, MI, USA, from 2018 to 2020, and California Institute of Technology, Pasadena, CA, USA, from 2020 to 2023. He has been an Assistant Professor of mechanical engineering with Wichita State University, Wichita, KS, USA, since 2023. His research interests include nonlinear dynamics and control, safety-critical control, and time delay systems with applications to connected automated vehicles, robotic systems, and autonomous systems.

**Suresh K. Kannan** received the Ph.D. degree in aerospace engineering from Georgia Institute of Technology, Atlanta, GA, USA, in 2005.

From 2000 to 2005, he developed adaptive neural network-based controllers for DARPA. The real-time adaptation during flight and associated hedging allowed DARPA to safely fly multiple advanced control and planning algorithms. This autonomy stack continues to fly today in academia, commercial drones, and in 2014 on a manned helicopter. From 2011 to 2017, his work focused on developing vision-based navigation and collision-free trajectories with verifiable guarantees. From 2014 to 2019, under the DARPA ALIAS Program for Sikorsky, he developed a formal language that enables software and human agents to interact verifiably and safely. He led the Autonomy Group, United Technologies Research Center (now Raytheon). He is currently the Chief Scientist with Nodein Autonomy Corporation, Farmington, CT, USA. In other areas, he helped write the recent 2021 revision of the ASTM F3269 standard for run-time assurance. His singular focus is the practical use of formal mathematics to aid the development of reliable and certifiable algorithms for autonomous aircraft and cars.

**James Cunningham** received the B.S. and M.S. degrees in computer science and engineering from The Ohio State University, Columbus, OH, USA.

He was primarily working with the Safe Autonomy Team, Autonomy Capability Team (ACT3), Air Force Research Laboratory, Wright-Patterson AFB, OH, USA, studying approaches toward safety in autonomous systems that use learning-based control methods. He is currently an AI Scientist with Parallax Advanced Research, Beavercreek, OH, USA. His previous experience includes work in efficient parameter-free online clustering, multidomain learning-based data embedding and retrieval, automatic supervised large-scale dataset generation, autonomous tracking in wide-area motion imagery, and discriminatory algorithms in the SAR domain.

**Kyle Dunlap** received the B.S. and M.S. degrees in aerospace engineering from the University of Cincinnati, Cincinnati, OH, USA.

He was primarily working with the Safe Autonomy Team, Autonomy Capability Team (ACT3), Air Force Research Laboratory, Wright-Patterson AFB, OH, USA, where he investigates real-time safety assurance methods for intelligent aerospace control systems. He is currently an AI Scientist with Parallax Advanced Research, Beavercreek, OH, USA. His previous experience includes developing a universal framework for run-time assurance (RTA) and comparing different RTA approaches during reinforcement learning training.

**Kerianne L. Hobbs** received the B.S. degree in aerospace engineering from Embry-Riddle Aeronautical University, Daytona Beach, FL, USA, the M.S. degree in astronautical engineering from the Air Force Institute of Technology, Wright-Patterson AFB, OH, USA, and the Ph.D. degree in aerospace engineering from Georgia Institute of Technology, Atlanta, GA, USA.

She is currently the Safe Autonomy and Space Lead of the Autonomy Capability Team (ACT3), Air Force Research Laboratory, Wright-Patterson AFB, OH, USA, where she investigates rigorous specification, analysis, bounding, and intervention techniques to enable safe, trusted, ethical, and certifiable autonomous and learning controllers for aircraft and spacecraft applications. Her previous experience includes work in automatic collision avoidance and autonomy verification and validation research.

Dr. Hobbs was selected for the 2024 AIAA Associate Fellow Class and 2020 AFCEA 40 Under 40 Award. She was a member of the team that won the 2018 Collier Trophy (Automatic Ground Collision Avoidance System Team).

**Aaron D. Ames** (Fellow, IEEE) received the B.S. degree in mechanical engineering and the B.A. degree in mathematics from the University of St. Thomas, in 2001, and the M.A. degree in mathematics and the Ph.D. degree in electrical engineering and computer sciences from the University of California at Berkeley (UC Berkeley), Berkeley, CA, USA, in 2006.

He served as a Post-Doctoral Scholar of control and dynamical systems with California Institute of Technology (Caltech), Pasadena, CA, USA, from 2006 to 2008, and began his faculty career at Texas A&M University, College Station, TX, USA, in 2008. He was an Associate Professor with the Woodruff School of Mechanical Engineering and the School of Electrical and Computer Engineering, Georgia Tech, Atlanta, GA, USA. He joined Caltech, in 2017, where he is currently the Bren Professor of mechanical and civil engineering and control and dynamical systems. His research interests span the areas of robotics, nonlinear, safety-critical control, and hybrid systems, with a special focus on applications to dynamic robots—both formally and through experimental validation.

Dr. Ames was a recipient of the 2005 Leon O. Chua Award for achievement in nonlinear science and the 2006 Bernard Friedman Memorial Prize in Applied Mathematics at UC Berkeley, the NSF CAREER Award in 2010, the 2015 Donald P. Eckman Award, and the 2019 IEEE CSS Antonio Ruberti Young Researcher Prize.