Unified MPC+CBF Control for Performant Safety: Mutual Benefits and Inherent Robustness Properties

Ryan K. Cosner, *Student Member,* Ryan M. Bena, *Member,* Aaron D. Ames *Fellow*

***Abstract*—Safety is essential for successful real-world deployment of robotic systems. During operation these systems must operate safely while also prioritizing their performance goals, ideally achieving high performance behavior simultaneously alongside mathematically verifiable safety guarantees. In this work we study the combination of two predominant control techniques, model predictive control (MPC) and control barrier function (CBF) based safety filters, both of which seek to enforce safety constraints on the system dynamics while minimally deviating from a performance objective. By combining the cost function and horizon-based planning of MPC with the CBF-based safety constraint we see both practical and theoretical benefits in nominal operation, operation under bounded uncertainty and operation under stochastic (and potentially unbounded) uncertainty, that extend beyond the capabilities of either of the individual methods. In this work we show that the combined MPC and CBF (MPC+DCBF) controller displays favorable safety, performance, and closed-loop feasibility properties, and we demonstrate the utility of this unified controller on quadrupedal and quadrotor robots performing dynamic obstacle avoidance tasks. A video of these hardware demonstrations can be found at: https://shorturl.at/fQ7BW.**

I. Introduction

SAFETY is a fundamental requirement for most real-world robotic systems spanning a wide array of modern application domains including autonomous vehicles, assistive devices, and medical and industrial robotics [1]. The safety-critical nature of these new and growing use cases mandates that dynamic safety be rigorously encoded in the controller design while practical utility requires that robots satisfy safety requirements with minimal cost to their performance goals.

A. Safety-Critical Control Methodologies

To make formal guarantees of system safety, we must first provide a mathematically rigorous definition of "safety". To this end, safety is often encoded in robotics and control theory as the forward invariance of a user-defined "safe-set" [2]–[4]. Several control methods been have developed for guaranteeing safety in this form including Control Barrier Functions (CBFs) [5], backwards Hamilton-Jacobi (HJ) reachability [4], and state-constrained model predictive control (MPC) [3]. While HJ methods provide strong guarantees of optimality and safety, they often have limited applications to high-dimensional and/or nonlinear systems due to their computational complexity. Alternatively, despite stronger theoretical assumptions required for their implementation, widespread experimental success has been achieved for both MPC [6], [7] and CBF-based [8], [9] methods by enforcing computationally simple safety constraints while optimizing for tractable proxies for performance.

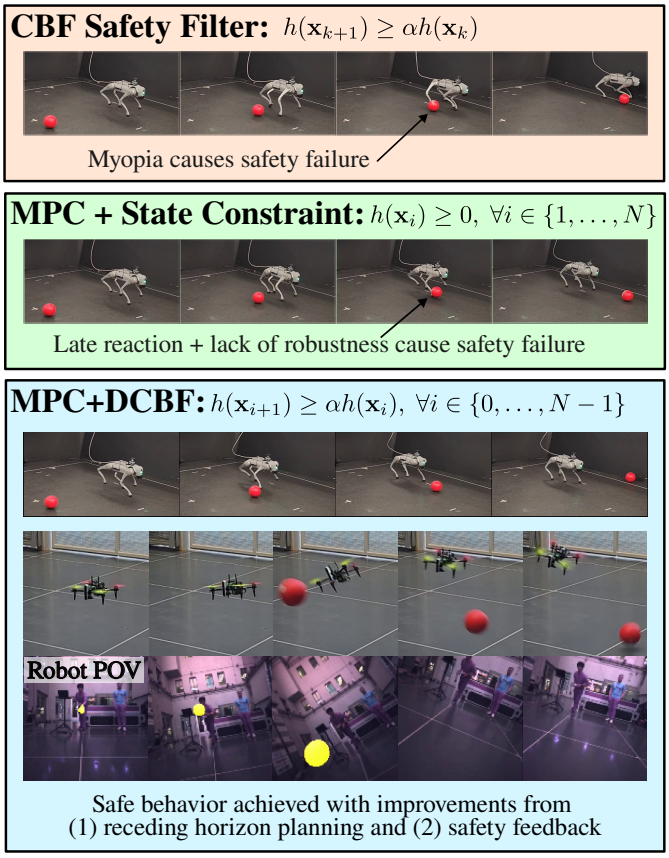


Fig. 1. In their standard implementations, DCBF-based safety filters and MPC controllers with state constraints suffer from myopia and fragility, respectively. We solve these problems by combining these two methods. This figure shows three controllers performing dynamic obstacle avoidance. **(Top)** The DCBF safety filter fails because the pointwise optimal safe action does not plan around the obstacle. **(Middle)** The MPC controller with state constraints fails because it does not account for model uncertainty and reacts too late. **(Bottom)** The MPC+DCBF controller achieves safety and benefits from the performance advantages of the MPC horizon and the inherent robustness properties of the DCBF. For the quadruped experiments the obstacle state estimation is performed off-board using an overhead camera and for the quadrotor experiments it is performed on-board using and RGBd camera and the first-person-view (FPV) masked image used for state estimation is shown. A video of these experiments can be found at the link in [10].

In particular, MPC-based methods generally utilize a discrete-time model of the system to approximate the optimal control problem over a finite time horizon [3]. In a receding horizon fashion, they apply a subset of their planned input

This research is supported in part by BP p.l.c.

R. K. Cosner, R. Bena, and A. D. Ames are with the Department of Mechanical and Civil Engineering at the California Institute of Technology, Pasadena, CA, USA. Email {rkcosner, bena, ames}@caltech.edu.

0000–0000/00$00.00 © 2021 IEEE

sequence before generating a new plan by re-solving the finite-time optimal control problem (FTOCP) again at the new time. Here, state constraints are used to encode both the system dynamics and safety requirements which must be enforced while solving the FTOCP that require each state in the planned horizon to satisfy the safety requirement. While these methods have achieved great success, they can suffer from infeasibility during closed-loop application if the strong assumptions used to guarantee recursive feasibility do not hold [11].

Alternatively, CBF-based safety filters generally take the form of a constraint on the change (e.g. derivative or finite difference) of the system's safety encoded as a scalar value [12]. In this work we consider the discrete-time reformulation of the more common continuous-time CBF constraint. The discrete-time control barrier function (DCBF) utilizes the discrete-time dynamics model to enforce safety at each discrete update of the system dynamics [13], similar to MPC. DCBFs differ from the MPC state constraint in that they establish a feedback relationship for safety that modulates future safety values based on current ones. This often creates a stricter constraint [14] that provides additional robustness properties [5], [15]. In their application, continuous-time CBF and DCBF-based safety filters generally assume the existence of a nominal controller that achieves performance goals, but may do so in an unsafe fashion. These techniques then filter this controller to enforce the safety requirement while simultaneously attempting to achieve performance goals by minimally modifying the nominal controller in a pointwise fashion [16]. Alternatively, they are occasionally enforced alongside a slackened control Lyapunov constraint, in which case, the safety constraint is enforced strictly and the convergence-based performance constraint is enforced with a slack variable to achieve performance as long as it does not conflict with safety. While myopic, pointwise optimal methods like CBFs[1] provide safety guarantees, they often create undesirable equilibrium points that render the system incapable of achieving its performance goals [18]–[20].

B. Robust Safety Considerations

Regardless of control methodology, as robots venture into the real world, they will be faced with increasing uncertainties stemming from imperfect perception, inaccurate world models, approximate dynamics, and other random disturbances. These error sources are often key causes of failure in the deployment of real-world robotics, and can undermine safety and performance guarantees that rely on perfect models of the robotic system and its environment. Along with this increasing uncertainty and undermined guarantees, these real-world applications place increasing importance on the safety of these systems. Thus, we seek robust control methods to ensure the safety under real-world uncertainty.

Significant achievements have been made to enhance the robustness of MPC and CBF-based methods, often by adopting a worst-case, adversarial model of uncertainties [5], [21]–[24]. While these methods provide strong guarantees of safety, they often do so at the cost of performance, since adversarial uncertainties are uncommon in practice. As an alternative to these conservative, worst-case robustness techniques, stochastic methods present a natural framework for generating risk-sensitive safety guarantees when statistics of the uncertainty distribution are known [15], [25]–[32]. These types of guarantees allow for desirable behavior to be found that balances safety risk with potential performance. Although they do not generally provide risk-free guarantees, these methods do allow for smooth degradation of safety via variable risk-aware levels of conservatism. Additionally, despite their utility, these methods are often computationally complex, preventing their widespread application to real-world robotics.

[1]Artificial potential fields (APFs) have been shown to be a special case of DCBFs [17] and display the same myopic failure modes.

C. Overview and Contributions of this Work

In this work we analyze the mutual benefit and inherent robustness properties obtained by combining the horizon-based optimization of MPC methods with the decay-based safety constraint of DCBFs. In Section II we provide mathematical preliminaries that define safety, our problem formulation, and the standard MPC and DCBF methodologies. The main body of the work can then be divided into deterministic analysis and experiments in Sections III, IV, and V and probabilistic analysis and experiments in Sections VI, VII, and VIII. Specifically, Section III presents the combined MPC+DCBF controller [33] and novel analysis and examples of its improved recursive feasibility guarantees and performance. Section IV considers bounded additive uncertainty and demonstrates the improved robustness properties of the MPC+DCBF controller. Section V provides a demonstration of this method on a quadrupedal robot performing dynamic obstacle avoidance. Next, Section VI generalizes the discussion of robustness to consider probabilistic uncertainties and shows the inherent benefits of the MPC+DCBF control paradigm. Section VII extends this paradigm to include state uncertainties and provides theoretical safety guarantees in that context. Section VIII provides a demonstration of this method on a quadrotor robot performing dynamical obstacle avoidance with onboard, vision-based state estimation of the obstacle. Finally, Section IX provides concluding remarks and a discussion of this methods limitations and avenues for future work.

In total, the contributions of this work are theoretical and practical demonstrations of the mutual benefits and inherent robustness properties of the unified MPC+DCBF controller. In particular, we achieve improved guarantees of closed-loop feasibility (despite reduced pointwise feasibility), amelioration of undesirable stable equilibrium points, proofs of inherent deterministic robustness properties that do not require *a priori* understandings of the disturbance bound, and risk-based guarantees naturally arising from the MPC+DCBF framework that extend beyond existing work to consider state uncertainty. Furthermore, we provide demonstrations of the control methodology running on quadrupedal and quadrotor robots performing dynamic obstacle avoidance tasks.

D. Related Work

Combinations of MPC and CBF methods have been considered in many formulations including: hierarchical multi-rate

approaches [7], [34] and learning-based combinations [35]. Our discussion aligns closest with the combinations presented in [14], [36]–[38] which apply the DCBF condition as a constraint in the MPC's FTOCP. We significantly extend these works by considering the *closed-loop* feasibility improvements and robustness properties (both worst-case and probabilistic) of the MPC+DCBF controller.

The robust safety guarantees achieved in this work are related to robust control methods in the MPC literature such as tube MPC [23], [24], but rely on a proof method from continuous-time CBFs [5] that does not require an *a priori* knowledge of the disturbance size and provides a desirably smooth degradation to safety instead of catastrophic failures as the disturbance size grows. Similarly, the stochastic safety guarantees diverge significantly from the standard quantile-based methods of the MPC literature [30], [39], [40] and leverage the safety-decay property of the DCBF constraint to create simpler-to-enforce martingale-based guarantees as in [15], [29], [41], [42], but which extends their utility to scenarios with state uncertainty and improves their performance through the addition of a receding planning horizon.

The experimental demonstrations of quadrotor-based dynamic obstacle avoidance provided in this paper are similar to previous learning + event cameras [43], artificial potential fields + event cameras [44], and MPC + motion capture [45] works on collision avoidance, but with onboard RGBd camera-based obstacle detection and improved theoretical safety guarantees and horizon-based performance optimization.

II. Preliminaries

In this work we consider discrete time systems of the form:

$$\mathbf{x}_{k+1} = \mathbf{F}(\mathbf{x}_k, \mathbf{u}_k), \qquad \forall k \in \mathbb{N}, \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^{n_x}$, $\mathbf{u} \in \mathcal{U} \subset \mathbb{R}^{n_u}$, and $\mathbf{F} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$ are the state, input, and dynamics of the system, respectively. Here we consider the case where the input set $\mathcal{U}$ may be bounded.

By selecting a state-feedback controller $\boldsymbol{\pi} : \mathbb{R}^{n_x} \to \mathcal{U}$, we can then modify (1) to define the closed-loop system:

$$\mathbf{x}_{k+1} = \mathbf{F}(\mathbf{x}_k, \boldsymbol{\pi}(\mathbf{x}_k)), \qquad \forall k \in \mathbb{N}. \tag{2}$$

Our goal for this paper is to select controllers, $\boldsymbol{\pi}$, that can be used to guarantee the safety of the closed-loop system (2).

A. Safety as Forward Invariance

In order to guarantee safety, we must first formalize our notion of safety. To do this we consider a user-defined *safe set* $\mathcal{C} \subset \mathbb{R}^{n_x}$ that the system must remain within. In particular, we consider the case where the safe set $\mathcal{C}$ is defined as the 0-superlevel set of some function[2] $h : \mathbb{R}^{n_x} \to \mathbb{R}$:

$$\mathcal{C} \triangleq \{\mathbf{x} \in \mathbb{R}^{n_x} \mid h(\mathbf{x}) \geq 0\}. \tag{3}$$

[2]While constructing safety requirements using semantic information about the system and its environment is an interesting and critical step, we assume that the safety requirements are given by the user as is common [2], [13], and we leave generalized construction of safe requirements to future work.

Given this user-defined safe set, we define safety for our system as the discrete-time forward invariance[3] of $\mathcal{C}$:

Definition 1 (Forward Invariance and Safety)**.** *A set $\mathcal{C} \subset \mathcal{X}$ is forward invariant for the closed-loop system* (2) *if $\mathbf{x}_0 \in \mathcal{C}$, implies that $\mathbf{x}_k \in \mathcal{C}$ for all $k \geq 0$. Additionally, we say that the closed-loop system* (2) *is "safe" with respect to $\mathcal{C}$ if $\mathcal{C}$ is forward invariant.*

Equipped with this definition of safety, we now explore model predictive control (MPC) with safety enforced via state constraints and discrete-time control barrier function (DCBF) safety filters as two important classes of methods for achieving safety in this form.

B. Model Predictive Control with State Cosntraints

MPC is a a control methodology which leverages a model-based prediction of the system dynamics along a finite-horizon to compute control actions. In MPC, at each time-step k the controller plans a sequence of open-loop control actions to minimize a cost function, and then the first action is applied to the system. The plan of actions is then recalculated using the updated state, and the new first control action is applied, creating a state-feedback controller.

In MPC, to calculate the plan of control actions, the following discrete, finite-time optimal control problem (FTOCP) is solved at each time-step k:

$$\begin{aligned} \min_{\substack{\boldsymbol{\xi}_{0:N} \in \mathbb{R}^{n_x} \\ \boldsymbol{\nu}_{0:N-1} \in \mathbb{R}^{n_u}}} \quad & \sum_{i=0}^{N-1} c(\boldsymbol{\xi}_i, \boldsymbol{\nu}_i) + V(\boldsymbol{\xi}_N) \qquad \text{(FTOCP)} \\ \text{s.t.} \quad & \boldsymbol{\xi}_{i+1} = \mathbf{F}(\boldsymbol{\xi}_i, \boldsymbol{\nu}_i), \quad \forall i \in \{0, \dots, N-1\} \\ & \boldsymbol{\xi}_i \in \mathcal{C}, \quad \boldsymbol{\nu}_i \in \mathcal{U}, \quad \forall i \in \{0, \dots, N-1\} \\ & \boldsymbol{\xi}_0 = \mathbf{x}_k, \quad \boldsymbol{\xi}_N \in \mathcal{C}_N \end{aligned}$$

where $c : \mathbb{R}^{n_x} \times \mathcal{U} \to \mathbb{R}$ is the stage cost and $V : \mathbb{R}^{n_x} \to \mathbb{R}$ is the terminal cost used to approximate the infinite-horizon optimal control problem. Here we use the variables $\boldsymbol{\xi}_i \in \mathbb{R}^{n_x}$ and $\boldsymbol{\nu}_i \in \mathbb{R}^{n_u}$ to represent the planned sequence of states and inputs given the current state $\mathbf{x}_k$, i.e. if the dynamics and the state are known exactly then using $\mathbf{u}_k = \boldsymbol{\nu}_0$ results in the plan being precisely executed so that $\mathbf{x}_{k+1} = \boldsymbol{\xi}_1$ for the $\boldsymbol{\xi}_1$ generated at $\mathbf{x}_k$.

In the FTOCP, the first constraint incorporates the discrete time model of the system (1) along the horizon of length N, the state constraint $\boldsymbol{\xi}_i \in \mathcal{C}$ (equivalently $h(\mathbf{x}) \geq 0$) ensures that each state in the plan is safe, the input constraint $\boldsymbol{\nu}_i \in \mathcal{U}$ ensures that the inputs are realizable on the system, the initial condition constraint $\boldsymbol{\xi}_0 = \mathbf{x}_k$ aligns the plan with the current state, and the terminal state constraint $\boldsymbol{\xi}_N \in \mathcal{C}_N \subset \mathcal{C}$ is used to achieve recursive feasibility of the feedback controller. In general it is assumed that $\mathcal{C}_N$ is a safe, control-invariant set for the inputs $\mathbf{u} \in \mathcal{U}$, in which case the MPC controller can be thought of as a domain-of-attraction expander for $\mathcal{C}_N$. For additional discussion of MPC and this FTOCP, please see [3].

[3]In this work we consider the safety of these systems at sample times. For systems that move continuously between sample times, please see [46] for a discussion of the sampled-data properties of these discrete-time formulations.

To generate the MPC input, the optimal plan of inputs for the FTOCP is computed as $[\boldsymbol{\nu}_0^*(\mathbf{x}_k), \ldots, \boldsymbol{\nu}_{N-1}^*(\mathbf{x}_k)]$ and then the first action is applied to the system, defining the MPC controller:

$$\boldsymbol{\pi}^{\text{MPC}}(\mathbf{x}_k) = \boldsymbol{\nu}_0^*(\mathbf{x}_k). \tag{MPC}$$

By enforcing safety in the form of a state constraint in the FTOCP, $\boldsymbol{\pi}^{\text{MPC}}(\mathbf{x})$ selects control actions which ensures the safety of the system at each discrete update of (2).

C. Control Barrier Functions

Alongside state-constrained MPC, CBFs have gained popularity as an alternative tool for achieving safety guarantees. While CBFs are more commonly studied in their continuous time form [2], in this work we focus on their discrete-time implementation as first presented in [13]. This discrete-time formulation will allow us to deploy them as a constraint in the FTOCP as first suggested in [33].

For a safe-set $\mathcal{C}$ defined as in (3), we refer to h as a DCBF if it satisfies the following definition:

Definition 2 (Discrete-time Control Barrier Function (DCBF) [13])**.** *Let $\mathcal{C}$ be the safe set given in* (3)*. A function $h : \mathbb{R}^{n_x} \to \mathbb{R}$ is a discrete-time control barrier function (DCBF) for* (1) *if there exists an $\alpha \in [0, 1]$ such that for each $\mathbf{x} \in \mathcal{C}$ there exists a $\mathbf{u} \in \mathcal{U}$ such that:*

$$h(\mathbf{F}(\mathbf{x}, \mathbf{u})) \geq \alpha h(\mathbf{x}), \tag{DCBF}$$

We note a state constraint over a horizon of length 1 is the special case of DTCBFs when $\alpha = 0$.

Intuitively, when $\alpha > 0$, the DCBF inequality requires that the system safety, as represented by $h(\mathbf{x})$, cannot decay faster than geometrically, i.e. $h(\mathbf{x}_k) \geq \alpha^k h(\mathbf{x}_0)$. Additionally, the maximum decrease in safety at each step goes to zero as the system approaches the boundary of the safe set[4]. The authors in [13] formally relate DCBFs to the safety of the closed-loop system (2) with respect to $\mathcal{C}$:

Theorem 1 (DCBF Safety [13, Prop. 4])**.** *If $h : \mathbb{R}^{n_x} \to \mathbb{R}$ in* (3) *is a DCBF for* (1)*, then any $\boldsymbol{\pi} : \mathbb{R}^{n_x} \to \mathcal{U}$ such that:*

$$h(\mathbf{F}(\mathbf{x}, \boldsymbol{\pi}(\mathbf{x}))) \geq \alpha h(\mathbf{x}), \quad \text{for all } \mathbf{x} \in \mathcal{C}, \tag{4}$$

renders the closed-loop system (2) *safe with respect to $\mathcal{C}$.*

As opposed to MPC's horizon-based cost, DCBFs are generally implemented as a safety filter [16] that takes a nominal (but potentially unsafe) controller $\boldsymbol{\pi}_{\text{nom}} : \mathbb{R}^{n_x} \to \mathcal{U}$ and modifies it using the DCBF h as:

$$\begin{aligned} \boldsymbol{\pi}^{\text{DCBF}}(\mathbf{x}) = \underset{\mathbf{u} \in \mathcal{U}}{\text{argmin}} \quad & \frac{1}{2}\|\mathbf{u} - \boldsymbol{\pi}_{\text{nom}}(\mathbf{x})\|^2 \qquad \text{(DCBF-OP)} \\ \text{s.t.} \quad & h(\mathbf{F}(\mathbf{x}, \mathbf{u})) \geq \alpha h(\mathbf{x}). \end{aligned}$$

[4]The standard continuous-time CBF condition $\frac{d}{dt}h(\mathbf{x}) \geq -\overline{\gamma}h(\mathbf{x})$ for $\overline{\gamma} > 0$ becomes $h(\mathbf{x}_{k+1}) - h(\mathbf{x}_k) \geq -\gamma h(\mathbf{x}_k)$ for $\gamma \in (0, 1)$ in discrete-time. Defining $\alpha = 1 - \gamma$ recovers the condition $h(\mathbf{x}_{k+1}) \geq \alpha h(\mathbf{x}_k)$

where the DCBF is used to minimally modify the nominal control action to achieve safety. Assuming feasibility[5], the $\boldsymbol{\pi}^{\text{DCBF}}$ controller guarantees safety for the system (1) by selecting inputs that satisfy the DCBF inequality.

As a safety filter on $\boldsymbol{\pi}_{\text{nom}}$, performance is indirectly achieved through the nominal controller. If the safety constraint and the performance objective of the nominal controller do not conflict, then the nominal controller allows the system to achieve its performance goal. However, if they do conflict, then myopic pointwise modifications are made to the nominal controller that enforce safety but may destroy the performance capabilities of the system [20].

III. Combined MPC+DCBF

The main theoretical focus of this work is to consider the safety, performance, and robustness properties of the unified controller generated from the FTOCP+DCBF problem:

$$\begin{aligned} \min_{\substack{\boldsymbol{\xi}_{0:N} \in \mathbb{R}^{n_x} \\ \boldsymbol{\nu}_{0:N-1} \in \mathbb{R}^{n_u}}} \quad & \sum_{i=0}^{N-1} c(\boldsymbol{\xi}_i, \boldsymbol{\nu}_i) + V(\boldsymbol{\xi}_N) \qquad \text{(FTOCP+DCBF)} \\ \text{s.t.} \quad & \boldsymbol{\xi}_{i+1} = \mathbf{F}(\boldsymbol{\xi}_i, \boldsymbol{\nu}_i), \quad \forall i \in \{0, \ldots, N-1\} \\ & h(\boldsymbol{\xi}_{i+1}) \geq \alpha h(\boldsymbol{\xi}_i), \quad \forall i \in \{0, \ldots, N-1\} \\ & \boldsymbol{\nu}_i \in \mathcal{U}, \quad \forall i \in \{0, \ldots, N-1\} \\ & \boldsymbol{\xi}_0 = \mathbf{x}_k \end{aligned}$$

where we replace the state constraint $\mathbf{x}_i \in \mathcal{C}$ (i.e., $h(\boldsymbol{\xi}_i) \geq 0$) with the DCBF constraint for some $\alpha \in [0, 1]$ and we remove the terminal constraint $\boldsymbol{\xi}_N \in \mathcal{C}_N$.

As with MPC, we derive a controller from the FTOCP+DCBF by using the first input of the open-loop plan:

$$\boldsymbol{\pi}^{\text{MPC+DCBF}}(\mathbf{x}_k) = \boldsymbol{\nu}_0^*(\mathbf{x}_k). \tag{MPC+DCBF}$$

A. Feasibility under the DCBF Assumption

Due to the assumption of feasibility of the DCBF inequality, the set $\mathcal{C}$ is assumed to be control invariant and we do not require a terminal constraint $\mathcal{C}_N$ to guarantee the safety of the closed-loop system (2) and recursive feasibility of the FTOCP or FTOCP+DCBF.

We formalize these guarantees in Thm. 2 and Cor. 1:

Theorem 2. *If h is a DCBF for* (1) *and $\mathbf{x}_0 \in \mathcal{C}$, then $\boldsymbol{\pi}^{\text{MPC+DCBF}}$ is recursively feasible and* (2) *is safe w.r.t. $\mathcal{C}$.*

Proof. Since the FTOCP+DCBF incorporates the true system dynamics and since the initial condition is safe, the FTOCP+DCBF problem is feasible because h satisfies Def. 2. Additionally, Thm. 1 implies the next planned state is safe. After implementing the first planned control action, the next state is safe and the same analysis applies to achieve feasibility and safety at the next step. By recursion, the MPC+DCBF

[5]If infeasible, a slack variable can be added to recover feasibility and its effect on safety can be analyzed using the ISSf framework [5]. Additionally, unlike the affine inequality constraint that arises with continuous-time CBFs [12], the optimization problem (DCBF-OP) is not necessarily convex. To ameliorate this issue, it is often assumed that $h \circ \mathbf{F}$ is concave with respect to $\mathbf{u}$ [13], [36], [47], [48], or the nonconvex problem is approximately solved through iterative convex optimization [49].

controller is feasible for all $k \geq 0$ and (2) is safe with respect to $\mathcal{C}$ for all $k \geq 0$. □

Additionally, since the DCBF definition (Def. 2) implies that $\mathcal{C}$ is a control invariant set, $\mathcal{C}$ can also be used as a terminal constraint in the standard FTOCP to guarantee safety and recursive feasibility.

Corollary 1. *If h is a DCBF for* (1) *and $\mathcal{C}_N = \mathcal{C}$, then $\boldsymbol{\pi}^{\text{MPC}}$ is recursively feasible and* (2) *is safe with respect to $\mathcal{C}$.*

Proof. Since h is a DCBF, $\mathcal{C}$ is a control invariant set for $\mathbf{u} \in \mathcal{U}$. Thus, since $\mathcal{C}_N = \mathcal{C}$, the controller $\boldsymbol{\pi}^{\text{MPC}}$ is recursively feasible and (2) is safe for all $k \geq 0$ [3, Thm. 12.1]. □

This assumption of feasibility of the DCBF constraint is a very strong assumption that may not hold in general and which is difficult to verify. When it does not hold, infeasibilities may occur in the closed-loop applications of the MPC or MPC+DCBF controllers. Before exploring the feasibility of these controllers when h is *not* assumed to be a DCBF, we first define the concept of time-to-failure which we will use to provide novel analysis showing that the DCBF tightened constraint can extend closed-loop feasibility, a perspective which drastically differs from the prior analysis on this topic [14], [37] which have previously focused on the reduction in pointwise feasibility caused by incorporating the DCBF constraint in the FTOCP.

B. Time-to-Failure

The difference between the MPC and MPC+DCBF controllers comes from the adjustment of the safety constraint from $h(\mathbf{x}_i) \geq 0$ to $h(\mathbf{x}_{i+1}) \geq \alpha h(\mathbf{x}_i)$. This creates a safety-feedback requirment that constrains the updated safety value by a function of its current value. For $\alpha > 0$ and $\mathbf{x} \in \mathcal{C}$, this constraint is tighter than the state constraint with $\alpha = 0$. In particular, it tightens this constraint so that the system not only maintains $\mathbf{x} \in \mathcal{C}$, but also maintains a safety-decay property that can be seen as a generalization of the standard time-to-collision metric for collision avoidance.

Time-to-collision, denoted as T_c, is a common safety indicator for vehicle collisions. It was originally introduced as:

> *The time that remains until a collision between two vehicles would have occurred if the collision course and speed differences were maintained.* [50]

Time-to-collision is a well studied metric to identify traffic safety [51]–[53] and generally represents a robustness margin for the system's current level of safety.

Just as vehicle collision avoidance can be analyzed as forward-invariance described using a CBF [8], we can similarly generalize time-to-collision to DCBFs[6] as:

Definition 3 (Time-to-Failure)**.** *Consider the discrete-time system* (2) *and a function $h : \mathbb{R}^{n_x} \to \mathbb{R}$. If the system satisfies $h(\mathbf{F}(\mathbf{x}, \boldsymbol{\pi}(\mathbf{x}))) \geq \alpha h(\mathbf{x})$ for some $\alpha \in [0, 1)$ and all $\mathbf{x} \in \mathcal{C}$, then the time-to-failure for this system is $T_f(\alpha) \triangleq \frac{1}{1-\alpha}$.*

[6]An analogous continuous-time version of time-to-failure can be defined for continuous-time CBFs as $T_{f,\text{CT}} = \frac{1}{\alpha}$. In this case, $T_{f,\text{CT}}$ represents the time constant of the worst-case, allowable safety decay rate $\frac{d}{dt}h(\mathbf{x}) = -\alpha h(\mathbf{x})$.

As with time-to-collision, time-to-failure is a heuristic emerging from the notion that the system safety will likely continue to decay at its current rate, e.g. if the system is initially safe (i.e., $h(\mathbf{x}_0) > 0$), then (4) bounds the change in safety $\Delta h(\mathbf{x}_0) \triangleq h(\mathbf{x}_1) - h(\mathbf{x}_0) \geq (\alpha - 1)h(\mathbf{x}_0)$. If this safety decrement $\Delta h(\mathbf{x}_0)$ is repeated iteratively, then $h(\mathbf{x}_k) \geq h(\mathbf{x}_0)(1 - (1 - \alpha)k)$. It follows that the value $h(\mathbf{x}_k)$ can be less than 0 for the first time[7] when $k = \lfloor T_f(\alpha) \rfloor + 1$.

Thus, the DCBF constraint with $\alpha > 0$ in the FTOCP +DCBF extends the horizon length over which safety is considered. While $\boldsymbol{\pi}^{\text{MPC}}$ uses the system model to determine if a safety failure will occur within N steps, $\boldsymbol{\pi}^{\text{MPC+DCBF}}$ uses a constant decay model to extend this safety prediction by an additional $\lfloor T_f(\alpha) \rfloor$ steps. In this way, $\boldsymbol{\pi}^{\text{MPC+DCBF}}$ can react earlier when system safety begins to decay.

C. Improved Feasibility Guarantees

Equipped with this definition of time-to-failure, we now return to the problem of controller feasibility when h is not a valid DCBF and the control invariance assumption does not hold. This is particularly important since synthesizing valid DCBFs can be very difficult and generally requires solving the recursive feasibility problem[8] [58]. In this case, Thm. 2 and Cor. 1 do not apply and it is possible for the MPC and MPC+DCBF controllers to become infeasible.

Since the DCBF constraint is a tightening of the standard MPC state constraint when $\mathbf{x} \in \mathcal{C}$, there are less states $\mathbf{x} \in \mathcal{C}$ for which $\boldsymbol{\pi}^{\text{MPC+DCBF}}$ is feasible. Where other works have sought to improve this pointwise feasibility by allowing α to vary [37] or by only enforcing the DCBF constraint on the first step of the horizon [14], we instead take an entirely different approach and show that the reduced pointwise feasibility can actually improve closed-loop feasibility.

To do this we consider the case where, for all $\mathbf{x} \in \mathcal{C}$ we assume that:

$$\forall \mathbf{u} \in \mathcal{U}, \quad \Delta h(\mathbf{x}, \mathbf{u}) \geq -\delta \tag{5}$$

$$\forall \zeta \in [\epsilon, \delta], \ \exists \mathbf{u} \in \mathcal{U} \text{ s.t.} \quad \Delta h(\mathbf{x}, \mathbf{u}) = -\zeta \tag{6}$$

where $\delta \geq \epsilon > 0$. The first assumption (5) captures the idea that, given bounded inputs and dynamics, the system can only decrease its safety by at most $-\delta$ within a single step. The second assumption (6) captures the idea that the system can only improve its safety degradation by so much over a single step. Notably, proving infinite horizon safety or feasibility guarantees using these assumptions is impossible so we instead seek to guarantee feasibility and safety over the longest possible finite horizon.

To this end we have the following feasibility guarantee for the $\boldsymbol{\pi}^{\text{MPC}}$ controller:

[7]Here we use the floor function $\lfloor \cdot \rfloor$ to account for the integer nature of $k \in \mathbb{N}$. The floor function returns the largest integer less than or equal to its argument.

[8]While several works have developed CBF synthesis methods for different classes of systems including hierarchical systems with tracking controllers [54], feedback-linearizable systems [55], and incorrect relative degree [56], [57], we focus on the general case when h might not be a DCBF.

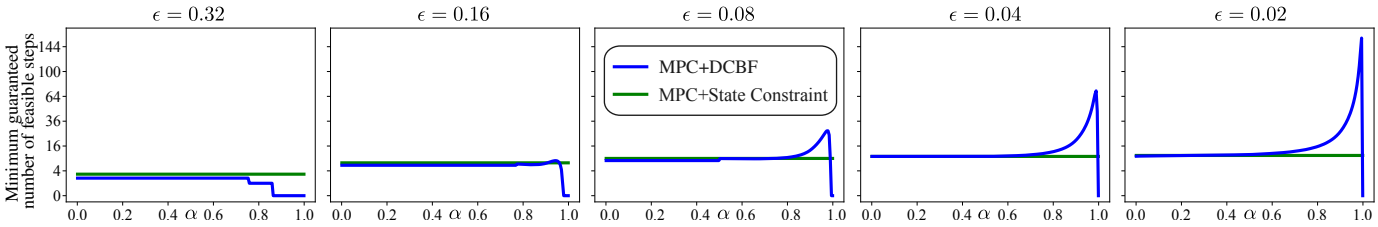


Fig. 2. Plots demonstrating the minimum number of guaranteed feasible solutions for the closed-loop application of the MPC+DCBF controller for varying α according to Prop. 1 ($\alpha = 0$) and Thm. 2 ($\alpha \in (0, 1)$). The y axis is shown using a squared scale to better capture the range of possible outputs. The minimum number of feasible steps is plotted for the state constraint (i.e., $\alpha = 0$) in green and for the MPC+DCBF controller for varying $\alpha \in (0, 1)$ in blue. The other parameters used to generate this plot include: $h(\mathbf{x}_0) = 10, \delta = 1$, and $N = 25$. The plots from left to right show the minimum number of feasible steps for varying minimum safety decay values ϵ. As the minimum safety decay value goes to zero, the minimum number of guaranteed feasible steps generated by the MPC+DCBF can dramatically outperform the state constraint-based feasibility guarantee despite the pointwise reduction in feasibility with respect to state. This is due to the closed-loop properties of applying the more conservative MPC+DCBF controller.

Proposition 1. *If the closed loop system* (2) *satisfies* (5) *and* (6)*, its initial safety is* $h(\mathbf{x}_0) > 0$*, and it attempts to enforce constraint* (4) *with* $\alpha = 0$ *over a horizon of length* $N \in \mathbb{N}$ *for* $\mathbf{u} \in \mathcal{U}$*, then the FTOCP will be feasible for all:*

$$k \leq \frac{h(\mathbf{x}_0) - N\epsilon}{\delta} + 1 \tag{7}$$

The proof of Prop. 1 is provided in Appx. A .

Next we present a closed loop feasibility guarantee for $\boldsymbol{\pi}^{\text{MPC+DCBF}}$ that can provide a *longer* guarantee of feasibility:

Theorem 3. *If the closed loop system* (2) *satisfies* (5) *and* (6)*, its initial safety is* $h(\mathbf{x}_0) > 0$*, and it attempts to enforce the constraint* (4) *for* $\alpha \in (0, 1)$ *over a horizon of length* $N \in \mathbb{N}$ *for* $\mathbf{u} \in \mathcal{U}$ *and the parameters of the system satisfy:*

$$T_f(\alpha) \triangleq \frac{1}{1-\alpha} \geq \frac{\delta + (N-2)\epsilon}{\delta + \epsilon}, \tag{8}$$

then the FTOCP+DCBF will be feasible $\forall k \leq k_\delta + k_\epsilon$ *where:*

$$k_\delta = \max\left\{ \left\lfloor \frac{h(\mathbf{x}_0)}{\delta} - T_f(\alpha) \right\rfloor, 0 \right\} \tag{9}$$

$$k_\epsilon = \log_\alpha \left(\frac{\epsilon \left(T_f(\alpha) + N - 1\right)}{h(\mathbf{x}_0) - k_\delta \delta} \right) + 1 \tag{10}$$

The proof of Thm. 2 is provided in Appx. B.

Figure 2 provides plots showing the relative feasibility guarantees of Prop. 1 and Thm. 2 for a variety of ϵ and α values. Intuitively, this figure and theorem show that, as the ability to achieve safety improves (i.e., $\epsilon \to 0$), higher α results in less pointwise feasibility, but also extends the guaranteed closed-loop feasibility, similar to how a reduction in pointwise feasibility for tube-MPC can be used to generate better recursive feasibility guarantees [59].

D. Improvements to Undesirable Equilibrium

Like infeasibility for MPC, a common deleterious property arising from the use of DCBFs is the appearance of undesirable equilibrium points which can destroy system performance by preventing the system from reaching its goal. These undesirable equilibrium points are often a result of controller continuity and topological obstructions that cause the desired control action and the DCBF-OP safety filter $\boldsymbol{\pi}^{\text{DCBF}}$ to interfere [18], [20], [60]. The MPC+DCBF controller $\boldsymbol{\pi}^{\text{MPC+DCBF}}$ reduces the impact of this problem (and potentially removes it entirely) by searching for optimal control actions which may be discontinuous with respect to space, but can performantly navigate around topological obstructions in the safe set.

We can see these undesirable stable equilibria arise in the discrete-time variant of the example from [20, Ex. 5.4]. In the following we show how the $\boldsymbol{\pi}^{\text{MPC+DCBF}}$ controller can be used to reduce the effects of these undesired equilibria by optimizing for performance metrics along a receding horizon.

Example 1. *Consider a two dimensional single integrator system with the dynamics* $\mathbf{x}_{k+1} = \mathbf{x}_k + \Delta_t \mathbf{u}_k$*, nominal controller* $\boldsymbol{\pi}_{\text{nom}}(\mathbf{x}) = -\mathbf{x}$*, and safety defined as* $h(\mathbf{x}) = -b^4 + \|\mathbf{x} - \mathbf{r}_1\|^2 \|\mathbf{x} - \mathbf{r}_2\|^2$ *for* $\alpha = e^{-1\Delta_t}$*,* $\mathbf{r}_1 = \begin{bmatrix} a & c_2 \end{bmatrix}^\top$*, and* $\mathbf{r}_2 = \begin{bmatrix} a & -c_2 \end{bmatrix}^\top$ *with* $a = 3$*,* $b = 1.05 * a$*,* $c_2 = 4$*, and* $\Delta_t = 0.05$*.*

We implement the $\boldsymbol{\pi}^{\text{DCBF}}$ *safety filter by linearizing the constraint at each step and we implement the MPC+DCBF using sequential quadratic programming where the first solution is initialized to either a semicircle on the left or the right of the obstacle depending on the sign of the* x *component of* $\mathbf{x}_0 + \boldsymbol{\rho}$ *where* $\boldsymbol{\rho} \sim \text{unif}(-0.01, 0.01)$ *is sampled from a 2D normal distribution. The results are shown in Fig. 3.*

We find that all trajectories generated by the $\boldsymbol{\pi}^{\text{MPC+DCBF}}$ *controller are safe and reach the goal location at* $(0, 0)$ *while avoiding the undesirable equilibrium point that captures many of the trajectories generated by the* $\boldsymbol{\pi}^{\text{DCBF}}$ *safety filter.*

IV. MPC+DCBF Deterministic Robustness

In the previous section, we explored the benefits to system performance and closed-loop feasibility that can be achieved by combining the standard MPC and DCBF formulations when the system dynamics are known exactly. In this section we extend this analysis to show the benefits of the unified $\boldsymbol{\pi}^{\text{MPC+DCBF}}$ controller under bounded dynamics uncertainty.

To perform this analysis, we now consider the discrete-time dynamical system (1) subject to additive dynamics uncertainty:

$$\mathbf{x}_{k+1} = \mathbf{F}(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{d}_k, \quad \forall k \in \mathbb{N}. \tag{11}$$

This uncertainty is represented by $\mathbf{d}_k$ which we assume is bounded by some $\overline{\delta} \geq \|\mathbf{d}_k\| \geq 0$ for all $k \geq 0$.

As with the undisturbed system (1), a controller can be added to generate the closed loop dynamical system:

$$\mathbf{x}_{k+1} = \mathbf{F}(\mathbf{x}_k, \boldsymbol{\pi}(\mathbf{x}_k)) + \mathbf{d}_k, \quad \forall k \in \mathbb{N}. \tag{12}$$

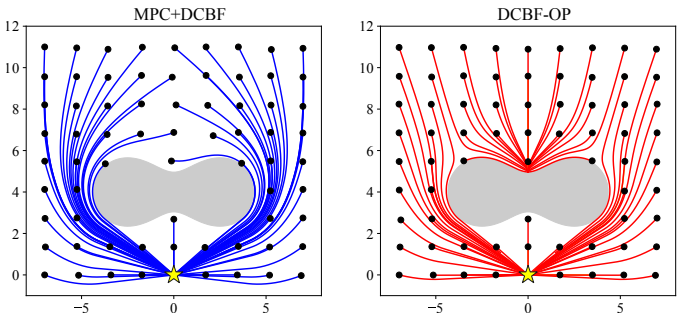


Fig. 3. Closed-loop trajectories from several initial conditions (black dots) for the system in Ex. 1 using $\boldsymbol{\pi}^{\text{MPC+DCBF}}$ (blue) and using $\boldsymbol{\pi}^{\text{DCBF}}$ (red). Here we see that $\boldsymbol{\pi}^{\text{MPC+DCBF}}$ is far better at avoiding the undesired equilibrium point and reaching the goal at (0,0) shown as a yellow star.

Next we consider the robustness safety properties of this closed-loop system.

A. Input-to-State Safety

In achieving robust safety, the feedback of the current safety value used in the DCBF constraint (4) produces a useful robustness property called input-to-state safety (ISSf) [5] that is dependent on the $\alpha \in [0, 1]$ parameter. Here we present a discrete-time variant of the continuous-time ISSf property which is a generalization of the input-to-state stability (ISS) property of continuous-time stable systems [61].

Theorem 4 (Input-to-State Safety (ISSf))**.** *Consider the disturbed system* (11) *with bounded disturbance* $\|\mathbf{d}_k\| \leq \overline{\delta}$ *for some* $\overline{\delta} \geq 0$*. If the closed-loop system* (12) *satisfies the DCBF condition* (4) *for some* $\alpha \in [0, 1)$*, some Lipschitz continuous* $h : \mathbb{R}^{n_x} \to \mathbb{R}$ *with Lipschitz constant* $\mathcal{L}_h \geq 0$*, and all* $\mathbf{x} \in \mathcal{C}_d$ *defined as*

$$\mathcal{C}_d = \{\mathbf{x} \mid h(\mathbf{x}) \geq -d\}, \tag{13}$$

with $d = \mathcal{L}_h \overline{\delta} T_f(\alpha)$*, then the closed loop system* (12) *is safe with respect to the enlarged set* $\mathcal{C}_d$ *and* $h(\mathbf{x}_k) \geq \alpha^k h(\mathbf{x}_0) - \sum_{i=0}^{k-1} \alpha^i \mathcal{L}_h \overline{\delta}$ *for all* $k \geq 0$*.*

Please see [42, Prop. 2] for a proof of Thm. 4.

Thus, under a bounded additive disturbance, the DCBF condition (4) for $\alpha \in [0, 1)$ can still generate guarantees of set invariance with respect to some larger set $\mathcal{C}_d \supset \mathcal{C}$ even when the original safe set $\mathcal{C}$ is not invariant. This robustness result differs from those generated by tube MPC [23], [24] approaches since the controller design does not require an *a priori* knowledge of the disturbance size.

Given this understanding of ISSf, we now compare the robustness of the $\boldsymbol{\pi}^{\text{MPC+DCBF}}$ controller for varying $\alpha \in [0, 1)$, where $\alpha = 0$ encodes the typical MPC state constraint. Importantly, α has the following effect on the system safety depending on whether $\mathbf{x}$ is inside or outside of $\mathcal{C}$:

- $\mathbf{x} \in \mathcal{C}$, larger α results in a *tighter* constraint that bounds the maximum rate that $h(\mathbf{x})$ can decrease to 0,
- $\mathbf{x} \notin \mathcal{C}$, larger α results in a *looser* constraint that bounds the minimum rate at which $h(\mathbf{x})$ must increase to 0.

Notably, the size of the expanded safe set $\mathcal{C}_d$ in (13) increases monotonically with α. However, $\alpha > 0$ may still be desirable since it gracefully brings the system back towards

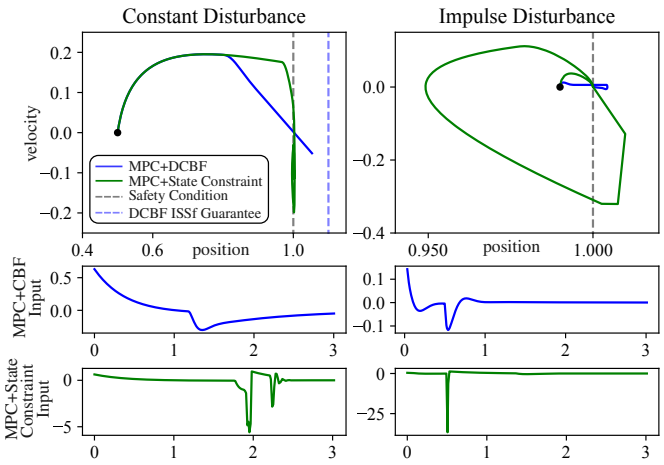


Fig. 4. Plots displaying the state (x, v_x) trajectories for the double integrator system (top), MPC+DCBF inputs (a_x) for $\alpha = e^{-\Delta_t}$ (middle) and the state constrained problem where $\alpha = 0$ (bottom). On the left, trajectories are shown for a constant additive disturbance of $\mathbf{d}_{\text{constant}} = \begin{bmatrix} 0.1\Delta_t, & 0 \end{bmatrix}^\top$ which occurs at every step with $\Delta = 10^{-3}$. In this case both trajectories cross the safety boundary (gray dashed line). The trajectory with $\alpha > 0$ has a larger violation of safety but still achieves safety of $\mathcal{C}_\delta$ whose boundary is shown as the blue dashed line. On the right, a single impulse disturbance of $\mathbf{d}_{\text{impulse}} = \begin{bmatrix} 10\Delta_t, & 0 \end{bmatrix}^\top$ occurs at $k = 500$ causing both trajectories to violate safety. Importantly, although the state constraint may result in smaller violations in some cases, it requires very large inputs, approximately an order of magnitude larger than those for the CBF, which may cause problem infeasibility when $\mathcal{U}$ is bounded.

$\mathcal{C}$ via geometric decay of $h(\mathbf{x})$ whereas $\alpha = 0$ will force the system to return to $\mathcal{C}$ in a single step, a behavior that will likely result in infeasibility or overly aggressive behavior as can be seen in the following example:

Example 2. *To demonstrate the effect of* α *on the ISSf property and the associated inputs, we consider a one-dimensional, discrete-time double integrator system with* $\Delta_t = 10^{-3}$*:*

$$\underbrace{\begin{bmatrix} x_{k+1} \\ v_{k+1} \end{bmatrix}}_{\mathbf{x}_{k+1}} = \begin{bmatrix} 1 & \Delta_t \\ 0 & 1 \end{bmatrix} \underbrace{\begin{bmatrix} x_k \\ v_k \end{bmatrix}}_{\mathbf{x}_k} + \begin{bmatrix} \frac{1}{2}\Delta_t \\ \Delta_t \end{bmatrix} \underbrace{[a_k]}_{\mathbf{u}_k}, \tag{14}$$

with a safety condition that conflicts with the goal location:

$$h(\mathbf{x}) = 1 - x, \qquad \mathbf{x}_{\text{goal}} = \begin{bmatrix} 2, & 0 \end{bmatrix}^\top. \tag{15}$$

We simulate the system with both $\alpha = 0$ *and* $\alpha = e^{-\Delta_t}$ *for both a constant disturbance and an impulsive disturbance:*

$$\mathbf{d}_{\text{constant}} = \begin{bmatrix} 0.1\Delta_t \end{bmatrix} \qquad \mathbf{d}_{\text{impulse}} = \begin{bmatrix} 10\Delta_t s \end{bmatrix} \tag{16}$$

The results of these simulations can be seen in Fig. 4.

Although $\alpha > 0$ *may lead to larger safety violations, it also results in much smoother trajectories and requires significantly smaller inputs (by approximately an order of magnitude) because constraint* (4) *enforces geometric convergence back to the set whereas the state constraint* ($\alpha = 0$) *requires the system to return to* $\mathcal{C}$ *in a single step.*

Here we find that the DCBF inequality (4) results in a smooth degradation of safety as the disturbance size increases and that, if the system becomes unsafe, the DCBF facilitates a graceful recovery of safety through its decay-based constraint. Alternatively, the step-wise safety requirement of the MPC controller requires the system to become safe again

immediately, which can lead to overly aggressive behavior and infeasible safety requirements. In practice we see this manifest on hardware when the true system and the model used in the FTOCP differ, which can lead to safety failures when either the FTOCP becomes infeasible or requires inputs which exceed the real-world bounds. These phenomena will be seen in the experimental demonstration in the next section.

V. Quadruped Experiments

To demonstrate the utility of the MPC+DCBF method and compare it to the effectiveness of other methods, we apply the proposed control algorithms to a dynamic collision avoidance scenario using a Unitree Go2 quadruped. We leverage a reduced order model (ROM) hierarchical control framework [54] for the platform based on a two-dimensional single-integrator high-level control interface enabling the assignment of safe translational velocity commands without modification of the low-level locomotion controller.

To generate the function h defining safety, we first perceive the experimental space using a fixed overhead RGB camera, which provides a persistent global image stream of the robot's 2D environment at 60 fps. This video stream is passed to the efficient Track-Anything-Model (efficientTAM) [62] image segmenter, a high-speed distillation of the Meta SAM2 segmentation model [63]. By segmenting the environment to detect predefined obstacles, we build a 2D occupancy map of the space. The occupancy map is buffered by the physical geometry of the Go2 quadruped, enabling safety of the robot to be defined via its centroid. Next, to produce h we use the Poisson-based algorithm developed in [64]. This method yields a single continuous h for the entire experimental environment, which can be queried during autonomous operation. The velocity of the obstacle is estimated using optical flow [65] on the segmented images and incorporated as a time-varying component in h. Additionally, an overhead OptiTrack motion capture system is used to estimate the translational and rotational states of the robot.

We employ this function h in the $\boldsymbol{\pi}^{\text{MPC+DCBF}}$ controller, producing safe velocities which the quadrupedal system tracks. The result of the model mismatch between the true quadrupedal system and the single integrator dynamics used by $\boldsymbol{\pi}^{\text{MPC+DCBF}}$ controller can be modeled as a disturbance to the system and analyzed using the recursive feasibility of Prop. 1 and Thm. 2 and through the ISSf theoretical lens, in which case the $\boldsymbol{\pi}^{\text{MPC+DCBF}}$ controller may produce extended periods of recursive feasibility and graceful safety degradation whereas the MPC controller may result in earlier and more catastrophic failures as it is practically unable to track large velocity commands.

To highlight safety-critical performance, we command the quadrupedal robot to hold a fixed reference coordinate $(1.75\,\text{m},\ 2.75\,\text{m})$ while staying in the safe set. We then roll a dodgeball into the environment, using a fixed-height ramp to produce a repeatable dynamic collision avoidance scenario. Across trials, safe set forward invariance was enforced via the three aforementioned methods: 1) $\boldsymbol{\pi}^{\text{MPC+DCBF}}$ – the focus of this work, 2) state-constrained $\boldsymbol{\pi}^{\text{MPC}}$ with $\alpha = 0$ – the naive MPC approach presented in Section II-B, and 3) the $\boldsymbol{\pi}^{\text{DCBF-OP}}$ safety filter using a proportional nominal controller – the myopic control approach presented in Section II-C. The resulting data for a single set of comparison experiments can be seen in Fig. 5.

By examining the top plot in the figure, it is immediately apparent that the MPC+DCBF method successfully enforces safe set forward invariance throughout the duration of the experiment. Meanwhile, state-constrained MPC and the DCBF safety filter both result in safety violations. Furthermore, the overhead camera images highlight key differences in how the Go2 quadruped attempt to avoid the dynamic obstacle. Due to its increased robustness and the incorporation of a planning horizon, the MPC+DCBF controller begins to command its avoidance maneuver significantly earlier than the other two methods. Although the decay of h for all three methods appears the same until 0.55 seconds, the MPC+DCBF begins to command motion before 0.25 seconds, moving along level sets of h to attain a more optimal position for future actions. This is a direct result of the tightening of the constraints of the underlying optimization problem since $\alpha > 0$, which forces the DTCBF constraint to activate sooner. Conversely, the state-constrained MPC controller reacts too late, commanding a large input to the single-integrator ROM which could not be tracked by the low-level locomotion controller. This inevitably led to a safety failure ($t = 0.75$ sec). Similarly, the small lateral gradients of h in the x direction cause the DCBF-OP controller to be unable to effectively "flow" around the obstacle given the real-time sampeld-data nature of the hardware experiment, eventually causing a safety failure as quadruped left the rectangular safe region ($t = 1.05$ seconds).

These experimental results were highly repeatable, as can be seen in the videos at the link in [10]. In fact, under these particular experimental conditions, the MPC+DCBF method maintained a 100% success rate, while the state-constrained MPC and DCBF-OP methods each had 0% success rates.

VI. Probabilistic Safety Guarantees

In this section, we extend our analysis of robustness beyond the worst-case ISSf property presented in Section IV to consider probabilistic and potentially unbounded disturbances. In particular, we will show that the stochastic reformulations of the DCBF constraint used in [15], [42] provide probabilistic robustness guarantees for the MPC+DCBF controller that may not exist for similar reformulations of the state-constrained MPC controller.

To provide these probabilistic guarantees, first let $(\Omega, \mathscr{F}, \mathbb{P})$ be a probability space where $\mathscr{F}_0 \subset \mathscr{F}_1 \subset \cdots \subset \mathscr{F}$ is a filtration of $\mathscr{F}$. Given this probability space, we now consider discrete-time dynamics with random uncertainty:

$$\mathbf{x}_{k+1} = \mathbf{F}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{d}_k), \quad \mathbf{d}_k \sim \mathcal{D}(\mathbf{x}_{k:0}), \quad \forall k \in \mathbb{N}, \quad (17)$$

where $\mathbf{d}_k$ is a random disturbance sampled from a distribution $\mathcal{D}(\mathbf{x}_{k:0})$ that is dependent on the history of the system's states $\mathbf{x}_{k:0}$ and that is a $\mathscr{F}_{k+1}$-measurable random variable taking values in $\mathbb{R}^{n_d}$. Here we generalize the analysis beyond

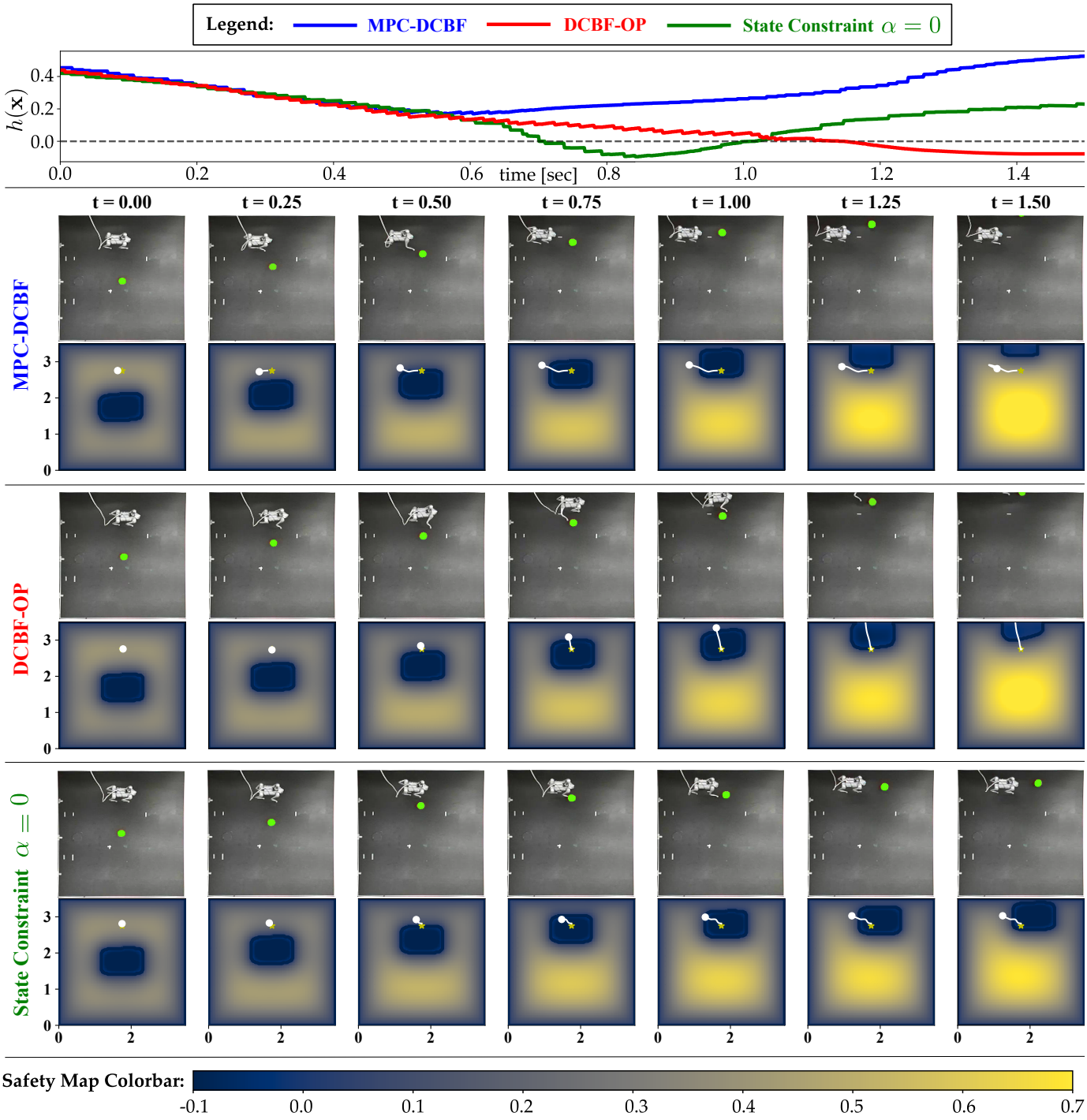


Fig. 5. Quadrupedal robot dynamic obstacle avoidance experiments for the DCBF-OP in red, state-constrained MPC (i.e., MPC+DCBF with $\alpha = 0$) in green, and the MPC+DCBF controller in blue. **(Top)** Time series plots of the safety value $h(\mathbf{x})$ for each controller. The MPC+DCBF successfully maintains safety during the experiment while the state-constrained MPC and the DCBF-OP both result in safety failures. **(2nd Row)** Overhead images with the dynamic obstacle highlighted in green and contour plots of $h(\mathbf{x})$ through time (left to right: $t = 0$ to $t = 1.5$) for the MPC+DCBF experiments. The quadruped successfully moves out of the way to avoid the dynamic obstacle. **(3rd Row)** Overhead images and safety contour plots for the DCBF-OP controller which is unable to plan around the obstacle and gets squeezed in between the wall and the dynamic obstacle until a failure occurs and the quadruped steps out of the safe region at $t = 1.0$ second. **(4th Row)** Overhead images and safety contour plots for the MPC controller which reacts too late and a safety failure and collision occur at approximately $t = 0.75$ sec. **(Bottom)** A colorbar showing the meaning of the colors in the $h(\mathbf{x})$ contour plots.

the additive disturbance of (11) to consider a discrete-time dynamics function that takes the disturbance uncertainty as an arbitrary input, i.e. $\mathbf{F} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_d} \to \mathbb{R}^{n_x}$.

As before we can use a state-feedback controller $\boldsymbol{\pi} : \mathbb{R}^{n_x} \to \mathbb{R}^{n_u}$ to create the closed-loop system:

$$\mathbf{x}_{k+1} = \mathbf{F}(\mathbf{x}_k, \boldsymbol{\pi}(\mathbf{x}_k), \mathbf{d}_k), \quad \mathbf{d}_k \sim \mathcal{D}(\mathbf{x}_{k:0}), \quad \forall k \in \mathbb{N}, \quad (18)$$

where the updated state is now a random variable given the current state information, and we may be unable to predict exactly how the state will evolve. In this case, our new goal

will be to define and guarantee a probabilistic notion of safety for this stochastic closed-loop system.

To make these guarantees, we next provide a important background information on stochastic safety and martingales and then present tools for generating probabilistic safety guarantees using various martingale concentration inequalities.

A. Background on Probabilistic Safety and Martingales

First, we must define the notion of safety that we seek to consider for the stochastic closed loop system (18). This definition is of particular importance since the probability-1, infinite-horizon guarantees of Thms. 1 and 4 may be impossible to achieve for systems like (18) given the presence of potentially unbounded uncertainty[9]. In fact, the probability that the system remains in any bounded set forever is zero in general due to the unbounded tails [68, Sec. IV].

This observation motivates a widely used (see [15], [29], [41], [69]) alternative finite-horizon definition of safety that is characterized by bounding the probability that the system will leave the safe set $\mathcal{C}$ within $K \in \mathbb{N}$ steps.

Definition 4 (K-Step Exit Probability)**.** *For any $K \in \mathbb{N}$ and initial condition $\mathbf{x}_0 \in \mathbb{R}^{n_x}$, the $K-$step exit probability of the set $\mathcal{C}$ for the closed-loop system* (18) *is:*

$$P_u(K, \mathbf{x}_0) \triangleq \mathbb{P}\{\mathbf{x}_k \notin \mathcal{C} \text{ for some } k \leq K\} \tag{19}$$

In order to provide a bound for this K-step exit probability, $P_u(K, \mathbf{x}_0)$, we first introduce martingales whose concentration inequalities will be useful in achieving robust safety guarantees. Martingales, and the related supermartingale, are classes of stochastic processes defined by the relationship between their mean and previous value:

Definition 5 (Martingale [70], [29])**.** *Let $(\Omega, \mathscr{F}, \mathbb{P})$ be a probability space with a filtration $\{\mathscr{F}_0, \mathscr{F}_1, \ldots, \mathscr{F}\}$. A stochastic process W_k that is adapted to the filtration and is integrable at each k is a martingale if*

$$\mathbb{E}[\ W_{k+1} \mid \mathscr{F}_k\ ] = W_k, \quad \forall k \in \mathbb{Z} \quad \text{(a.s.)}. \tag{20}$$

Additionally, if W_k is a supermartingale if it satisfies:

$$\mathbb{E}[\ W_{k+1} \mid \mathscr{F}_k\ ] \leq W_k, \quad \forall k \in \mathbb{Z} \quad \text{(a.s.)}. \tag{21}$$

For the probablistic guarantees that will be presented in the remainder of this work, we will rely on this notion of a *supermartingale* which can be thought of as processes that, on average, decay over time. This notion aligns naturally with the decay-based DCBF constraint. Just as supermartingales are defined by an inequality relationship between their current and previous values, the DCBF condition (4) enforces a similar inequality relationship between the current and previous values of safety $h(\mathbf{x}_{k+1})$ and $h(\mathbf{x}_k)$. To extend DCBFs to the stochastic setting and solidify their connections to martingales we consider the following stochastic variant of the DCBF condition (4):

$$\mathbb{E}[\ h(\mathbf{x}_{k+1}) \mid \mathscr{F}_k\ ] \geq \alpha h(\mathbf{x}_k), \tag{22}$$

for some $\alpha \in (0, 1)$. This expectation-based modification to (4) regulates the expected value of safety at the next step instead of the true value, since the exact value of $h(\mathbf{x}_{k+1})$ is random and unknowable at time-step k with filtration $\mathscr{F}_k$.

In practice, the stochasticity of real-world systems is often disregarded, in which case the practitioner may create controllers which consider the expected value of safety to be the true value, especially when the variance is small. If this is the case, then those standard implementations provide the inherent robustness properties (modulo Jensen's gap) presented in the remainder of this section without requiring additional modifications. This can be thought of as the inherent stochastic robustness property of the MPC+DCBF framework.

In the next two subsections we present how the stochastic variant of the DCBF inequality (22) has direct connections to martingales, which allows concentration inequalities from stochastic process theory to be leveraged to provide guaranteed bounds on the $K-$step exit probabilities for (18).

B. Probabilistic Safety Guarantees for DCBFs using Martingale Concentration Inequalities

In this section, we show how supermartingale concentration inequalities can be used to make theoretical guarantees on the closed-loop system (18) that satisfy the stochastic DCBF condition (22). In this context, one particularly useful martingale concetration inequality is Ville's inequality [71] which bounds the probability that a non-negative supermartingale rises above a particular threshold. Several works have incorporated this lemma into safety [15], [29], [69] and stability [41], [68] guarantees. The safety guarantees generated by Ville's inequality for DCBF-based systems were summarized in [42] as:

Theorem 5 (Safety using Ville's Inequality [15])**.** *If* (18) *satisfies* (22) *for all $k \leq K$ and if, for some $B > 0$, the function $h : \mathbb{R}^{n_x} \to \mathbb{R}$ satisfies $h(\mathbf{x}) \leq B$ for all $\mathbf{x} \in \mathbb{R}^{n_x}$, then*

$$P_u(K, \mathbf{x}_0) \leq 1 - \frac{\alpha^K h(\mathbf{x}_0)}{B}. \tag{23}$$

This guarantees that the risk of the system becoming unsafe within K steps is upper bounded by a function that decays to 1 with time and which depends on the system's initial safety "fraction", $h(\mathbf{x}_0)/B$. While the original definition of h may not be bounded, saturation can be used to build functions that satisfy the upper boundedness requirement of Thm. 5, especially since positive values of h do not affect the shape of $\mathcal{C}$ or its expansion $\mathcal{C}_d$. The utility of this guarantee was demonstrated on a quadrotor drone with significant chaotic disturbances avoiding collisions with the ground in [72]. Ultimately, these experiments showed that the probabilistic guarantee of Thm. 5 could be used to achieve highly-performant safe behavior.

Another useful supermartingale concentration inequality is Freedman's inequality [73, Thm. 4.1] which requires an alternative set of assumptions and has been shown to provide

[9]While probability-1, infinite-horizon guarantees have been made for continuous-time stochastic systems governed by stochastic differential equations (SDEs) [66], [67], they require infinite controller bandwidth which is impossible to achieve for real-world robotic systems with zero-order-hold, sampled-data implementations.

tighter guarantees than Thm. 5 in certain settings [42, Prop. 1]. The resulting safety guarantee generated by applying Freedman's inequality for DCBF-based systems was achieved in [42] as:

Theorem 6 (Safety with Freedman's [42])**.** *If* (18) *satisfies* (22) *for all* $k \leq K$ *and if, for some* $\delta_F, \sigma_F > 0$*, the following bounds on the difference between the true and predictable update* (24) *and the conditional variance* (25) *hold for all* $k \leq K$*:*

$$\mathbb{E}[\ h(\mathbf{x}_k) \mid \mathscr{F}_{k-1}\ ] - h(\mathbf{x}_k) \leq \delta_F, \tag{24}$$

$$\text{Var}(\ h(\mathbf{x}_{k+1}) \mid \mathscr{F}_k\ ) \leq \sigma_F^2, \tag{25}$$

then the K*-step exit probability is bounded as:*

$$P_u(K, \mathbf{x}_0) \leq \left(\frac{\xi^2}{\lambda+\xi^2}\right)^{\lambda+\xi^2} e^{\lambda} \tag{26}$$

where $\lambda = \frac{\alpha^K h(\mathbf{x}_0)}{\delta_F}$ *and* $\xi = \frac{\sigma_F\sqrt{K}}{\delta_F}$.

As with Thm. 5, the utility of the guarantee provided by Thm. 6 was shown via experimental demonstration. In this case it enabled safe navigation of complex environments using a bipedal robot operating with significant uncertainty [74].

Furthermore, since Thm. 6 requires a bound on the difference between the true and expected values of safety, the worst-case bounding analysis of ISSf and Thm. 4 also applies. To compare with ISSf's worst-case safe set $\mathcal{C}_d$, [42] anlyzes the $K-$step exit probabilities over this expanded safe set. A simulated comparison of the ISSf property with the probabilistic bound that can be generated using Freedman's inequality is provided in [42, Sec. III.C]. These simulations show that, by utilizing a probabilistic understanding of the underlying disturbance distribution in place of the adversarial disturbance bound, we can generate useful risk-based safety probabilities for a variety of $\mathcal{C}_d$ level sets that are significantly less conservative than standard ISSf.

C. Probabilistic Safety with MPC+DCBF

As with the deterministic case, safety filters that enforce the expectation-based DCBF condition (22) provide pointwise optimal control actions while guaranteeing a K-step exit probability bound, but they suffer from the same myopia and undesirable equilibrium that their deterministic counterpart demonstrated in Ex. 1 and Fig. 3. Thus, to benefit from the horizon-long optimization of MPC, in this section, we consider risk-based safety guarantees for system (18) under the influence of a modified MPC+DCBF controller that accounts for stochastic dynamics uncertainty.

Similarly, we note that the stochastic guarantees of Thms. 5 and 6 rely on $\alpha \in (0, 1)$ and cannot be used to provide guarantees when $\alpha = 0$. The self-referential, safety-feedback property of the DCBF constraint is critical in creating the necessary supermartingale relationship to invoke the concentration inequalities, and when α increases toward 1, these bounds guarantee a lower risk of safety failure. Alternatively, when $\alpha = 0$, as in the typical state-constraint formulation, these methods can no longer be used to make probabilistic guarantees and return vacuous probability bounds. Thus, the stochastic DCBF constraint provides inherent probabilistic robustness that the standard MPC problem with an expectation-enforced state constraint does not.

As in the deterministic case, to benefit from the horizon based planning of MPC and the inherent robustness properties of stochastic DCBFs, we propose unifying them in the form of a stochastic Model-Aware Risk-Informed Optimization (MARIO) optimization problem:

$$\begin{aligned}
\min_{\substack{\boldsymbol{\xi}_{0:N}\in\mathbb{R}^{n_x}\\ \boldsymbol{\nu}_{0:N-1}\in\mathbb{R}^{n_u}}} \quad & \mathbb{E}\left[\sum_{i=0}^{N-1} c(\boldsymbol{\xi}_i, \boldsymbol{\nu}_i) + V(\boldsymbol{\xi}_N)\middle|\mathscr{F}_k\right] && \text{(MARIO)}\\
\text{s.t.} \quad & \boldsymbol{\xi}_{i+1} = \mathbf{F}(\boldsymbol{\xi}_i, \boldsymbol{\nu}_i, \mathbf{d}_i), \quad \forall i \in \{0, \dots, N-1\}\\
& \mathbb{E}[h(\boldsymbol{\xi}_{i+1})|\mathscr{F}_k] \geq \alpha h(\boldsymbol{\xi}_i), \quad \forall i \in \{0, \dots, N-1\}\\
& \boldsymbol{\nu}_k \in \mathcal{U}, \quad \forall i \in \{0, \dots, N-1\}\\
& \mathbf{d}_i \sim \mathcal{D}(\mathbf{x}_{k:0})\\
& \boldsymbol{\xi}_0 = \mathbf{x}_k
\end{aligned}$$

with the MARIO controller $\boldsymbol{\pi}^{\text{MARIO}}(\mathbf{x}_k) = [\boldsymbol{\nu}_0^*(\mathbf{x}_k)]$.

In the reformulation of FTOCP+DCBF to MARIO, we replaced (4) with the expectation-based condition:

$$\mathbb{E}[h(\mathbf{F}(\boldsymbol{\xi}_i, \boldsymbol{\nu}_i, \mathbf{d}_i))|\mathscr{F}_k] \geq \alpha h(\boldsymbol{\xi}_i),\ \forall i \in \{0, \dots, N\}. \tag{27}$$

Importantly this reformulation is always conditioned on $\mathscr{F}_k$ for all prediction steps i along the horizon. Thus, the planner's understanding of the uncertainty distribution at each step is only dependent on the current state history, $\mathcal{D}(\mathbf{x}_{k:0})$, making this controller causal and realizable.

When practitioners ignore the stochasticity of real-world systems in implementing the $\boldsymbol{\pi}^{\text{MPC+DCBF}}$ controller, they often assume that they can predict the "true" value of safety at the next step [33]. This "true" prediction can be thought of as an approximation of the expected value of safety at the next time-step $k+1$ given the current information about the system (i.e. $\mathbf{x}_k$), in which case the FTOCP+DCBF can be seen as an approximation of MARIO.

Since this controller enforces the expectation-based DCBF constraint (22) at the first step, the closed loop system (18) under this controller satisfies the DCBF condition required for Thms. 5 and 6 to hold. As mentioned above, these theorems allow us to make guarantees for the $\boldsymbol{\pi}^{\text{MARIO}}$ controller only when $\alpha > 0$ and not when $\alpha = 0$, meaning that the $\boldsymbol{\pi}^{\text{MARIO}}$ controller provides robustness guarantees that the state-constrained MPC does not, since we cannot use the constraint to construct a supermartingale from the system.

Thus, the $\boldsymbol{\pi}^{\text{MARIO}}$ controller, which can be considered as a stochastic formalization of the $\boldsymbol{\pi}^{\text{MPC+DCBF}}$, immediately benefits from inherent robustness guarantees of Thms. 5 or 6 which do not apply to the stochastic reformulation of the $\boldsymbol{\pi}^{\text{MPC}}$ controller with $\alpha = 0$. Instead, most stochastic MPC methods rely on a quantile-based chance constraint [30], [75] which can require significantly more distribution information than the supermartingale methods which are based exclusively on the first-moment. Similar to how the ISSf property of Sec. IV provides robust safety guarantees when only the undisturbed model of the system is known, and tube MPC methods require knowledge of the tube size to provide a guarantee, the

guarantees of this section ensure bounded failure probability using only the first moment of safety. In contrast, quantile-based methods would require significantly more distributional information. Additionally, because the supermartingale methods rely on an inequality on the expectation, they can be thought of as distributionally robust, since the guarantees hold for all distributions which satisfy the first-moment property. This robustness is achieved at the cost of looseness of the probability bound.

Notably, enforcing the expectation-based DCBF constraint (22) may require additional consideration since the dynamics and safety function h may reshape the disturbance distribution. In this case, sampling-based methods may be used to approximate the expectation constraint, or the following proposition can be used to account for the effect of Jensen's inequality in the DCBF constraint:

Proposition 2. *Assume that the dynamics are affine with respect to the disturbance, $\mathbf{x}_{k+1} = \mathbf{F}(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{d}_k$. Consider two cases for the properties of $h : \mathbb{R}^{n_x} \to \mathbb{R}$:*

1) If h is convex, then

$$h(\mathbf{F}(\mathbf{x}_k, \mathbf{u}_k) + \mathbb{E}[\mathbf{d}|\mathscr{F}_k]) \geq \alpha h(\mathbf{x}_k) \tag{28}$$
$$\implies \mathbb{E}[h(\mathbf{F}(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{d})|\mathscr{F}_k] \geq \alpha h(\mathbf{x}_k) \tag{29}$$

2) If h is concave, twice continuously-differentiable, and has a bounded second-derivative norm (i.e., $\sup_{\mathbf{x}\in\mathbb{R}^{n_x}} \|\nabla^2 h(\mathbf{x})\| \leq \lambda_{\max}$ for some $\lambda_{\max} \geq 0$), then

$$h(\mathbf{F}(\mathbf{x}_k, \mathbf{u}_k) + \mathbb{E}[\mathbf{d}|\mathscr{F}_k]) - \frac{\lambda_{\max}}{2}\mathrm{tr}(\mathrm{cov}(\mathbf{d})) \geq \alpha h(\mathbf{x}_k)$$
$$\implies \mathbb{E}[h(\mathbf{F}(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{d})|\mathscr{F}_k] \geq \alpha h(\mathbf{x}_k) \tag{30}$$

Proof. For (1) apply Jenson's inequality for the convex function h. For (2) see [15, Thm. 6]. □

This proposition provides two methods for practically enforcing the necessary constraint to achieve safety in certain circumstances. Since only the first constraint along the planning horizon of the MARIO FTOCP is leveraged for the trajectory long guarantees of Thms. 5 and 6, further simplifications can be made for $i > 1$ to ease the computational burden. While this may reduce the optimality of the $\boldsymbol{\pi}^{\text{MARIO}}$ controller, it will still maintain its probabilistic guarantees.

Furthermore, we note that practically implementing the $\boldsymbol{\pi}^{\text{MARIO}}$ controller may be difficult, as it requires propagation of compounding uncertainty through the dynamics, cost, and the DCBF h. With this practical applicability in mind, we introduce a more computationally tractable version of the $\boldsymbol{\pi}^{\text{MARIO}}$ controller in the next section that also considers state uncertainty.

VII. MPC+DCBF WITH STATE UNCERTAINTY

Finally, we consider systems with stochastic dynamic uncertainty as in (18) where we do not have direct access to the state (the maximal uncertainty discussed in the work). Instead we only have indirect access through noisy measurements:

$$\mathbf{y}_k = \mathbf{M}(\mathbf{x}_k, \mathbf{v}_k), \qquad \mathbf{v}_k \sim \mathcal{V}(\mathbf{x}_{k:0}) \tag{31}$$

where, $\mathbf{y}_k \in \mathbb{R}^{n_y}$ is a system measurement, $\mathbf{v}_k$ is a $\mathscr{F}_{k+1}$-measurable random variable taking values in $\mathbb{R}^{n_v}$ that represents the measurement noise, and $\mathbf{M} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_v} \to \mathbb{R}^{n_y}$ is the system's measurement function that obtains measurement $\mathbf{y}_k$ given the state $\mathbf{x}_k$; for example, $\mathbf{M}$ could be a camera that produces a noisy image $\mathbf{y}_k$ given the current position $\mathbf{x}_k$ and the noise $\mathbf{v}_k$.

In the context of real-world robotics and control systems, we never have access to the true state of the system due to uncertainties in our measurements. Because of this inability to access the true state, the state evolution of $\mathbf{x}_k$ becomes a partially observable Markov decision process (POMDP), and it is common to seek guarantees on the belief-state distribution instead of guarantees on the true state [47], [76]. In this work we will first seek a bound on the belief space safety and then extend this to the true state through a union bounding method.

Since we are considering systems where we do not know the true state $\mathbf{x}_k$, we now discuss a filtration generated by the $\sigma-$algebra over only the observations, $\mathbf{y}_k$; that is, we consider $\mathscr{G}_k = \sigma(\mathbf{y}_{k:0})$ which is contained in the filtration $\mathscr{F}_k$, i.e., $\mathscr{G}_k \subset \mathscr{F}_k$. Since the inputs are selected based on the system measurements, the input vector $\mathbf{u}_k$ is also $\mathscr{G}_k$-measurable. On the other hand, since the true state of the system $\mathbf{x}_k$ is not $\mathscr{G}_k$-measurable, we turn our attention to the expectation of the state conditioned on the measurement-based filtration $\mathscr{G}_k$. We will use the previously introduced martingale constructions to make safety guarantees with respect to the expected value of the belief state:

$$\widehat{\mathbf{x}}_{k|k-1} \triangleq \mathbb{E}[\ \mathbf{x}_k \mid \mathscr{G}_{k-1}\ ], \qquad \widehat{\mathbf{x}}_{k|k} \triangleq \mathbb{E}[\ \mathbf{x}_k \mid \mathscr{G}_k\ ], \tag{32}$$

where, as in a Kalman filter [77], $\widehat{\mathbf{x}}_{k|k-1}$ is the expected value of the predicted belief state at the next step and $\widehat{\mathbf{x}}_{k|k}$ is the expected value of the updated belief state after a new measurement has been taken.

Next we seek to produce risk-based bounds on the K-step exit probability of the expected value of the belief state:

$$P_u(K, \widehat{\mathbf{x}}_{0|0}) \triangleq \mathbb{P}\{\ \widehat{\mathbf{x}}_{k|k} \notin \mathcal{C} \text{ for some } k \leq K\ \}, \tag{33}$$

where the dynamics for $\widehat{\mathbf{x}}_{k|k}$ include both the system dynamics for the prediction propagation to $\widehat{\mathbf{x}}_{k+1|k}$ and the system measurement update step to obtain $\widehat{\mathbf{x}}_{k+1|k+1}$ once $\mathbf{y}_{k+1}$ is measured.

To bound (33) we consider the following condition on $h(\widehat{\mathbf{x}}_{k+1|k+1})$ in order to generate stochastic guarantees:

$$\mathbb{E}[\ h(\widehat{\mathbf{x}}_{k+1|k+1}) \mid \mathscr{G}_k\ ] \geq \alpha h(\widehat{\mathbf{x}}_{k|k}), \tag{34}$$

for some $\alpha \in (0, 1)$ where the expectation-based DTCBF condition in (22) is now applied to the expected value of the updated belief state.

A. SUP-MARIO for Belief-Space Safety

To practically implement constraint (34) and use it in conjunction with the results from Sec. VI to generate guarantees, we assume that the dynamics and measurements are linear:

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k + \mathbf{d}_k \tag{35}$$
$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k + \mathbf{v}_k \tag{36}$$

and that the disturbance $\mathbf{d}_k$ and noise $\mathbf{v}_k$ are sampled from zero-mean distributions[10].

To construct an FTOCP for this system, instead of constraining the initial state in the plan to be the current state $\boldsymbol{\xi}_0 = \mathbf{x}_k$, which we do not have access to, we constrain it to the current expected belief state $\boldsymbol{\xi}_0 = \widehat{\mathbf{x}}_{k|k}$. Using this adjustment and the linear dynamics assumption we have the State-Uncertain Probabilistic Model-Aware Risk Informed Optimization (SUP-MARIO) optimization problem:

$$\min_{\substack{\boldsymbol{\xi}_{0:N}\in\mathbb{R}^{n_x}\\ \boldsymbol{\nu}_{0:N-1}\in\mathbb{R}^{n_u}}} \sum_{i=0}^{N-1} c(\boldsymbol{\xi}_i, \boldsymbol{\nu}_i) + V(\boldsymbol{\xi}_N) \qquad \text{(SUP-MARIO)}$$

$$\begin{aligned} \text{s.t.}\quad & \boldsymbol{\xi}_{i+1} = \mathbf{A}\boldsymbol{\xi}_i + \mathbf{B}\boldsymbol{\nu}_i, \quad \forall i \in \{0,\dots,N-1\} \\ & h(\boldsymbol{\xi}_{i+1}) \geq \alpha h(\boldsymbol{\xi}_i), \quad \forall i \in \{0,\dots,N-1\} \\ & \boldsymbol{\nu}_k \in \mathcal{U}, \quad \forall i \in \{0,\dots,N-1\} \\ & \boldsymbol{\xi}_0 = \widehat{\mathbf{x}}_{k|k} \end{aligned}$$

for some $\alpha \in (0,1)$ which can be used as before to define the SUP-MARIO controller $\boldsymbol{\pi}^{\text{SUP-MARIO}}(\mathbf{x}_k) = [\boldsymbol{\nu}_0^*(\mathbf{x}_k)]$.

Notably, whereas $\boldsymbol{\pi}^{\text{MARIO}}$ may be difficult to implement, the implementation of $\boldsymbol{\pi}^{\text{SUP-MARIO}}$ is straightforward. This is because $\boldsymbol{\pi}^{\text{SUP-MARIO}}$ relies on the expected-value of the state to implement the FTOCP+DCBF problem[11]. It therefore does not require the uncertainty distributions be propagated across the planning horizon.

Next, we show that this simple-to-implement controller still satisfies the DCBF condition in expectation (34). In particular, the $\boldsymbol{\pi}^{\text{SUP-MARIO}}$ controller satisfies the desired DCBF constraint on the safety of the belief state (34), which allows it to leverage Thm. 5 and/or Thm. 6 to provide bounds on the $K-$step exit probability of the belief state (33).

Theorem 7. *For systems with linear dynamics* (35)*, linear measurements* (36)*, and zero-mean disturbance* $\mathbf{d}_k$ *and measurement noise* $\mathbf{v}_k$*, if* $h : \mathbb{R}^{n_x} \to \mathbb{R}$ *is convex*[12]*, then the closed-loop system* (18) *with the* $\boldsymbol{\pi}^{\text{SUP-MARIO}}$ *controller satisfies:*

$$\mathbb{E}[\ h(\widehat{\mathbf{x}}_{k+1|k+1}) \mid \mathscr{G}_k\ ] \geq \alpha h(\widehat{\mathbf{x}}_{k|k}) \tag{37}$$

A proof of Thm. 7 can be found in Appx. C.

Since the $\boldsymbol{\pi}^{\text{SUP-MARIO}}$ satisfies condition (34), Thms. 5 and 6 can be used to guarantee bounds on the K-step exit probability of $\widehat{\mathbf{x}}_{k|k}$ when their respective hypotheses regarding bounds on h or step-wise and predictable quadratic variation (PQV) bounds are satisfied.

B. Ground Truth Safety Guarantees

Finally, to analyze the safety achieved by the $\boldsymbol{\pi}^{\text{SUP-MARIO}}$ controller with respect to the true state $\mathbf{x}$, we can leverage tail-bounding methods like Cantelli's inequality (one-sided Chebychev's inequality) in conjunction with the union bound (Boole's inequality) to extend beyond the $K-$step exit probability bounds on $\widehat{\mathbf{x}}_{k|k}$.

Using Cantelli's inequality we can extend a safety guarantee on the expected belief state $\widehat{\mathbf{x}}_{k|k}$ to a safety guarantee on the true state $\mathbf{x}_k$.

Theorem 8. *Assume that the variance of safety is bounded as* $\text{Var}(h(\mathbf{x}_k)) \leq \sigma_h$ *for some* $\sigma_h > 0$*, all* $\mathbf{x}_k$*, and all* $k \leq K$ *and that* h *is convex. If the system achieves the* K*-step exit probability* $P_u(K, \widehat{\mathbf{x}}_0) \leq \epsilon$ *for the belief state* $\widehat{\mathbf{x}}_{k|k}$*, then failure probability for the* $\mathcal{C}_{\delta_C}$ *for the true state* $\mathbf{x}$ *and some* $\delta_C \geq 0$ *is bounded as:*

$$\begin{aligned} &\mathbb{P}\{h(\mathbf{x}_k) \leq -\delta_C \text{ for some } k \leq K\} \\ &\qquad \leq \epsilon + (1+K)\left(\frac{\sigma_h^2}{\sigma_h^2 + \delta_C^2}\right). \end{aligned} \tag{38}$$

This final theorem allows us to place theoretical guarantees on the probability that the true state of the system will be safe despite indirect knowledge of $\mathbf{x}$ due to noisy measurements. Its proof can be found in Appx. D.

VIII. Quadrotor Experiments

In this section we apply the $\boldsymbol{\pi}^{\text{SUP-MARIO}}$ controller to a quadrotor robot to achieve dynamic obstacle avoidance. To do this we consider the following model of the quadrotor [72]:

$$\underbrace{\frac{d}{dt}\begin{bmatrix}\mathbf{p}\\ \mathbf{q}\\ \mathbf{v}\end{bmatrix}}_{\dot{\mathbf{x}}} = \begin{bmatrix}\mathbf{v}\\ \mathbf{0}\\ -\mathbf{e}_z g\end{bmatrix} + \begin{bmatrix}\mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{I}\\ \frac{1}{m}\mathbf{R}(\mathbf{q})\mathbf{e}_z & \mathbf{0}\end{bmatrix} \underbrace{\begin{bmatrix}\tau\\ \boldsymbol{\omega}\end{bmatrix}}_{\mathbf{u}}, \tag{39}$$

where the state $\mathbf{x} = (\mathbf{p} \in \mathbb{R}^3, \mathbf{q} \in \mathbb{S}^3, \mathbf{v} \in \mathbb{R}^3)$ represents the position, orientation, and velocity of the system. Here g represents gravity, $m = 1.12$ kg is the robot's mass, and the system has inputs of angular rate $\boldsymbol{\omega} \in \mathbb{R}^3$ and thrust force $\tau \in \mathbb{R}_{\geq 0}$. Here $\mathbf{e}_z$ is a unit vector in the z-direction and $\mathbf{R} : \mathbb{S}^3 \to \text{SO}(3)$ maps the quaternion representation of orientation to the respective rotation matrix.

To control the quadrotor robot we use a hierarchical control scheme that consists of three layers. At the lowest layer we use an opensource Betaflight controller to track commanded thrust and angle rates at 8 kHz. At the mid-layer, we implement the geometric tracking controller presented in [78] at 800 Hz to generate thrust and angle rate commands based on desired position trajectories. Finally, we generate twice continuously differentiable position outputs using the $\boldsymbol{\pi}^{\text{SUP-MARIO}}$ controllerat 20 Hz, where the linear model used in the SUP-MARIO FTOCP is:

$$\underbrace{\begin{bmatrix}\mathbf{p}_{k+1}\\ \mathbf{v}_{k+1}\end{bmatrix}}_{\boldsymbol{\xi}_{k+1}} = \begin{bmatrix}\mathbf{I} & \Delta_t\mathbf{I}\\ \mathbf{0} & \mathbf{I}\end{bmatrix} \underbrace{\begin{bmatrix}\mathbf{p}_k\\ \mathbf{v}_k\end{bmatrix}}_{\boldsymbol{\xi}_k} + \begin{bmatrix}\frac{\Delta_t^2}{2}\mathbf{I}\\ \Delta_t\end{bmatrix} \underbrace{\begin{bmatrix}\mathbf{a}_k\end{bmatrix}}_{\boldsymbol{\nu}_k}. \tag{40}$$

To limit the angle rate commands produced by the tracking controller and ensure smoother flight, we add a constraint on the system jerk by bounding the difference between the current and next acceleration inputs in the FTOCP and SUP+MARIO

[10]To prove Thm. 7 we assume zero mean, but any bias can also be accounted for by modeling it and including it as a part of the nominal dynamics and measurement model.

[11]The relationship between the FTOCP+DCBF and SUP-MARIO is similar to that between a linear quadratic regulator (LQR) and a linear quadratic guassian (LQG) controller.

[12]When h is concave, the method in Prop. 2 can be used. Alternatively, sampling-based methods can be used to approximate $\mathbb{E}[h(\mathbf{x})]$ from $h(\mathbb{E}[\mathbf{x}])$.

problems. Furthermore, to avoid infeasibility during flight, we implement these finite-difference-based jerk bounds as soft constraints with slack variables.

Although the tracking controller can be used to establish the differential flatness of the quadrotor system [79] that ensures the (almost everywhere) tracking of the desired trajectories, any resulting error in the model that occurs transiently due to initial condition error, angle-rate convergence, or lack of smoothness between solution updates can be analyzed through the robustness frameworks presented in Sections IV and VI.

For safety, we consider collision avoidance between our quadrotor drone and a dynamic projectile obstacle. Mathematically we define this safety using the function:

$$h_0(\mathbf{x}) = \|\tilde{\mathbf{p}}_{x:y}\| - r \tag{41}$$

where $\tilde{\mathbf{p}}$ represents the relative position of the quadrotor with respect to the obstacle, the subscript indices indicate the extraction of the first two elements of $\tilde{\mathbf{p}}$, and r is the radius of the obstacle, which accounts for the maximum dimension of the quadrotor. The 0-superlevel set of h_0 functionally defines safety for our quadrotor system as staying outside the planar (x, y) region containing the obstacle. To implement the horizon-based planning of the MPC and SUP-MARIO controllers, we use a constant-velocity model of the dynamic obstacle.

While $h_0(\mathbf{x})$ is used in the implementations of the $\boldsymbol{\pi}^{\text{MPC}}$ with state constraints, we instead use a higher-order CBF (HOCBF) extension [56] to implement the $\boldsymbol{\pi}^{\text{DCBF-OP}}$ and $\boldsymbol{\pi}^{\text{SUP-MARIO}}$ controllers:

$$h(\mathbf{x}) = \underbrace{\frac{\tilde{\mathbf{p}}_{x:y}^{\top}}{\|\tilde{\mathbf{p}}_{x:y}\|}\dot{\tilde{\mathbf{p}}}_{x:y}}_{\dot{h}_0(\mathbf{x})} + \gamma h_0(\mathbf{x}) \tag{42}$$

for a $\gamma > 0$. This is a relative degree 1 DCBF designed using the HOCBF method [57] for the double integrator model used in the $\boldsymbol{\pi}^{\text{SUP-MARIO}}$ controller. While the use of h as in (42) in place of h_0 is not theoretically necessary, and in fact moves from a convex h_0 to a non-convex h, we find that it provides significantly better closed-loop system behavior.

An additional DCBF constraint was also implemented in the $\boldsymbol{\pi}^{\text{MPC}}$ and $\boldsymbol{\pi}^{\text{SUP-MARIO}}$ controllers to avoid collisions with the ground.

A. Simulation Experiments

In simulation we compare the effectiveness of the $\boldsymbol{\pi}^{\text{DCBF-OP}}$, $\boldsymbol{\pi}^{\text{MPC}}$, and $\boldsymbol{\pi}^{\text{SUP-MARIO}}$ controllers[13] at achieving dynamic collision avoidance for the quadrotor system (39). The results of this comparison simulation can be seen in Fig. 6.

To approximate real-world uncertainty, random noise was added to both the state and obstacle values and measurements and Kalman filters were implemented to estimate both the robot and obstacle states. The goal position for each controller was $(0, 0)$ which the MPC and SUP-MARIO controllers tracked using a quadratic receding horizon cost. The DCBF controller tracked the goal position using a proportional-derivative nominal controller which was pointwise modified to enforce safety. The obstacle was a sphere with radius $r_{\text{obs}} = 0.15$ m that tracked a trajectory that passed through $(0, 0)$ which was both the robot's initial and goal position.

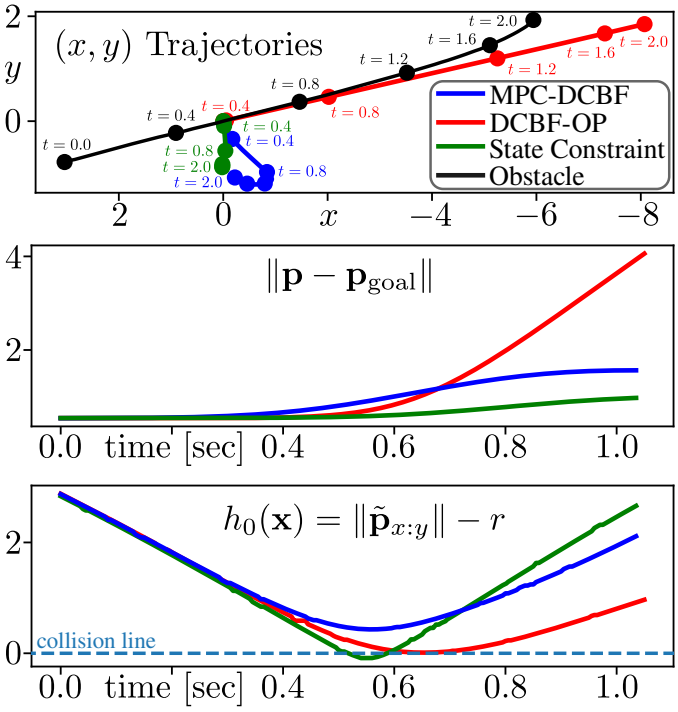


Fig. 6. Simulated demonstrations of the $\boldsymbol{\pi}^{\text{MPC}}$, $\boldsymbol{\pi}^{\text{DCBF-OP}}$, and $\boldsymbol{\pi}^{\text{SUP-MARIO}}$ controllers performing a dynamical obstacle avoidance task. **(Top)** The planar (x, y) trajectories for the obstacle (black), DCBF-OP (red), MPC+DCBF (blue), and MPC with state constraints (green). **(Middle)** The distance to the goal through time for each controller. **(Bottom)** The signed-distance function representing collision between the drone and the obstacle. Aided by their planning horizons, the $\boldsymbol{\pi}^{\text{MPC}}$ controller with state constraints and the $\boldsymbol{\pi}^{\text{SUP-MARIO}}$ controller both produce trajectories with relatively small deviations away from the goal point $(0, 0)$. Meanwhile, the $\boldsymbol{\pi}^{\text{DCBF-OP}}$ and $\boldsymbol{\pi}^{\text{SUP-MARIO}}$ controllers manage to avoid collisions, but the $\boldsymbol{\pi}^{\text{MPC}}$ controller results in a safety failure due to a the model-mismatch and lack of robustness. Here the DCBF-OP achieves safety during the scenario, but its myopic, pointwise optimization does so by moving in the same direction as the obstacle whereas the $\boldsymbol{\pi}^{\text{SUP-MARIO}}$ controller optimizes performance while achieving safety by moving in a direction which is predominantly orthogonal to the obstacle's velocity. Videos of these simulations can be found at [10].

While the $\boldsymbol{\pi}^{\text{DCBF-OP}}$ and $\boldsymbol{\pi}^{\text{SUP-MARIO}}$ controllers both achieve safety, the myopia of the DCBF-OP safety filter causes it to produce trajectories that are safe but highly suboptimal by moving in the same direction as the obstacle, resulting in significant departure from the goal position. The $\boldsymbol{\pi}^{\text{MPC}}$ controller with state constraints achieves performant behavior by planning a trajectory that moves perpendicular to the obstacle path; however, it lacks sufficient robustness resulting in a safety failure in simulation. Finally, the $\boldsymbol{\pi}^{\text{SUP-MARIO}}$ controller combines the benefits of both methods and achieves safe and performant behavior, effectively side-stepping with sufficient margin to avoid collision.

The simulations were performed in a ROS-based simulation environment which models the full-robot multi-layer control and communication system.

[13] To practically implement the $\boldsymbol{\pi}^{\text{DCBF-OP}}$ we added an ISSf-CBF [5] term to account for model uncertainty. This term was not added in the implmentation of the $\boldsymbol{\pi}^{\text{SUP-MARIO}}$ controller.

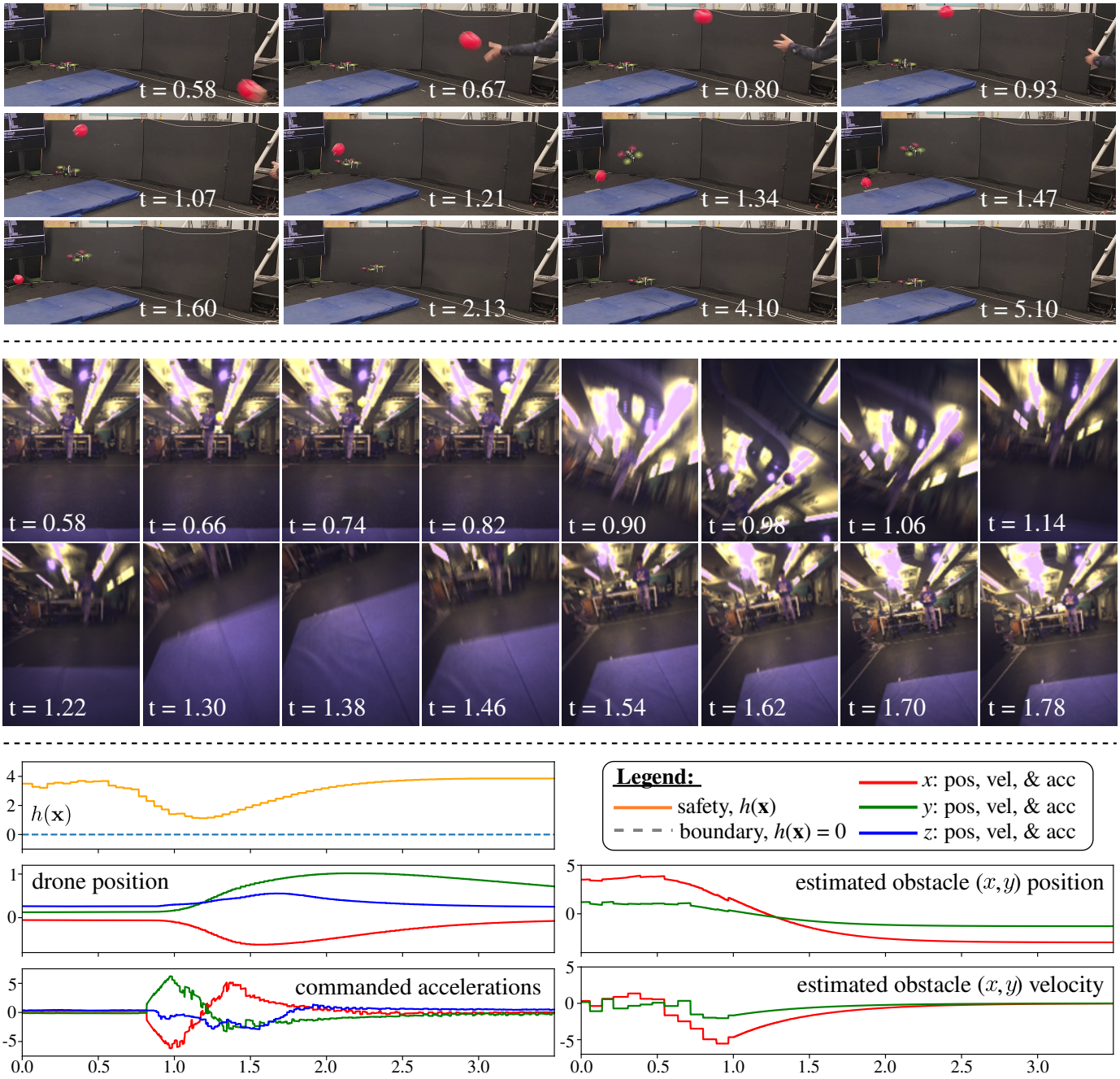


Fig. 7. Experimental demonstration of the $\pi^{\text{SUP-MARIO}}$ controller on a quadrotor drone with onboard dynamic obstacle detection, state estimation, and avoidance. **(Top)** The top two rows show the scene through time from the perspective of an external camera. The drone starts in the top left and a red ball moves towards it. The drone then moves back and to the side to avoid a collision. **(Middle)** The middle two rows show the scene from the robot's perspective. The obstacle mask, as seen by the robot, is shown in yellow and can be seen to have significant noise with an innacurate mask at $t = 0.58$, significant motion blur, and missing detections at $t = 0.90$ and 0.98. **(Bottom)** The bottom plots show the system's safety value $h(\mathbf{x})$ in orange (which never drops below zero), the drone position, the commanded acceleration, and the obstacle's estimated position and velocity. The different components of these vectors are shown as x in red, y in green, and z in blue. The maximum velocity of the obstacle during the experiment was measured using a motion capture system to be 6.24 m/s. The video of this experiment can be found at the link in [10].

B. Hardware Setup

The quadrotor hardware platform is built on a Chimera 7" frame with four iFlight XING X2806.5 1300KV brushless motors, a T-Motor F55 A Prop II 4-in-1 ESC, a MAMBA BASIC F722 running a betaflight flight controller, a Teensy 4.1 microcontroller, a VectorNav VN-200 IMU, an Intel RealSense D455 depth camera, and a NVIDIA Jetson Orin NX computer. We utilize the IMU and its internal Kalman filter for orientation state estimation, and we use an OptiTrack motion capture system for global position measurements from which velocities are also estimated via finite-difference and a low-pass filter. A diagram of the quadrotor is provided in Fig. 8.

The flight controller is used to track desired angle rate and thrust commands generated by the Jetson Orin NX computer. With the expception of the global position measurements provided by the motion capture system over WiFi, all computations for image processing, state estimation, and control are performed onboard. The environment sensing system utilizes an RGBd video stream (4 channels: 3 color channels and 1

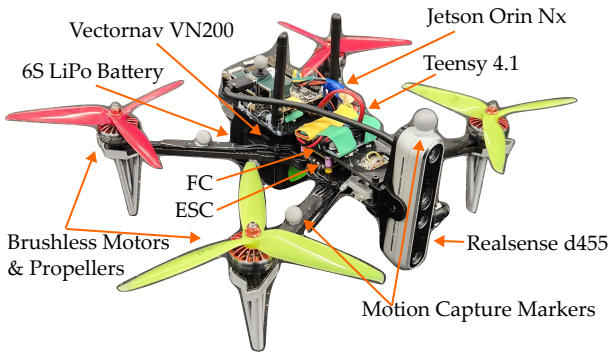


Fig. 8. A diagram of the quadrotor robot used in the experiments in Section VIII and Figs. 1 and 7.

depth channel) generated by the Intel Realsense D455 stereo depth camera. To ensure time alignment between the color and depth images, the stereo image used to calculate the depth is simultaneously used as the RGB image, and the active infrared projector is disabled. To segment the obstacle within an image and track it between frames, we utilize the efficient Track-Anything-Model (efficientTAM) [62], a small distillation of Meta's Segment-Anything-Model [63], which achieves faster than 11 Hz image segmentation and tracking on the Jetson Orin NX. To identify obstacles, we initialize the efficientTAM model with prompts (mouse clicks on an image) which identify key points in the image. For the obstacle we predominately used the red ball shown in Figs. 1 and 7, but the experimental video found at the link in [10] also shows a demonstration of the robot performing state estimation on and avoiding a toy green turtle shell. This demonstration of the SUP-MARIO controller dodging a green turtle shell can be seen in Fig. 9.

Once a segmentation mask is obtained for an RGBd image, the intrinsic camera matrix and the geometry of the robot are used to generate 3D vectors representing the relative position of the pixelized masked image contents from the robot body reference frame. We then perform a weighted averaging of those vectors based on their distance to the center of the mask to estimate the relative position of the obstacle centroid with respect to the camera frame. This relative position is converted to a global frame using the time-synchronized drone state. Given this position estimate, a Kalman filter for a double integrator system is used to estimate the relative obstacle position $\tilde{\mathbf{p}}$ and velocity $\dot{\tilde{\mathbf{p}}}$ which relate to system safety via h_0 in (41) and h in (42).

Finally, the $\boldsymbol{\pi}^{\text{SUP-MARIO}}$ controller is implemented at 20 Hz with a horizon length of $N = 20$ and a $\Delta_t = 0.05$ sec, for a total real-time horizon length of 1 second. To solve the non-convex optimization problem we use a sequential quadratic programming (SQP) method in a real-time iteration (RTI) implementation [80]. The DCBF-OP and MPC controllers were not demonstrated on hardware due to practical safety concerns with the simulated trajectories in Fig. 6. Results from these experiments can be seen in Figs. 1 and 7 and a video can be found at the link in [10] where the quadrotor drone successfully avoids dynamic projectile obstacles over several trials. For the experiment shown in Fig. 7, motion capture markers were added to the obstacle to provide ground truth state information, but this ground truth was *not used for real-time collision avoidance*. These ground truth measurements only used to obtain the true obstacle velocities, and from this we know that the quadrotor robot successfully dodged obstacles moving at upwards of $6.24\ \mathrm{m/s}$, overcoming the significant uncertainty resulting from the noisy, low-frame-rate (11 Hz) environmental perception and from its uncertain, reduced-order-model of its dynamics.

Fig. 9. The quadrotor robot avoiding a collision with the toy turtle shell.

IX. Conclusion

In this work we studied the combination of two predominant control techniques: model predictive control (MPC) and control barrier function (CBF) based safety filters. By combining the cost function and horizon-based planning of the MPC problem with the DCBF-based safety constraint, we found both practical and theoretical benefits in nominal operation, operation under bounded uncertainty, and operation under (potentially unbounded) stochastic state and dynamic uncertainty, that extend the capabilities beyond either of the individual methods. We show that the unified MPC+DCBF controller displays favorable safety, performance, and closed-loop feasibility properties, and we demonstrate the utility of this controller via quadrupedal and quadrotor experiments for dynamic obstacle avoidance.

There are several avenues for future work that explore the current limitations of these methods. Firstly, although we can create probabilistic guarantees of safety using real-time controllers, these probability bounds are often quite loose. Future work will examine extensions beyond martingale concentration, Cantelli's, and Boole's inequalities that improve theoretical guarantees without sacrificing real-time capabilities. Secondly, significant work should be invested in defining the functions h given sensor output, developing methods for converting sensor information to scene understanding and finally to mathematical representations of safety. Thirdly, the methods presented in this work rely heavily on the use of system models $\mathbf{F}(\mathbf{x}, \mathbf{u})$. While the analysis here provides an understanding when this model is not known exactly, the degradation of this model knowledge has significant effects on the system guarantees. Future work will explore how these methods can be extended to learned dynamics models, particularly how the probabilistic guarantees can be retained for learned models and how safety constraints can be embedded in the training phase of reinforcement learning (RL) policies to improve optimality without the need for MPC's online horizon-based calculations. Finally, the examples presented here generally require solving non-convex optimization problems. While this work utilized sequential quadratic programming (SQP) and real-time iteration (RTI) methods, the solutions generated by these methods can be suboptimal or even unstable and additional work should investigate improved solution methods for the MPC+DCBF controllers.

REFERENCES

[1] J. C. Knight, "Safety critical systems: challenges and directions," in *International Conference on Software Engineering (ICSE)*, pp. 547–550, 2002.

[2] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control Barrier Functions: Theory and Applications," in *2019 18th European Control Conference (ECC)*, pp. 3420–3431, June 2019.

[3] F. Borrelli, A. Bemporad, and M. Morari, *Predictive control for linear and hybrid systems*. Cambridge University Press, 2017.

[4] S. Bansal, M. Chen, S. Herbert, and C. J. Tomlin, "Hamilton-jacobi reachability: A brief overview and recent advances," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, 2017.

[5] S. Kolathaya and A. D. Ames, "Input-to-State Safety With Control Barrier Functions," *IEEE Control Systems Letters*, vol. 3, Jan. 2019.

[6] R. Grandia, F. Jenelten, S. Yang, F. Farshidian, and M. Hutter, "Perceptive locomotion through nonlinear model-predictive control," *IEEE Transactions on Robotics*, vol. 39, no. 5, pp. 3402–3421, 2023.

[7] U. Rosolia and F. Borrelli, "Learning how to autonomously race a car: a predictive control approach," *IEEE Transactions on Control Systems Technology*, vol. 28, no. 6, pp. 2713–2719, 2019.

[8] A. Alan, A. J. Taylor, C. R. He, A. D. Ames, and G. Orosz, "Control barrier functions and input-to-state safety with application to automated vehicles," *IEEE Transactions on Control Systems Technology*, vol. 31, no. 6, pp. 2744–2759, 2023.

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

[10] "Supplemental video for this work," *https://shorturl.at/fQ7BW*.

[11] J. Löfberg, "Oops! i cannot do it again: Testing for recursive feasibility in mpc," *Automatica*, vol. 48, no. 3, pp. 550–555, 2012.

[12] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control Barrier Function Based Quadratic Programs for Safety Critical Systems," *IEEE Transactions on Automatic Control*, vol. 62, pp. 3861–3876, Aug. 2017.

[13] A. Agrawal and K. Sreenath, "Discrete Control Barrier Functions for Safety-Critical Control of Discrete Systems with Application to Bipedal Robot Navigation," in *Robotics: Science and Systems XIII*, July 2017.

[14] H. Ma, X. Zhang, S. E. Li, Z. Lin, Y. Lyu, and S. Zheng, "Feasibility enhancement of constrained receding horizon control using generalized control barrier function," in *2021 4th IEEE International Conference on Industrial Cyber-Physical Systems (ICPS)*, pp. 551–557, IEEE, 2021.

[15] R. Cosner, P. Culbertson, A. Taylor, and A. Ames, "Robust Safety under Stochastic Uncertainty with Discrete-Time Control Barrier Functions," in *Proceedings of Robotics: Science and Systems*, 2023.

[16] T. Gurriet, M. Mote, A. D. Ames, and E. Feron, "An Online Approach to Active Set Invariance," in *2018 IEEE Conference on Decision and Control (CDC)*, pp. 3592–3599, Dec. 2018. ISSN: 2576-2370.

[17] A. Singletary, K. Klingebiel, J. Bourne, A. Browning, P. Tokumaru, and A. Ames, "Comparative Analysis of Control Barrier Functions and Artificial Potential Fields for Obstacle Avoidance," Tech. Rep. arXiv:2010.09819, arXiv, Oct. 2020. arXiv:2010.09819 [cs, eess] type: article.

[18] M. F. Reis, A. P. Aguiar, and P. Tabuada, "Control barrier function-based quadratic programs introduce undesirable asymptotically stable equilibria," *IEEE Control Systems Letters*, vol. 5, no. 2, pp. 731–736, 2020.

[19] N. S. F. Doria, E. O. Freire, and J. C. Basilio, "An algorithm inspired by the deterministic annealing approach to avoid local minima in artificial potential fields," in *2013 16th International Conference on Advanced Robotics (ICAR)*, pp. 1–6, IEEE, 2013.

[20] P. Mestres, Y. Chen, E. Dall'anese, and J. Cortés, "Control barrier function-based safety filters: Characterization of undesired equilibria, unbounded trajectories, and limit cycles," *arXiv preprint arXiv:2501.09289*, 2025.

[21] S. Dean, A. J. Taylor, R. K. Cosner, B. Recht, and A. D. Ames, "Guaranteeing Safety of Learned Perception Modules via Measurement-Robust Control Barrier Functions," 2020.

[22] R. K. Cosner, A. W. Singletary, A. J. Taylor, T. G. Molnar, K. L. Bouman, and A. D. Ames, "Measurement-Robust Control Barrier Functions: Certainty in Safety with Uncertainty in State," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, (Prague, Czech Republic), pp. 6286–6291, IEEE, Sept. 2021.

[23] W. Langson, I. Chryssochoos, S. Raković, and D. Q. Mayne, "Robust model predictive control using tubes," *Automatica*, vol. 40, no. 1, pp. 125–133, 2004.

[24] J. Sieber, S. Bennani, and M. N. Zeilinger, "A system level approach to tube-based model predictive control," *IEEE Control Systems Letters*, vol. 6, pp. 776–781, 2021.

[25] A. Abate, M. Prandini, J. Lygeros, and S. Sastry, "Probabilistic reachability and safety for controlled discrete time stochastic hybrid systems," *Automatica*, vol. 44, 2008.

[26] M. Fränzle, E. M. Hahn, H. Hermanns, N. Wolovick, and L. Zhang, "Measurability and safety verification for stochastic hybrid systems," 2011. 14th Conference on Hybrid systems: computation and control.

[27] M. P. Chapman, R. Bonalli, K. M. Smith, I. Yang, M. Pavone, and C. J. Tomlin, "Risk-sensitive safety analysis using conditional value-at-risk," *IEEE Transactions on Automatic Control*, vol. 67, 2021.

[28] L. Lindemann, N. Matni, and G. J. Pappas, "Stl robustness risk over discrete-time stochastic processes," in *2021 60th IEEE Conference on Decision and Control (CDC)*, pp. 1329–1335, IEEE, 2021.

[29] J. Steinhardt and R. Tedrake, "Finite-time regional verification of stochastic non-linear systems," *The International Journal of Robotics Research*, vol. 31, pp. 901–923, June 2012.

[30] A. Mesbah, "Stochastic model predictive control: An overview and perspectives for future research," *IEEE Control Systems Magazine*, vol. 36, no. 6, pp. 30–44, 2016.

[31] S. X. Wei, A. Dixit, S. Tomar, and J. W. Burdick, "Moving Obstacle Avoidance: a Data-Driven Risk-Aware Approach," Mar. 2022. arXiv:2203.14913 [cs].

[32] A. Dixit, L. Lindemann, S. X. Wei, M. Cleaveland, G. J. Pappas, and J. W. Burdick, "Adaptive conformal prediction for motion planning among dynamic agents," in *Learning for Dynamics and Control Conference*, pp. 300–314, PMLR, 2023.

[33] J. Zeng, B. Zhang, and K. Sreenath, "Safety-critical model predictive control with discrete-time control barrier function," in *2021 American Control Conference (ACC)*, pp. 3882–3889, 2021.

[34] R. Grandia, A. J. Taylor, A. D. Ames, and M. Hutter, "Multi-layered safety for legged robots via control barrier functions and model predictive control," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8352–8358, IEEE, 2021.

[35] K. P. Wabersich and M. N. Zeilinger, "Linear Model Predictive Safety Certification for Learning-Based Control," in *2018 IEEE Conference on Decision and Control (CDC)*, pp. 7130–7135, Dec. 2018. ISSN: 2576-2370.

[36] J. Zeng, B. Zhang, and K. Sreenath, "Safety-Critical Model Predictive Control with Discrete-Time Control Barrier Function," in *2021 American Control Conference (ACC)*, pp. 3882–3889, IEEE, May 2021.

[37] J. Zeng, Z. Li, and K. Sreenath, "Enhancing feasibility and safety of nonlinear model predictive control with discrete-time control barrier functions," in *2021 60th IEEE Conference on Decision and Control (CDC)*, pp. 6137–6144, IEEE, 2021.

[38] S. Liu, J. Zeng, K. Sreenath, and C. A. Belta, "Iterative Convex Optimization for Model Predictive Control with Discrete-Time High-Order Control Barrier Functions," Tech. Rep. arXiv:2210.04361, arXiv, Oct. 2022. arXiv:2210.04361 [cs, eess, math] type: article.

[39] M. Cannon, B. Kouvaritakis, S. V. Raković, and Q. Cheng, "Stochastic tubes in model predictive control with probabilistic constraints," *IEEE Transactions on Automatic Control*, vol. 56, no. 1, pp. 194–200, 2011.

[40] B. Kouvaritakis, M. Cannon, S. V. Raković, and Q. Cheng, "Explicit use of probabilistic distributions in linear predictive control," *Automatica*, vol. 46, no. 10, pp. 1719–1724, 2010.

[41] H. Kushner, "Stochastic Stability and Control.," 1967. Academic Press.

[42] R. K. Cosner, P. Culbertson, and A. D. Ames, "Bounding stochastic safety: Leveraging freedman's inequality with discrete-time control barrier functions," *IEEE Control Systems Letters*, 2024.

[43] N. J. Sanket, C. M. Parameshwara, C. D. Singh, A. V. Kuruttukulam, C. Fermüller, D. Scaramuzza, and Y. Aloimonos, "EVDodgeNet: Deep Dynamic Obstacle Dodging with Event Cameras," *arXiv:1906.02919 [cs]*, Mar. 2020. arXiv: 1906.02919.

[44] D. Falanga, K. Kleber, and D. Scaramuzza, "Dynamic obstacle avoidance for quadrotors with event cameras," *Science Robotics*, vol. 5, p. eaaz9712, Mar. 2020.

[45] B. Lindqvist, S. S. Mansouri, A.-a. Agha-mohammadi, and G. Nikolakopoulos, "Nonlinear MPC for Collision Avoidance and Control of UAVs With Dynamic Obstacles," *IEEE Robotics and Automation Letters*, vol. 5, pp. 6001–6008, Oct. 2020. Conference Name: IEEE Robotics and Automation Letters.

[46] J. Breeden, K. Garg, and D. Panagou, "Control barrier functions in sampled-data systems," *IEEE Control Systems Letters*, vol. 6, pp. 367–372, 2022.

[47] M. Ahmadi, A. Singletary, J. W. Burdick, and A. D. Ames, "Safe Policy Synthesis in Multi-Agent POMDPs via Discrete-Time Barrier Functions," in *2019 IEEE Conference on Decision and Control (CDC)*.
[48] A. J. Taylor, V. D. Dorobantu, R. K. Cosner, Y. Yue, and A. D. Ames, "Safety of sampled-data systems with control barrier functions via approximate discrete time models," in *2022 IEEE 61st Conference on Decision and Control (CDC)*.
[49] S. Liu, J. Zeng, K. Sreenath, and C. A. Belta, "Iterative convex optimization for model predictive control with discrete-time high-order control barrier functions," in *2023 American Control Conference (ACC)*, pp. 3368–3375, IEEE, 2023.
[50] M. M. Minderhoud and P. H. Bovy, "Extended time-to-collision measures for road traffic safety assessment," *Accident Analysis Prevention*, vol. 33, no. 1, pp. 89–97, 2001.
[51] K. Vogel, "A comparison of headway and time to collision as safety indicators," *Accident analysis & prevention*, vol. 35, no. 3, pp. 427–433, 2003.
[52] V. Cavallo and M. Laurent, "Visual information and skill level in time-to-collision estimation," *Perception*, vol. 17, no. 5, pp. 623–632, 1988.
[53] D. N. Lee, "A theory of visual control of braking based on information about time-to-collision," *Perception*, vol. 5, no. 4, pp. 437–459, 1976.
[54] T. G. Molnar, R. K. Cosner, A. W. Singletary, W. Ubellacker, and A. D. Ames, "Model-Free Safety-Critical Control for Robotic Systems," *arXiv:2109.09047 [cs, eess, math]*, Sept. 2021. arXiv: 2109.09047.
[55] M. H. Cohen, R. K. Cosner, and A. D. Ames, "Constructive safety-critical control: Synthesizing control barrier functions for partially feedback linearizable systems," *IEEE Control Systems Letters*, 2024.
[56] Q. Nguyen and K. Sreenath, "Exponential control barrier functions for enforcing high relative-degree safety-critical constraints," in *2016 American Control Conference (ACC)*, pp. 322–328, 2016.
[57] W. Xiao and C. Belta, "High-order control barrier functions," *IEEE Transactions on Automatic Control*, vol. 67, no. 7, pp. 3655–3662, 2022.
[58] S. Bansal, M. Chen, S. Herbert, and C. J. Tomlin, "Hamilton-jacobi reachability: A brief overview and recent advances," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pp. 2242–2253, 2017.
[59] D. Q. Mayne, "Model predictive control: Recent developments and future promise," *Automatica*, vol. 50, no. 12, pp. 2967–2986, 2014.
[60] X. Tan and D. V. Dimarogonas, "On the undesired equilibria induced by control barrier function based quadratic programs," *Automatica*, vol. 159, p. 111359, 2024.
[61] E. D. Sontag and Y. Wang, "On characterizations of the input-to-state stability property," *Systems & Control Letters*, vol. 24, pp. 351–359, Apr. 1995.
[62] Y. Xiong, C. Zhou, X. Xiang, L. Wu, C. Zhu, Z. Liu, S. Suri, B. Varadarajan, R. Akula, F. Iandola, *et al.*, "Efficient track anything," *arXiv preprint arXiv:2411.18933*, 2024.
[63] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, E. Mintun, J. Pan, K. V. Alwala, N. Carion, C.-Y. Wu, R. Girshick, P. Dollár, and C. Feichtenhofer, "Sam 2: Segment anything in images and videos," 2024.
[64] G. Bahati, R. M. Bena, and A. D. Ames, "Dynamic safety in complex environments: Synthesizing safety filters with poisson's equation," in *Robotics: Science and Systems*, 2025 (to appear).
[65] G. Farnebäck, "Two-frame motion estimation based on polynomial expansion," in *Image Analysis*, pp. 363–370, Springer Berlin Heidelberg, 2003.
[66] O. So, A. Clark, and C. Fan, "Almost-sure safety guarantees of stochastic zero-control barrier functions do not hold," 2023.
[67] A. Clark, "Control Barrier Functions for Complete and Incomplete Information Stochastic Systems," in *2019 American Control Conference (ACC)*, pp. 2928–2935, July 2019. ISSN: 2378-5861.
[68] P. Culbertson, R. K. Cosner, M. Tucker, and A. D. Ames, "Input-to-State Stability in Probability," 2023. IEEE Conference on Decision and Control (CDC).
[69] C. Santoyo, M. Dutreix, and S. Coogan, "A barrier function approach to finite-time stochastic system verification and control," *Automatica*, vol. 125, p. 109439, Mar. 2021.
[70] G. Grimmett and D. Stirzaker, *Probability and Random Processes*. Oxford University Press, July 2020.
[71] J. Ville, "Etude critique de la notion de collectif," 1939.
[72] R. K. Cosner, I. Sadalski, J. K. Woo, P. Culbertson, and A. D. Ames, "Generative modeling of residuals for real-time risk-sensitive safety with discrete-time control barrier functions," May 2023.
[73] D. A. Freedman, "On tail probabilities for martingales," *the Annals of Probability*, pp. 100–118, 1975.
[74] L. Yang, B. Werner, R. Cosner, D. Fridovich-Keil, P. Culbertson, and A. Ames, "Shield: Safety on humanoids via cbfs in expectation on learned dynamics,"
[75] J. Schilliger, T. Lew, S. M. Richards, S. Hänggi, M. Pavone, and C. Onder, "Control barrier functions for cyber-physical systems and applications to nmpc," *IEEE robotics and automation letters*, vol. 6, no. 4, pp. 8623–8630, 2021.
[76] M. Vahs, C. Pek, and J. Tumova, "Belief control barrier functions for risk-aware control," *IEEE Robotics and Automation Letters*, 2023.
[77] S. Thrun, "Probabilistic robotics," *Communications of the ACM*, vol. 45, pp. 52–57, Mar. 2002.
[78] T. Lee, M. Leok, and N. H. McClamroch, "Geometric tracking control of a quadrotor UAV on SE(3)," in *49th IEEE Conference on Decision and Control (CDC)*, (Atlanta, GA), pp. 5420–5425, IEEE, Dec. 2010.
[79] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *2011 IEEE international conference on robotics and automation*, pp. 2520–2525, IEEE, 2011.
[80] S. Gros, M. Zanon, R. Quirynen, A. Bemporad, and M. Diehl, "From linear to nonlinear mpc: bridging the gap via the real-time iteration," *International Journal of Control*, vol. 93, no. 1, pp. 62–80, 2020.

APPENDIX

A. Proof of Prop. 1

Proof. Consider the input sequence that decreases safety as much as possible at each step while satisfying the constraints along the horizon of length N. According to the assumption in (5), the worst-case safety decrement at each step is $-\delta$ and, according to assumption (6), it is possible the controller to make h decrease by as little as $-N\epsilon$ along the trajectory. Thus, if safety decays by $-\delta$ for $k-1$ steps, the safety constraints are feasible at time k if:

$$0 \leq h(\mathbf{x}_k) - N\epsilon = h(\mathbf{x}_0) - \delta(k-1) - N\epsilon. \tag{43}$$

Solving for k yields: $k \leq \frac{h(\mathbf{x}_0)-N\epsilon}{\delta} + 1$. Since this feasibility is guaranteed for the worst-case input sequence, this bound holds regardless of the cost in the FTOCP problem. □

B. Proof of Thm. 2

Proof. Consider the input sequence that decreases safety as much as possible at each step while satisfying the constraints along the horizon of length N. According to the assumption in (5), the worst-case safety decrement when the constraints are all inactive is $-\delta$ and, according to assumption (6), it is possible for h to decrease by as little as $-N\epsilon$ along the trajectory. Thus, safety can continue decrementing by $-\delta$ until the sequence $[-\delta, -\epsilon, \ldots, -\epsilon]$ violates a DCBF constraint. Under this worst-case sequence, the first modifications occur at step $k+1$ when either the first or last constraint is violated:

$$-\delta < (\alpha - 1)(h(\mathbf{x}_0) - k_1\delta) \tag{44}$$

$$-\epsilon < (\alpha - 1)(h(\mathbf{x}_0) - (k_2 + 1)\delta - (N-2)\epsilon) \tag{45}$$

This is because the first constraint can modify the larger $-\delta$ decrement whereas the last constraint is the tightest since safety is positive and only decrease along the horizon.

Rearranging for the times k_1 and k_2 which trigger input modifications caused by (44) and (45) respectively yields:

$$k_1 > \frac{h(\mathbf{x}_0)}{\delta} - \frac{1}{1-\alpha} \tag{46}$$

$$k_2 > \frac{h(\mathbf{x}_0)}{\delta} - 1 - \left(N - 2 + \frac{1}{1-\alpha}\right)\frac{\epsilon}{\delta} \tag{47}$$

The case in (46) becomes true first since

$$\frac{1}{1-\alpha} \geq 1+\left((N-2)+\frac{1}{1-\alpha}\right)\frac{\epsilon}{\delta} = \frac{\delta+(N-2)\epsilon}{\delta+\epsilon} \quad (48)$$

was assumed in the theorem statement.

Thus, the system can apply $-\delta$ as a safety decrement $k_\delta = \max\left\{\left\lfloor \frac{h(\mathbf{x}_0)}{\delta} - \frac{1}{1-\alpha}\right\rfloor, 0\right\}$ times before the DCBF constraint becomes active. After this point, the input must be modified in order to satisfy the first constraint. If $k_\delta = 0$, then the input is modified at the first step.

Now that the first input must be modified to satisfy the first constraint, consider the new worst-case and maximally feasible planned safety decrement sequence $[(\alpha-1)h(\mathbf{x}_k), -\epsilon, \ldots, -\epsilon]$. Here assumption (6) ensures that the first decrement can be achieved. Using this sequence until $k-1$, the controller is feasible at step k if the sequence sequence of all $-\epsilon$ satisfies the last (and tightest) constraint:

$$\begin{aligned} -\epsilon &\geq (\alpha-1)\left(h(\mathbf{x}_k) - \epsilon(N-1)\right) && (49)\\ &= (\alpha-1)\left(\alpha^{k-1-k_\delta}h(\mathbf{x}_{k_\delta}) - \epsilon(N-1)\right) && (50)\\ &= (\alpha-1)\left(\alpha^{k-1-k_\delta}(h(\mathbf{x}_0) - k_\delta\delta) - \epsilon(N-1)\right) && (51)\end{aligned}$$

Rearranging for k yields:

$$k \leq \log_\alpha\left(\frac{\epsilon\left(\frac{1}{1-\alpha}+N-1\right)}{h(\mathbf{x}_0)-k_\delta\delta}\right) + 1 + k_\delta, \quad (52)$$

as claimed in the theorem statement. □

C. Proof of Thm. 7

Proof. At $i=0$, the safety constraint implies

$$\begin{aligned} 0 &\geq -h(\boldsymbol{\xi}_1) + \alpha h(\boldsymbol{\xi}_0) && (53)\\ &= -(\mathbf{A}\boldsymbol{\xi}_0 + \mathbf{B}\boldsymbol{\nu}_0^*(\widehat{\mathbf{x}}_{k|k})) + \alpha h(\boldsymbol{\xi}_0) && (54)\\ &= -h(\mathbf{A}\widehat{\mathbf{x}}_{k|k} + \mathbf{B}\mathbf{u}_k) + \alpha h(\widehat{\mathbf{x}}_{k|k}) && (55)\\ &= -h(\mathbf{A}\widehat{\mathbf{x}}_{k|k} + \mathbf{B}\mathbf{u}_k + \mathbb{E}[\mathbf{d}_k]) + \alpha h(\widehat{\mathbf{x}}_{k|k}) && (56)\\ &= -h(\mathbb{E}[\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}_k + \mathbf{d}_k \mid \mathscr{G}_k]) + \alpha h(\widehat{\mathbf{x}}_{k|k}) && (57)\\ &= -h(\mathbb{E}[\mathbf{x}_{k+1}|\mathscr{G}_k]) + \alpha h(\mathbf{x}_k) && (58)\\ &= -h(\mathbb{E}[\mathbb{E}[\mathbf{x}_{k+1}|\mathscr{G}_{k+1}]|\mathscr{G}_k]) + \alpha h(\widehat{\mathbf{x}}_{k|k}) && (59)\\ &= -h(\mathbb{E}[\widehat{\mathbf{x}}_{k+1|k+1}|\mathscr{G}_k]) + \alpha h(\widehat{\mathbf{x}}_{k|k}) && (60)\\ &\geq -\mathbb{E}[\, h(\widehat{\mathbf{x}}_{k+1|k+1}) \mid \mathscr{G}_k\,] + \alpha h(\widehat{\mathbf{x}}_{k|k}) && (61)\\ &\implies \mathbb{E}[\, h(\widehat{\mathbf{x}}_{k+1|k+1}) \mid \mathscr{G}_k] \geq \alpha h(\widehat{\mathbf{x}}_{k|k}). && (62)\end{aligned}$$

Here, (53) is enforced by the controller's safety constraint, (54) is enforced by the controller's dynamics constraint, (55) is a result of using the first planned action $\boldsymbol{\nu}_0^*(\widehat{\mathbf{x}}_{k|k})$ as the current input $\mathbf{u}_k$, (56) is a result of the zero mean assumption on the disturbance $\mathbf{d}_k$, (57) is through the definition of $\widehat{\mathbf{x}}_{k|k}$ along with the linearity of expectation and the full-knowledge of $\mathbf{u}_k$ and $\mathbf{d}_k$ given $\mathscr{G}_k$. Next, (59) is a result of the assumed linearity of the measurement function and the tower rule, and (60) is by definition of $\widehat{\mathbf{x}}_{k+1|k+1}$. Finally, (61) is a result of Jensen's inequality since h is to be convex and (62) comes from rearranging terms. □

D. Proof of Thm. 8

Before proving Thm. 8, we provide Cantelli's inequality as a Lemma for reference.

Lemma 1 (Cantelli's Inequality [70])**.** *For any real-valued random variable X, $\mathbb{P}\{X - \mathbb{E}[X] \leq -\lambda\} \leq \frac{\sigma^2}{\sigma^2+\lambda^2}$ for any $\lambda > 0$ and where $\sigma \geq 0$ is the variance of X .*

Next we provide the proof of Thm. 8 which leverages this lemma to extend belief-based safety guarantees to safety guarantees for the true state of the system even when relying on uncertain measurements.

Proof. Consider the events :

$$\begin{aligned} A &= \{h(\widehat{\mathbf{x}}_{k|k}) < 0 \text{ for some } k \leq K\}, && (63)\\ \overline{A} &= \{\mathbb{E}[h(\mathbf{x})|\mathscr{G}_k] < 0 \text{ for some } k \leq K\}, && (64)\\ B_k &= \{h(\mathbf{x}_k) \leq \mathbb{E}[h(\mathbf{x}_k) \mid \mathscr{G}_k] - \delta_C\}. && (65)\end{aligned}$$

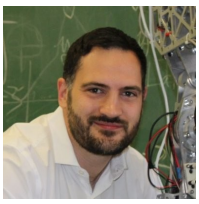
Since h is convex, Jensen's inequality and the definition $\widehat{\mathbf{x}}_{k|k} \triangleq \mathbb{E}[\, \mathbf{x}_k \mid \mathscr{G}_k\,]$ establish the containment $\overline{A} \subset A$ and the probability bound $\mathbb{P}\{\overline{A}\} \leq \mathbb{P}\{A\}$. Next, consider the combined event: $U = \left\{\overline{A} \cup \left\{\cup_{k=0}^K B_k\right\}\right\}$. Note that U^c, the complement of U, is the event that $\mathbb{E}[h(\mathbf{x})|\mathscr{G}_k] \geq 0$ and $h(\mathbf{x}_k) > \mathbb{E}[h(\mathbf{x}_k)|\mathscr{G}_k] - \delta$ for all $k \in \{0,\ldots,K\}$. Thus U^c is a sufficient condition for $h(\mathbf{x}_k) \geq -\delta_C$ for all $k \in \{0,\ldots,K\}$ and U is a necessary condition for $h(\mathbf{x}_k) < -\delta_C$ for some $k \leq K$. From this, we can establish the probability bounds:

$$\begin{aligned} &\mathbb{P}\{h(\mathbf{x}_k) \leq -\delta_C \text{ for some } k \leq K\} \leq \mathbb{P}\{U\} && (66)\\ &\leq \mathbb{P}\{\overline{A}\} + \sum_{i=0}^K \mathbb{P}\{B_i\} && (67)\\ &\leq \mathbb{P}\{A\} + \sum_{i=0}^K \mathbb{P}\{B_i\} \leq \epsilon + \sum_{i=0}^K \mathbb{P}\{B_i\} && (68)\\ &\leq \epsilon + (1+N)\left(\frac{\sigma_h^2}{\sigma_h^2+\delta^2}\right) && (69)\end{aligned}$$

The first bound is due to the fact that U is a necessary condition for the true state to be unsafe with respect to $\mathcal{C}_{\delta_C}$ for some $k \leq K$, the second bound is an application of Boole's inequality [70], the third bound is due to the containment $\overline{A} \subset A$, the fourth bound is from the assumed bound on $P_u(K, \widehat{\mathbf{x}}_0) \leq \epsilon$, and the final bound is an application of Cantelli's Inequality (Lem. 1) with the assumption that $\text{Var}(h(\mathbf{x}_k)|\mathscr{G}_k) \leq \sigma_h$. □

Ryan K. Cosner received a B.S. in mechanical engineering from the University of California, Berkeley in 2019 and a M.S. in mechanical engineering from the California Institute of Technology in 2021. He is currently working towards the Ph.D. degree in mechanical engineering at the California Institute of Technology. In 2026 he will begin his faculty career as an Assistant Professor in mechanical engineering at Tufts University, Medford, Massachusetts. His research interests include machine learning, nonlinear control theory, and robotics.

Ryan M. Bena received the B.S. defree in mechanical engineering from the University of California, Berkeley, CA, USA, in 2012 and the M.S. degree in aerospace engineering from the University of Southern California (USC), Los Angeles, CA, USA, in 2018. From 2012 to 2019, he was an Aerospace Engineer with the US Air Force. He received his Ph.D. degree in aerospace engineering from USC in 2024, working in the Autonomous Microrobotic Systems Laboratory and the Dynamic Robotics and Control Laboratory. He is currently a Postdoctoral Scholar at the California Institute of Technology working in the Advanced Mechanical Bipedal Experimental Robotics Lab. His research interests include safety-critical controller design and analysis, collision avoidance for aerial robotics, and optimization-based control techniques.



Aaron D. Ames received a B.S. degree in mechanical engineering and a B.A. in mathematics from the University of St. Thomas in 2001, and he received a M.A. in mathematics and a Ph.D. in electrical engineering and computer sciences from the University of California, Berkeley in 2006.

He is a Bren Professor of Mechanical and Civil Engineering and Control and Dynamical Systems at Caltech. Prior to joining Caltech in 2017, he was an Associate Professor at Georgia Tech in the Woodruff School of Mechanical Engineering and the School of Electrical & Computer Engineering. He was a Postdoctoral Scholar in Control and Dynamical Systems at the California Institute of Technology from 2006 to 2008 and began his faculty career at the Texas A&M University in 2008. His research interests span the areas of robotics, nonlinear control and hybrid systems, with a special focus on applications to bipedal robotic walking both formally and through experimental validation. His lab designs, builds and tests novel bipedal robots, humanoids and prostheses with the goal of achieving human-like bipedal robotic locomotion and translating these capabilities to robotic assistive devices.

Dr. Ames received the 2005 Leon O. Chua Award for achievement in nonlinear science, the 2006 Bernard Friedman Memorial Prize in Applied MAthematics, the NSF CAREER award in 2010, and the 2015 Donald P. Eckman Award.