Safe Collective Control under Noisy Inputs and Competing Constraints via Non-Smooth Barrier Functions*

Clinton Enwerem[1]

John S. Baras[1]

Abstract—We consider the problem of safely coordinating ensembles of identical autonomous agents to conduct complex missions with conflicting safety requirements and under noisy control inputs. Using non-smooth control barrier functions (CBFs) and stochastic model-predictive control as springboards, and by adopting an extrinsic approach where the ensemble is treated as a unified dynamic entity, we devise a method to synthesize safety-aware control inputs for uncertain collectives. Drawing upon stochastic CBF theory and recent developments in Boolean CBF composition, our method proceeds by smoothing a Boolean-composed CBF and solving a stochastic optimization problem where each agent's forcing term is restricted to the affine subspace of control inputs certified by the combined CBF. For the smoothing step, we employ a polynomial approximation scheme, providing evidence for its advantage in generating more conservative yet sufficiently-filtered control inputs than the smoother but more aggressive equivalents produced from an approximation technique based on the log-sum-exp function. To further demonstrate the utility of the proposed method, we present an upper bound for the expected CBF approximation error, along with results from simulations of a single-integrator collective under velocity perturbations. Lastly, we compare these results with those obtained using a naive state-feedback controller lacking safety filters.

Index Terms—control barrier functions, multi-agent systems, safety-critical control, stochastic model-predictive control

I. INTRODUCTION

Intelligent cyber-physical systems — such as teams of aerial robots or autonomous vehicle platoons — face a fundamental challenge: they must safely navigate complex environments while efficiently completing their objectives. This challenge is non-trivial because the sub-tasks that constitute the ensemble's objective may often be varied and conflicting, making it imperative to conduct appropriate trade-offs between tasks. The inherent uncertainty in each agent's model and operational vicinity also imposes additional constraints on the multiagent system, making safe multiagent control an acute problem. Control barrier functions (CBFs) — an arguably de facto technique for safety-critical controller synthesis — have excelled in many applications in autonomous driving, robotics [1], and allied fields. However, conventionally, most of the associated results in CBF-enabled control are founded on deterministic assumptions, making it challenging to apply them in uncertain settings. Moreover, designing CBFs for complex missions with competing safety requirements
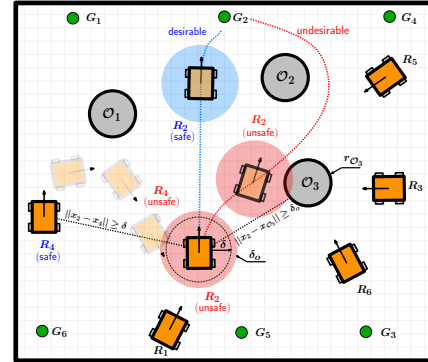
Fig. 1: **Motivating example**: A multi-agent reach-avoid mission with complex safety requirements. Here, the dynamic agents (depicted as mobile robots in orange (R_i)) must navigate to specified goal positions (green circles (G_i)) while avoiding multiple obstacles (gray circles $(\mathcal{O}_i)$) and inter-agent collisions. Figure annotations describe associated notions of safety, with agent arrows indicating direction of travel.

typically entails a composition of distinct barrier functions, an operation that may render the resulting function non-smooth, so that traditional results of local Lipschitz continuity fail to apply.

In composing barrier functions, one also runs the risk of potentially losing safe controller existence guarantees. Again, this is due to the fact that many CBF-derived control synthesis algorithms generate filtered control signals by solving quadratic programs (QPs) — or some optimization problem with state and control costs — via gradient descent or other derivative-based techniques. As such, even if one could come up with CBFs that sufficiently capture all the given safety requirements at hand, there might not exist any control sequence for which the constraints of the resulting CBF quadratic program are satisfied [2]. Moreover, while an existence of non-negative Lagrange multipliers for each CBF constraint has been shown to guarantee feasibility of the resulting QP [2], such conditions are for the deterministic case and do not apply in the stochastic setting.

Regarding CBF composition, a number of studies have presented techniques for capturing diverse safety requirements in the deterministic setting (see [1], [2]). These methods typically involve computing the pointwise maximum (resp. minimum) of all barrier functions that encode the safety requirements, equivalent to an OR (resp. AND) Boolean logical operation. Because the foregoing operations yield non-smooth functions, however, synthesizing safe controllers

TABLE I: NOTATION & NOMENCLATURE

Symbol	Description
$x_i \in \mathfrak{X} \subset \mathbb{R}^d$	Node- or agent-level state variable
$u_i \in \mathfrak{U} \subset \mathbb{R}^m$	Node-level forcing term
ζ_d	Vector $[\zeta, \zeta, \dots \zeta]^\top \in \mathbb{R}^d$
$\mathbb{I}_d$	$d \times d$ identity matrix
$\|x_i\|_p$	p-norm (set to two where p is omitted)
$[A_{ij}]$; $\mathrm{tr}(A)$; $A^\top$	Matrix (A) with elements A_{ij}; trace; transpose
$\|H\|_{L^1[a,b]}$	$\int_a^b |H(\tau)|d\tau$, H Lipschitz continuous on $[a, b]$
$\Lambda([\star_i]_{i \in \mathcal{I}})$	Block diagonal matrix with ith block $\star_i$
$\mathbb{R}_+$; $\mathbb{Z}_{\geq 0}$, $\mathbb{Z}_+$	Set of positive reals; non-negative & positive integers
C^k	Set of k-times continuously-differentiable functions
$f(\cdot)$; $g(\cdot)$	Drift and control vector fields
$\otimes$; $\alpha(\cdot)$;	Kronecker product; Extended κ-class function
$h(\cdot)$	Control barrier function (CBF)
$L_f h(\cdot)$; $L_g h(\cdot)$;	Lie derivatives of h along f and g
$\mathbb{E}$; $\Pr(E_i)$	Expectation operator; Probability of event E_i

becomes a challenge, since certifying control inputs via barrier functions requires computing the latter's Lie derivatives along the control and drift vector fields. Despite these challenges, recent work [2] has demonstrated that safe control inputs can still be synthesized by approximating the non-smooth CBF (NCBF) via the log-sum-exp (LSE) function. They further posit that the resulting CBF is bounded tightly by asymptotically-decaying terms containing the NCBF, under such a smoothing scheme. In the stochastic case and for multiagent systems, however, no such analysis or guarantees exist, to the best of our knowledge. With the recent emergence of results extending CBF theory to stochastic systems [3], we now have the tools to address the foregoing research gaps.

A. Scientific Contributions

Leveraging new-found ideas on Boolean composition of CBFs with smooth approximation, we report results on synthesizing safe controllers for ensembles under complex tasking and noisy inputs. In contrast to similar work [2], however, we apply a polynomial smoothing function in the smoothing step and synthesize control inputs via a *stochastic* quadratic program. Finally, we present two numerical examples to demonstrate the proposed technique and validate the approach through comparisons with a naive state-feedback controller and an LSE-approximated CBF-enabled control scheme.

B. Outline

Notations that appear throughout the paper have been organized on Table I. Occasionally, several notations appearing there will contain a subscript, a superscript, or both, but they will be elucidated if unclear from the context. In the sections that follow, we briefly discuss the mathematical elements upon which our work is founded, viz. non-smooth CBFs (Section II), Boolean composition (Section III), and stochastic model-predictive control (SMPC) (Section V). These sections set the background for our theoretical results (Section VI), with numerical simulations (Section VIII), associated findings (Section IX), the article's conclusion (Section X), and the appendices appearing last.

II. NON-SMOOTH CONTROL BARRIER FUNCTIONS

To preface our discussion on NCBFs, we consider a collective comprising N identical agents, each evolving according to nonlinear control-affine dynamics with noisy inputs given by the following equation (with subscript i denoting the ith agent):

$$\dot{x}_i = f(x_i) + g(x_i)u_i + K_w(x_i)w_i, \quad (1)$$

with admissible control signals taking values in the set, $\mathfrak{U}$, of piecewise continuous and absolutely-integrable functions. $w_i \in \mathbb{R}^d$ represents the realization of an unknown disturbance (with a known probability distribution) that is scaled by a state-dependent and positive-definite gain matrix, $K_w \in \mathbb{R}^{d \times d}$. The equation in (1) coincides with a stochastic differential equation comprising a *diffusion* term equal to $K_w(x_i)$ and an affine-in-control *drift* term. A few useful definitions follow.

Definition 1 (*Safe Set*): Let x_i satisfy (1). The safe set, $\mathfrak{C}$, with non-empty interior, $\mathrm{Int}(\mathfrak{C})$, and boundary, $\partial \mathfrak{C}$, is the zero super-level set of the sufficiently smooth map, $h(\cdot) : \mathfrak{C} \to \mathbb{R}$, that is, $\mathfrak{C} = \{x_i \in \mathfrak{X} : h(x_i) \geq 0\}$.

Definition 2 (*Control Barrier Function [4]*): Assume agent-level dynamics of the form (1), with f and g locally Lipschitz, and suppose there exists a function, α, belonging to the family of locally Lipschitz continuous extended $\mathcal{K}$-class functions[1] on $\mathbb{R}^d$. The C^2 function, $h : \mathfrak{C} \subset \mathfrak{X} \to \mathbb{R}_+ \cup \{0\}$, is a control barrier function[2] for the system in (1) if there is a u_i for which the following property holds for all $x_i \in \mathfrak{X}$ satisfying $h(x_i) > 0$, with $\gamma > 0$:

$$L_f h + L_g h u_i + \frac{1}{2}\mathrm{tr}\left(K_w^\top \frac{\partial^2 h}{\partial x_i^2} K_w\right) \geq -\alpha(\gamma, h(x_i)). \quad (2)$$

The family of controllers satisfying the foregoing condition are important because they establish the forward invariance of $\mathfrak{C}$, for $\mathfrak{C}$ compact for all $t \geq 0$. In particular, under certain conditions on the drift and diffusion terms corresponding to dh/dt (the exact derivative of h), then $\Pr(x_i(t) \in \mathfrak{C} \,\forall\, t \geq 0) = 1$, provided that $x_i(0) \in \mathrm{Int}(\mathfrak{C})$ (see the recent paper [3], Theorem 1, and Proposition 2 with its proof for the details). This result thus enables the synthesis of safe controllers via stochastic quadratic programming, set here as the expectation minimization problem

$$\min_{u_i \in \mathfrak{U}} J = \mathbb{E}\left[\sum_{\tau=0}^{T-1} c\left(x_i(\tau), u_i(\tau)\right)\right] \quad (3)$$

$$\text{s.t. } \Pr\left(L_f h + L_g h u_i + \frac{1}{2}\mathrm{tr}\left(K_w \frac{\partial^2 h}{\partial x_i^2} K_w\right) \geq -\alpha(\gamma, h)\right)$$
$$\geq 1 - \delta_h,$$

[1] An extended $\mathcal{K}$-class function is a strictly increasing function that vanishes at the origin.

[2] To be precise, h as delineated in Definition 2 refers to the special class of CBFs known as zeroing or *zero control* CBFs [5], so named for their property of vanishing at the boundary of the safe set, i.e., $\partial \mathfrak{C}$. They are distinct from *reciprocal* CBFs (RCBFs) that instead grow unbounded as the system state approaches $\partial \mathfrak{C}$.

for $\delta_h \in (0, 1)$ small. The symbol $T \in \mathbb{Z}_+$ represents some time horizon, and $c(x_i(t), u_i(t))$ is a positive-definite cost function (typically made up of quadratic terms in u_i and x_i) chosen to capture a desired weighting of the control inputs corresponding to each component of x_i. With this primary setting in place, we transition to discussing Boolean composition of CBFs.

III. COMPOSING CBFs VIA BOOLEAN LOGIC

Composing barrier certificates for complex safety requirements via Boolean logical operations appears in several studies in the literature [2], [1], [6]. Besides Boolean logic, however, it is theoretically possible to enforce global safety requirements through distinct CBFs by passing them as independent constraints to a QP. Nevertheless, non-smooth compositions — comprising some combination of maximization, minimization, or negation — are standard, for reasons already alluded to. These operations are respectively analogous to the Boolean logical operators of OR ($\vee$), AND ($\wedge$), and NOT ($\neg$), and the set-theoretic operations of union ($\bigcup$), intersection ($\bigcap$), and complement (^{c}). More concretely, suppose we have a finite number of independent safety constraints completely indexed by the set, $\mathcal{I}$. We are interested in computing a common CBF with attendant control inputs that drive each agent in the ensemble only to configurations that respect all $|\mathcal{I}|$ safety constraints. Denote by $\mathrm{x} = [[x_i^\top]_{i=1}^N]^\top \in \mathbb{R}^{Nd}$ and $\mathrm{u} = [[u_i^\top]_{i=1}^N]^\top \in \mathbb{R}^{Nm}$ the ensemble analogs of the state and control vectors. The ensemble-level safe set, hereafter denoted as $\mathfrak{C}_\Sigma$, with corresponding (zeroing) barrier function, h_Σ, is the set

$$\mathfrak{C}_\Sigma = \{\mathrm{x} \in \mathbb{R}^{Nd} : h_\Sigma(\mathrm{x}) \geq 0\} \tag{4}$$

$$= \bigcap_{|\mathcal{I}|} \mathfrak{C}_\star, \text{ for } \star = 1, 2, \ldots, |\mathcal{I}| \tag{5}$$

$$= \left\{\mathrm{x} \in \mathbb{R}^{Nd} : \min_{\star \in \mathcal{I}} h_\star(\mathrm{x}) \geq 0\right\}, \tag{6}$$

where h_Σ is a conjunction of all independent CBFs, over $\mathcal{I}$, given by the expression:

$$h_\Sigma(\mathrm{x}) = \bigwedge_{\star=1}^{|\mathcal{I}|} h_\star(\mathrm{x}). \tag{7}$$

Similarly, one can construct the OR analog of (5) (resp. (7)) through a union (resp. disjunction) over $\mathcal{I}$. Even more complicated safety requirements can be captured via successive Boolean logical compositions [1], [2], as we illustrate in Section VII.

IV. APPROXIMATING NCBFs VIA POLYNOMIAL SMOOTHING

Since h_Σ is non-smooth and, thus, unfit for quadratic optimization, in this section, we discuss a parametric polynomial smoothing technique for finding an approximation of h_Σ, hereafter denoted as $\hat{h}_\Sigma$. Rewriting $h_\Sigma = \min_{\star \in \mathcal{I}} h_\star$ (with x suppressed) as

$$h_\Sigma(\mathrm{x}) = -\frac{1}{2}\big(\ell\phi(\ell) - \ell'\big), \tag{8}$$

where the expression on the right-hand side of (8) follows from a known equivalent expression for the min. function [7], with $\ell = \sum_{\star \in \mathcal{I}\setminus\{1\}} h_\star(\mathrm{x}) - h_1(\mathrm{x})$ and $\ell' = \sum_{\star \in \mathcal{I}} h_\star(\mathrm{x})$. The function $\phi : \mathbb{R} \to \{-1, 1\}$ characterizes the discontinuity of h_Σ and is given by

$$\phi(\ell) = \begin{cases} 1, & \ell \geq 0 \\ -1, & \ell < 0. \end{cases} \tag{9}$$

Smoothing h_Σ thus reduces to the task of redefining (9) to address the jump discontinuity at $\ell = 0$. In particular, if we redefine ϕ (hereafter denoted as $\hat{\phi}(\ell, \beta)$ to make its distinction from (9) clear) as

$$\hat{\phi}(\ell, \beta) = \begin{cases} 1, & \ell > \beta \\ M_k(\ell, \beta), & -\beta \leq \ell \leq \beta \\ -1, & \ell < -\beta, \end{cases} \tag{10}$$

where β is the smoothing parameter and $M_k(\ell, \beta) = a_0(\beta) + a_1(\beta)\ell + a_2(\beta)\ell^2 + \cdots + a_p(\beta)\ell^p$, we arrive at the following expression for $\hat{h}_\Sigma(\mathrm{x})$:

$$\hat{h}_\Sigma(\mathrm{x}) = -\frac{1}{2}\big(\ell\hat{\phi}(\ell, \beta) - \ell'\big). \tag{11}$$

The choice of p in M_k depends on the desired level of continuous differentiability for $\hat{h}_\Sigma$, i.e., $p = 2k + 1, k \in \mathbb{Z}_+$, for $\hat{h}_\Sigma \in C^k$. Denoting the ensemble drift and control vector fields respectively as F and G, we can write the corresponding ensemble Lie derivatives (omitting higher derivatives for brevity) as

$$L_F\hat{h}_\Sigma(\mathrm{x}) = \frac{\partial \hat{h}_\Sigma}{\partial \mathrm{x}} F(\mathrm{x}) \text{ and } L_G\hat{h}_\Sigma(\mathrm{x}) = \frac{\partial \hat{h}_\Sigma}{\partial \mathrm{x}} G(\mathrm{x}),$$

where $F(\mathrm{x})$ and $G(\mathrm{x})$ are given respectively by

$$\begin{bmatrix} f^\top(x_1), & f^\top(x_2), & \ldots, & f^\top(x_N) \end{bmatrix}^\top \in \mathbb{R}^{Nd} \text{ and}$$
$$\bigwedge\big(\begin{bmatrix} g(x_1), & g(x_2), & \ldots, & g(x_N) \end{bmatrix}\big) \in \mathbb{R}^{Nd \times Nm},$$

with

$$\frac{\partial \hat{h}_\Sigma}{\partial \mathrm{x}} = -\frac{1}{2}\Bigg[\bigg(\sum_{\star \in \mathcal{I}\setminus\{1\}} \frac{\partial h_\star(\mathrm{x})}{\partial \mathrm{x}} - \frac{\partial h_1(\mathrm{x})}{\partial \mathrm{x}}\bigg)\hat{\phi}(\ell, \beta) \tag{12}$$
$$+ \bigg(\sum_{\star \in \mathcal{I}\setminus\{1\}} h_\star(\mathrm{x}) - h_1(\mathrm{x})\bigg)\frac{\partial \hat{\phi}(\ell, \beta)}{\partial \mathrm{x}} - \sum_{\star \in \mathcal{I}} \frac{\partial h_\star(\mathrm{x})}{\partial \mathrm{x}}\Bigg].$$

V. SMPC WITH POLYNOMIAL-APPROXIMATED NCBFs

We segue now to generating control commands for each agent via SMPC. In particular, having constructed a valid and smooth CBF from the given independent safety constraints, we can now synthesize control inputs by solving the (ensemble-level) stochastic quadratic program

$$\min_{\mathrm{u} \in \mathbb{R}^{Nm}} \mathbb{E}\left[\sum_{\tau=0}^{\mathrm{T}_\mathrm{u}-1} ||\mathrm{u}(\tau) - \mathrm{u}_d(\tau)||^2 \cdot \big(1 - \mathbb{1}_{\{||\mathrm{x}-\mathrm{x}_g|| \leq \varepsilon_g\}}\big)\right]$$

$$\text{s.t. } \Pr\big(L_F\hat{h}_\Sigma(\mathrm{x}) + L_G\hat{h}_\Sigma(\mathrm{x})\mathrm{u}(\tau)$$
$$+ \frac{1}{2}\mathrm{tr}\big((K_w \otimes I_N)^\top \frac{\partial^2 \hat{h}_\Sigma(\mathrm{x})}{\partial \mathrm{x}^2}(K_w \otimes I_N)\big)$$
$$+ \alpha(\gamma, \hat{h}_\Sigma(\mathrm{x})) \geq 0\big) \geq 1 - \delta_h \tag{13a}$$
$$- u_{\max} \cdot 1_{Nm} \leq \mathrm{u}(\tau) \leq u_{\max} \cdot 1_{Nm}, \tag{13b}$$

with $u_d \in \mathbb{R}^{Nm}$ denoting the nominal and unsafe ensemble-level control input, and where T_u is the MPC control horizon. We include an indicator function, $\mathbb{1}_{\{||x - x_g|| \leq \varepsilon_g\}}$, to prevent the accumulation of costs after all agents have reached positions within a goal region defined by the closed ball centered at x_g with radius, ε_g. The agents' respective goal locations are given by each element of the goal vector, $x_g = [x_{i,g}^\top]^\top$. Finally, we also place bounds on the control inputs to respect physical actuation limits (13b), characterized by a positive scalar, $u_{\max}$. Due to the hard input constraints and chance constraints, SMPC problems of the form (13) are generally intractable [8]. Thus, a few assumptions (presented next) are in order, to guarantee the problem's feasibility.

Assumptions:

(A1) *The dynamical system for the ith agent is fully observable, so that access to full state information is assumed.*

(A2) *The disturbance vector is zero-mean Gaussian with bounded covariance, i.e., $w_i \sim \mathcal{N}(0_d, [\Sigma_{w_{ij}}])$, with $0 < \Sigma_{w_{ii}} \leq \sigma_{\max} \ \forall \ i \in [1, 2, \ldots, d]$, where $\sigma_{\max} \in \mathbb{R}_+$ represents a bound on the noise. This assumption is important to argue for the convexity of the set of safe controls using the log-concavity of the multivariate normal distribution.*

(A3) *u_d is parameterized by the disturbance as $u_d(t) = u_f(t) + (K_w \otimes I_N)w$, with $w = [[w_i^\top]_{i=1}^N]^\top \in \mathbb{R}^{Nd}$ denoting the ensemble disturbance vector and where u_f is a simple feedback law (see Section VII for its form).*

These assumptions together with a restriction of the control inputs to an invariant set (13a) and the quadratic cost guarantee the feasibility of (13) [8], since they enable a transformation of the chance constraints into convex constraints via a logarithm operation. One can then solve the resulting QP using optimization software with quadratic programming support.

VI. THEORETICAL RESULTS

A. Upper Bound on the Expected CBF Approximation Error

To quantify the accuracy of the CBF approximation with respect to the smoothing parameter, we introduce the following lemma, which we will invoke in the proposition to follow.

Lemma 1: *Let the functions $\hat{\phi}(\ell, \beta)$ and ϕ be as defined in (10) and (9), respectively. Then, for $\beta > 0$ and on $\mathcal{I}_\beta = [-\beta, \beta]$, the following inequality holds:*

$$\mathbb{E}\left[||\phi(\ell) - \hat{\phi}(\ell, \beta)||_{L^1[a,b]}\right] = \frac{15}{4}\beta. \tag{14}$$

Proof. Since $\hat{\phi}(\ell, \beta)$ and $\phi(\ell)$ coincide for $\ell < -\beta$ and $\ell > \beta$, when $\beta > 0$, and the $L^1[a,b]$ norm is equivalent to the integral, $\int_{-\beta}^{\beta} |\phi(\ell) - \hat{\phi}(\ell, \beta)| d\ell$, on $\mathcal{I}_\beta$, it is sufficient to prove that (14) holds on $\mathcal{I}_\beta$. By straightforward integration on $\mathcal{I}_\beta$, with $k = 2$ in (10) (see the corresponding coefficients for $M_2(\ell, \beta)$ on Table II), and using the definition of the $L^1[a,b]$ norm and the fact that $\mathbb{E}[c] = c$, if c is a constant, we arrive at the expression in (14). ∎

Proposition 1: *Let $\hat{h}_\Sigma$ be defined as in (11). Define the CBF approximation error as $\hat{e} = \hat{h}_\Sigma - \min_{\star \in \mathcal{I}} h_\star$. Then, under the polynomial smoothing scheme, with $k = 2$ in (10), we have the following inequality for the expected CBF approximation error, $\mathbb{E}\left[||\hat{e}||_{L^1[a,b]}\right]$:*

$$\mathbb{E}\left[||\hat{e}||_{L^1[a,b]}\right] \leq \frac{15\beta^2}{8}. \tag{15}$$

Proof. See Appendix A. ∎

Remark 1: *Since the upper bound on the CBF approximation error consists of terms quadratic in β, successively higher values of β may lead to smoother approximations under the polynomial smoothing scheme at the cost of losing information about the original function. Conversely, the approximations for β values on $(0, \frac{2\sqrt{2}}{\sqrt{15}})$ may be comparatively less smooth while being less prone to distortion. It turns out, in fact, that no perceptible change in the smoothing performance is recorded for β values on $(0, 1]$ (see Fig. 2), with safe controls assured nonetheless (see Fig. 3).*

B. Forward Invariance of the Ensemble-Level Safe Set

Considering that our method relies on the synthesis of control inputs via a polynomial-smoothed NCBF (i.e., $\hat{h}_\Sigma$), that itself defines the ensemble-level safe set ($\mathfrak{C}_\Sigma$), we need to verify that $\hat{h}_\Sigma$ is indeed a valid CBF. The following result from [3] specifies conditions that guarantee almost-sure safety via stochastic *zeroing* CBFs that will be useful in establishing the validity of $\hat{h}_\Sigma$ in our next result (Proposition 2).

Theorem 1 (*Conditions for Almost-Sure Safety via Zeroing CBFs (Corollary 11 in [3])*): *Assume there exists a function $h : \mathfrak{X} \to \mathbb{R}$ and a $\mathcal{K}$-class function, $\alpha_\star$, where, for all $x_i \in \mathfrak{X}$ satisfying $h(x_i) > 0$, there exists a control $u_i \in \mathfrak{U}$ such that[3]*

$$\tilde{\mu}_i - \frac{\tilde{\sigma}_i^2}{h(x_i)} \geq -h^2(x_i)\alpha_\star(h), \tag{16}$$

with $\tilde{\mu}_i$ and $\tilde{\sigma}_i$, respectively denoting the drift and diffusion terms corresponding to dh/dt, with x_i evolving according to (1). Then, for all $t \geq 0, \Pr(x(t) \in \mathfrak{C}) = 1$, provided that $x(0) \in \text{Int}(\mathfrak{C})$.

Proposition 2 (*Forward Invariance of $\mathfrak{C}_\Sigma$*): *Provided that $x(0) \in \mathfrak{C}_\Sigma$, if there exists a $u : [0, \infty) \to \mathfrak{U}$ satisfying the ensemble analog of (16) (given in Appendix B), then $x(t) \in \mathfrak{C}_\Sigma \ \forall \ t \geq 0$, with probability 1.*

Proof. This proof follows from an invocation of Theorem 1, from $\mathfrak{C}_\Sigma$'s definition, and from the smoothness of $\hat{h}_\Sigma$. We provide the details in Appendix B. ∎

Remark 2: *Although the proof for Proposition 2 establishes the invariance of $\mathfrak{C}$ for all $t \geq 0$, we note here that we require safety to be assured only for a finite interval, since the controls*

---

[3]Note that $\alpha_\star$ ($\mathcal{K}$-class function) and α (extended $\mathcal{K}$-class function), while similar, denote separate objects throughout this article.

are computed (and applied) for $t \in \{0, 1, \ldots, \mathrm{T}_{\max} - 1\}$ (see the limits of the summation operator in Eqs. (3) and (13)).

Algorithm 1 Safety-Aware Collective Control via Polynomial-Smoothed NCBFs and Stochastic Quadratic Programming

---

Require: N, T_u, x_g, ε_g, Dynamics: f, g, K_w, w_i.

1: Max. mission time (in number of time steps): $\mathrm{T}_{\max}$.
2: $\mathrm{T}_g \leftarrow 0$.
3: **repeat**
4:      $\mathrm{x}_0 = \text{STATEESTIMATOR}()$.
5:      $\mathrm{x}_c \leftarrow \mathrm{x}_0$ ▷ *Store current state.*
6:      Sample ensemble disturbance realization, w.
7:      **for** $\tau \leftarrow 1$ to T_u **do**
8:          Compute $\mathrm{u}_d(\tau) = (\mathrm{x}_c - \mathrm{x}_g) + (K_w \otimes I_N)\mathrm{w}$.
9:          $\mathrm{x}_c \leftarrow F(\mathrm{x}_c) + G(\mathrm{x}_c)\mathrm{u}_d(\tau)$.
10:     Solve stochastic QP (13) for $\{\mathrm{u}_\star(t)\}_{t=0}^{\mathrm{T}_u - 1}$.
11:     Send $\mathrm{u}_\star(0)$ to agents' actuators.
12:     **for** $\tau \leftarrow 1$ to $\mathrm{T}_u - 2$ **do**
13:         $\mathrm{u}(\tau - 1) \leftarrow \mathrm{u}_\star(\tau)$.
14:     $\mathrm{u}(\mathrm{T}_u - 1) \leftarrow \mathrm{u}_{\text{init}}$ ▷ *Initial element of* u.
15:     **if** $\mathrm{T}_g = \mathrm{T}_{\max}$ **then**
16:         **break**   ▷ *Exit loop.*
17:     $\mathrm{T}_g \leftarrow \mathrm{T}_g + 1$.
18: **until** $||\mathrm{x}_c - \mathrm{x}_g|| \leq \varepsilon_g$.

VII. MULTI-AGENT SAFETY NOTIONS

While different notions for safety exist, each depending on the specific task requirements at hand, we define safety in this article using the two constraints provided next (see Fig. 1 for a visual summary of the tasks under consideration).

Safety Constraints:

(C1) an inter-agent collision avoidance constraint, which requires each agent to maintain a distance of δ meters (m) from other agents, and

(C2) an obstacle avoidance constraint that ensures agents do not collide with N_o static circular obstacles of radius r_o in the mission field by keeping a distance of $\delta_o > \delta$ m, with δ_o at least twice the width of each agent[4].

The aforementioned conditions can be enforced separately by the CBFs (i signifies the agent index)

$$h_i^\delta = ||x_i - x_j||^2 - \delta^2 \qquad (17)$$

$$h_i^o = ||x_i - x_k^o||^2 - \delta_o^2, \qquad (18)$$

where x_k^o is the absolute position of the kth obstacle. Using (7), we select the following CBF candidate for the ensemble $h_\Sigma = h_\delta \wedge h_o$, with h_δ and h_o given, respectively, as

$$\bigwedge_{i=1}^{N} \bigwedge_{j=1+1}^{N} h_i^\delta \text{ and } \bigwedge_{i=1}^{N} \bigwedge_{k=1}^{N_o} h_i^o. \qquad (19)$$

---

[4]We will assume that these distances are detected by range sensors fitted on each agent. Obstacles of disparate sizes can also be considered with an appropriate modification of δ_o.

The entire safe set $\mathfrak{C}_\Sigma$ is, thus, $\mathfrak{C}_\Sigma = \bigcap_{\star \in \{\delta, o\}} \mathfrak{C}_\star$, with corresponding CBF, $\min_{\star \in \{\delta, o\}} h_\star$. Finally, we will assume that u_f in (A3) evolves according to the following feedback control law: $\mathrm{u}_f(t) = \mathrm{x}(t) - \mathrm{x}_g$. This nominal controller is thus adjusted by the safety filter provided by the CBF only as needed, to adhere to the specified safety requirements.

VIII. NUMERICAL EXAMPLES

We discuss two mission scenarios in this section. In the first mission, the agents must each navigate to their respective goal locations within a planar mission space that includes a single obstacle whose position is known; in the second, multiple obstacles are introduced. For both examples, we consider single-integrator systems with dynamics of the form

$$\dot{p}_{x_i} = (v_{x_i} + w_{x_i}) \qquad (20a)$$

$$\dot{p}_{y_i} = (v_{y_i} + w_{y_i}), i = 1, 2, \ldots, N, \qquad (20b)$$

with control inputs perturbed by w_{x_i} and w_{y_i}, the x and y components of the normally-distributed disturbance, w_i. p_{x_i} and p_{y_i} are the ith agent's position in the plane along the abscissa and ordinate axes, respectively, with corresponding linear velocities, v_{x_i} and v_{y_i}. It is trivial to see that (20) is equivalent to the control-affine form in (1), with $f = 0_2$ and $g = \mathbb{I}_2$, and where $u_i = [v_{x_i}, v_{y_i}]^\top$ and $x_i = [p_{x_i}, p_{y_i}]^\top$ are the ith agent's control and state vectors, respectively. For convenience, we will take the disturbance terms to encode the nonlinearity in the agents' models. Furthermore, the agents are tasked with the multi-objective problem set forth in Section VIII. Table II lists values of the parameters we used during simulations. Assuming identical agents, we can then write the ensemble-level dynamics for the single-integrator collective as $\dot{\mathrm{x}} = \mathrm{u} + (K_w \otimes I_N)\mathrm{w}$, with $\mathrm{x} = [p_{x_1}, p_{y_1}, p_{x_2}, p_{y_2}, \ldots, p_{x_N}, p_{y_N}]^\top$ and $\mathrm{u} = [v_{x_1}, v_{y_1}, v_{x_2}, v_{y_2}, \ldots, v_{x_N}, v_{y_N}]^\top$. In Algorithm 1,

TABLE II: SIMULATION PARAMETERS

Parameter	Value	Parameter	Value
β	$[1, \frac{1}{2}, \frac{1}{10}]$	a_3	$-\frac{5}{4\beta^3}$
$[\delta, \delta_0]$	$[0.14, 0.15]$	a_5	$\frac{3}{8\beta^5}$
δ_h	0.01	K_w	$\mathbb{I}_2$
a_1	$\frac{15}{8\beta}$	$\alpha(\gamma, h)$	$\gamma^3 h$
$a_{2k, k \in \mathbb{Z}_{\geq 0}}$	0	Σ_w	$0.1 \cdot \mathbb{I}_d$

we provide pseudocode for implementing the proposed safety-aware control scheme. T_g is the number of time steps it takes for all agents to reach positions within their respective goal sets. Its value is unknown; however, for finite goal reachability, we require that T_g be bounded by a finite number, $\mathrm{T}_{\max} \in \mathbb{Z}_+$. The STATEESTIMATOR sub-routine returns an estimate of the current ensemble state.

IX. RESULTS & DISCUSSIONS

A. Safety-Critical Control under Varied Safety Specifications

In Fig. 2a, we plot the collective's trajectories and associated CBFs, with both smoothing schemes and for the single-obstacle mission. Similar plots corresponding to the multiple-
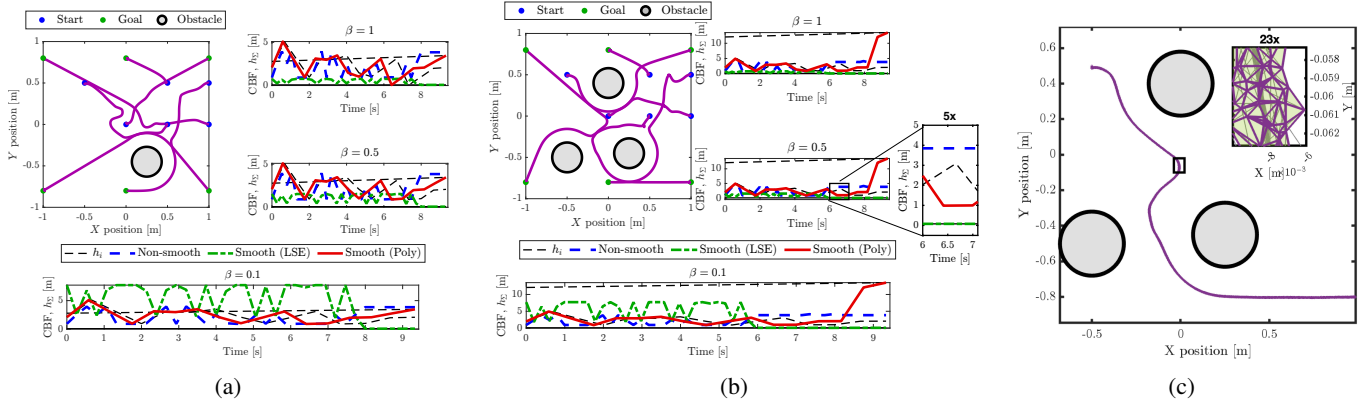
Fig. 2: **Comparing controller reach-avoid performance**: (a) Time-evolution of agent trajectories and CBFs showing the effect of smoothing technique and smoothing parameter (β) on h_Σ, with a horizontal line at $h_\Sigma = 0$ for reference. (b) Results for the multiple-obstacle example. From the inset on the right, we notice that the LSE-smoothed CBF assumes a zero value rather abruptly on the interval $6 \leq t \leq 7$. (c) Envelope of 100 safe trajectories (with magnified inset) for agent 3.

obstacle example are provided in Fig. 2b, from where we notice comparable observations as in the single-obstacle case, with the controller driving each agent locally to the goal while ensuring that they avoid obstacles and inter-agent collisions en route. From the magnified view of the highlighted area in Fig. 2b, we also notice that $\hat{h}_\Sigma$ admits a zero value for the LSE approximation scheme, which may cause the controller to revert to its nominal value, hence violating the safety constraints and effectively invalidating such CBF. $\hat{h}_\Sigma$ is also less sensitive to the smoothing parameter for the polynomial approximation scheme than for the LSE technique. Finally, we notice from Fig. 2c, that under the proposed controller, assured safety is guaranteed for all 100 experiment runs.

B. Effect of β on Safety-Critical Control

Next, in Fig. 3, we compare the nominal and CBF-filtered control inputs for $\beta \in \{0.1, 0.5, 1\}$, (for the multiple-obstacle setting only, due to page limits). From the plot, we observe that the polynomial smoothing technique tends to generate more conservative control actions that are adequate for the task and closely track the nominal CBF controller, modifying it only minimally, while that of the log-sum-exp function is more aggressive, exhibiting larger deviations from the nominal CBF-based controller, for unsafe control actions.

C. Control Input Computation Time

Lastly, given that Algorithm 1 relies on polynomial approximation and solving an online optimization problem, we computed average control input computation times for 100 simulation runs (see Table III) to roughly assess our method's performance. Although limited, the results show a direct relationship between the control horizon and average control synthesis time. Simulations were conducted using a workstation equipped with an AMD 7950x 16-core processor and 128 GB of RAM, with software programs optimized for parallel processing.
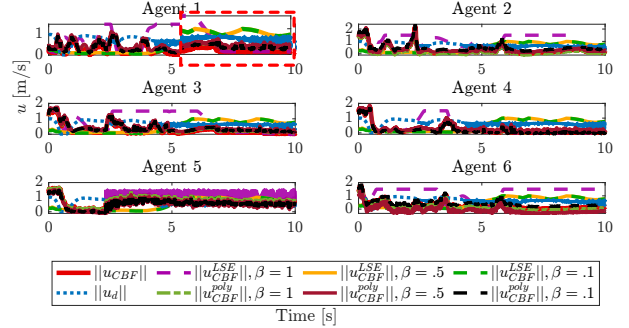

Fig. 3: **Synthesized control inputs**: Time-evolution of the norms of the nominal and CBF-filtered control inputs with $\beta = 1, 0.5$, and 0.1, and for the multiple-obstacle mission setting. A red inset highlighting the control inputs corresponding to the nominal and CBF-filtered cases is also depicted.

TABLE III: COMPUTATION TIME (IN MILLISECONDS)

T_u [time steps]	Computation Time [ms]	
	I	II
10	0.6 ± 0.03	1.05 ± 0.2
20	0.647 ± 0.035	1.08 ± 0.23
30	0.66 ± 0.037	1.11 ± 0.239

X. CONCLUSIONS

This paper set out to illuminate an approach for safely coordinating autonomous collectives operating in environments with obstacles and under noisy control inputs. We showed that by polynomial smoothing of Boolean-composed CBFs and stochastic MPC, safe controls can be synthesized (locally) for collectives by considering the ensemble as one agent, computing controls, and propagating the resulting optimal values (component-wise) to individual agents, leading to both safety and performance guarantees. In addition, we provided evidence for the robustness of our proposed method

to variations in approximation for small-enough values of the approximation parameter, with a fractional error upper bound. Our work, however, relies on the assumption of identical agent dynamics and a static environment, making controller synthesis for uncertain heterogeneous dynamical systems under varied tasking an interesting topic for future work. Other valid directions include an extension of the foregoing ideas to the non-Gaussian noise scenario or for more complex and dynamic mission environments. One can also consider gradient-free metaheuristic techniques that depart altogether from the QP-based framework presented here.

REFERENCES

[1] P. Glotfelter, J. Cortes, and M. Egerstedt, "Nonsmooth Barrier Functions With Applications to Multi-Robot Systems," *IEEE Control Systems Letters*, vol. 1, pp. 310–315, Oct. 2017.

[2] T. G. Molnar and A. D. Ames, "Composing Control Barrier Functions for Complex Safety Specifications," Sept. 2023. arXiv:2309.06647 [cs, eess, math].

[3] O. So, A. Clark, and C. Fan, "Almost-sure safety guarantees of stochastic zero-control barrier functions do not hold," *arXiv preprint arXiv:2312.02430*, 2023.

[4] 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.

[5] 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.

[6] 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 58th Conference on Decision and Control (CDC)*, pp. 4797–4803, Dec. 2019. ISSN: 2576-2370.

[7] A. Sahiner, N. Yilmaz, and S. A. Ibrahem, "Smoothing approximations to non-smooth functions," *Journal of Multidisciplinary Modeling and Optimization*, vol. 1, no. 2, pp. 69–74, 2018.

[8] 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.

APPENDICES

A. Proof of Proposition 1

Proof. Since $\hat{e} = 0$ everywhere on $\mathbb{R}$ where $\phi(\ell)$ and $\hat{\phi}(\ell, \beta)$ coincide (see Lemma 1), and $\ell \leq \beta$ on $\mathcal{I}_\beta = [-\beta, \beta]$, using the definition of the $L^1[a, b]$ norm and from Eqs. (8) and (11), we can write

$$\mathbb{E}\left[||\hat{e}||_{L^1[a,b]}\right] = \mathbb{E}\left[\int_{-\beta}^{\beta} |\hat{e}| d\ell\right]$$

$$= \mathbb{E}\left[\int_{-\beta}^{\beta} \left| -\frac{1}{2}(\ell\hat{\phi} - \ell') + \frac{1}{2}(\ell\phi - \ell')\right| d\ell\right]$$

$$= \mathbb{E}\left[\int_{-\beta}^{\beta} \left|\frac{1}{2}\ell(\phi - \hat{\phi})\right| d\ell\right]$$

$$\leq \frac{\beta}{2}\mathbb{E}\left[\int_{-\beta}^{\beta} |\phi - \hat{\phi}| d\ell\right] \text{ (by the monotonicity of } \mathbb{E}[\cdot])$$

$$= \frac{\beta}{2}\mathbb{E}\left[||\phi - \hat{\phi}||_{L^1[a,b]}\right] = \frac{\beta}{2} \cdot \frac{15\beta}{4} = \frac{15\beta^2}{8},$$

by Lemma 1. Thus, we arrive at the intended result given in Proposition 1 (i.e., $\mathbb{E}\left[||\hat{e}||_{L^1[a,b]}\right] \leq \frac{15\beta^2}{8}$), which completes the proof. ∎

B. Proof of Proposition 2

Proof. Let μ_h and σ_h denote the ensemble analogs of μ_i and σ_i, defined respectively as

$$\mu_h := L_F\hat{h}_\Sigma(\mathbf{x}) + L_G\hat{h}_\Sigma(\mathbf{x})\mathbf{u}$$
$$+ \frac{1}{2}\text{tr}\left(\left((K_w \otimes I_N)^\top \frac{\partial^2 \hat{h}_\Sigma(\mathbf{x})}{\partial \mathbf{x}^2}(K_w \otimes I_N)\right)\right) \quad (21a)$$

$$\sigma_h := \frac{\partial \hat{h}_\Sigma(\mathbf{x})}{\partial \mathbf{x}}(K_w(\mathbf{x}) \otimes I_N). \quad (21b)$$

According to Theorem 1, we need to show that $\hat{h}_\Sigma$ satisfies

$$\mu_h - \frac{\sigma_h^2}{\hat{h}_\Sigma(\mathbf{x})} \geq -\hat{h}_\Sigma^2(\mathbf{x})\alpha_\star(\hat{h}_\Sigma). \quad (22)$$

Our proof will proceed in two steps. First, we will show that $\hat{h}_\Sigma$ is a valid zeroing CBF (in Part I of the proof). To finish the proof, in Part II, we will then construct a *reciprocal* CBF (RCBF) from $\hat{h}_\Sigma$ and prove that such an RCBF satisfies valid conditions for almost-sure safety as set forth in Theorem 10 in [3]. Part I's proof follows.

From Eqs. (10) and (11), by the definitions of ℓ and ℓ', and since $\inf_\ell M_2(\ell, \beta) = -1$ and $\sup_\ell M_2(\ell, \beta) = 1$ on $-\beta \leq \ell \leq \beta$, it can be shown that $\hat{h}_\Sigma > 0$ for all $h_*, * \in \mathcal{I}$ such that $\forall \mathbf{x} \in \partial\mathfrak{C}_\Sigma, h_*(\mathbf{x})$ is positive[5]. Note that, by the sum and difference law of limits and the continuous differentiability of its constituent CBFs, $\hat{h}_\Sigma$ is smooth and twice continuously-differentiable, i.e., in C^2. Thus, by Lemma 2 in [5], if $\hat{h}_\Sigma(\mathbf{x}) > 0 \ \forall \ \mathbf{x} \in \partial\mathfrak{C}_\Sigma$, then for each $k \in \mathbb{Z}_+$, there exists a constant $\gamma > 0$ s.t.

$$\frac{d\hat{h}_\Sigma(\mathbf{x})}{d\mathbf{x}} \geq -\gamma\hat{h}_\Sigma^k(\mathbf{x}), \ \forall \ \mathbf{x} \in \text{Int}(\mathfrak{C}). \quad (23)$$

Setting $\alpha(\gamma, \hat{h}_\Sigma) = \gamma\hat{h}_\Sigma^k(\mathbf{x})$, we conclude that $\hat{h}_\Sigma$ satisfies the ensemble analog of (2), thus completing Part I's proof.

For Part II, motivated by Corollary 11 in [3], we define the RCBF, $\hat{B}_\Sigma := 1/\hat{h}_\Sigma$, and write its corresponding drift term, μ_B, in terms of μ_h and σ_h as

$$\mu_B = -\hat{h}_\Sigma^{-2}\mu_h + \hat{h}_\Sigma^{-3}\sigma_h^2. \quad (24)$$

By the definition of an RCBF (see [5], [3]), from (24), we can write

$$-\hat{h}_\Sigma(\mathbf{x})^{-1}\mu_B \leq \alpha_\star(\hat{h}_\Sigma)$$

$$\implies -\hat{h}_\Sigma(\mathbf{x})^{-1}\left(-\hat{h}_\Sigma^{-2}\mu_h + \hat{h}_\Sigma^{-3}\sigma_h^2\right) \leq \alpha_\star(\hat{h}_\Sigma)$$

$$\implies \mu_h - \frac{\sigma_h^2}{\hat{h}_\Sigma(\mathbf{x})} \geq -\hat{h}_\Sigma^2(\mathbf{x})\alpha_\star(\hat{h}_\Sigma),$$

which is the same expression in (22), where $\alpha_\star$ is a $\mathcal{K}$-class function. We have now proved almost-sure safety for $\hat{h}_\Sigma$. Invoking Proposition 1 in [5], since the previous argument establishes the validity of $\hat{h}_\Sigma$ as a zeroing CBF for all time $t \geq 0$, the forward invariance of $\mathfrak{C}_\Sigma$ follows immediately, thus completing the proof. ∎

---

[5]This statement is valid for all $h_*(\mathbf{x})$ cases, i.e., for $\sum_{\star\in\mathcal{I}\backslash\{1\}} h_\star(\mathbf{x}) - h_1(\mathbf{x}) \geq 0$ and for $\sum_{\star\in\mathcal{I}\backslash\{1\}} h_\star(\mathbf{x}) - h_1(\mathbf{x}) \leq 0$. In all cases, $\ell' > 0$, since $h_* > 0$, for all $* \in \mathcal{I}$ by Definition 2.