Safe Control Synthesis With Uncertain Dynamics and Constraints

Kehan Long, *Student Member, IEEE*, Vikas Dhiman, *Member, IEEE*, Melvin Leok, Jorge Cortés, *Fellow, IEEE*, and Nikolay Atanasov, *Member, IEEE*

***Abstract*—This paper considers safe control synthesis for dynamical systems with either probabilistic or worst-case uncertainty in both the dynamics model and the safety constraints. We formulate novel probabilistic and robust (worst-case) control Lyapunov function (CLF) and control barrier function (CBF) constraints that take into account the effect of uncertainty in either case. We show that either the probabilistic or the robust (worst-case) formulation leads to a second-order cone program (SOCP), which enables efficient safe and stable control synthesis. We evaluate our approach in PyBullet simulations of an autonomous robot navigating in unknown environments and compare the performance with a baseline CLF-CBF quadratic programming approach.**

***Index Terms*—Optimal control and optimization, robot safety under uncertainty, robust control.**

I. Introduction

AUTONOMOUS robotic systems are increasingly employed in warehouse and home automation, transportation, and security applications. A crucial aspect of successfully deploying such systems is the satisfaction of safety and stability requirements, even in the presence of uncertainty in the system model or constraints. The notion of safety in the context of program correctness was first introduced in the 1970's [1]. Around the same time, Artstein [2] introduced control Lyapunov functions (CLFs) to enforce stability in the context of nonlinear system control. The seminal work of Sontag [3] established a universal formula for constructing feedback control laws that stabilize nonlinear systems. In the 2000's, barrier certificates were proposed to formally prove the safety of closed-loop nonlinear and hybrid systems [4], [5]. Control barrier functions (CBFs) were developed to support task-independent safe control synthesis, serving as a barrier certificate for a closed-loop nonlinear system [6].

A key observation is that, for control-affine systems, the CLF and CBF conditions are linear in the control input, allowing a formulation of safe and stable control synthesis as a quadratic program (QP) [7]–[9]. CLF-CBF-QP techniques have been successfully employed in a variety systems, including aerial robots [10], walking robots [11], and automotive systems [12]. Most existing work, however, assumes complete knowledge of the system dynamics and control barrier functions. In reality, the dynamics model and safety constraints are obtained using noisy sensor data and simplifying assumptions, leading to uncertainty and errors that should be captured when ensuring safety and stability.

Capturing system-model and barrier-function estimation errors impacts the formulation of CLF and CBF constraints, and no longer give rise to QPs. Our main contribution is to show that such uncertainty-aware stability and safety constraints can still be formulated as convex constraints under two different models of uncertainty: probabilistic and worst-case. To capture probabilistic uncertainty, we specifically consider *Gaussian Process* (GP) regression as an example approach for modeling a probability distribution over a function space. When the estimated barrier function and system dynamics are described by a GP, we aim to ensure probabilistic safety and stability up to a user-specified risk tolerance. We compute the distribution of the CLF and CBF constraints, and use Cantelli's inequality [13] to bound the computed means with a margin dependent on the variances and the desired risk-tolerance. The control input appears linearly in the mean and quadratically in the variance of the CLF and CBF constraints. This allows us to restate the probabilistic constraints as second-order cone constraints, leading to a second-order cone program (SOCP), which is convex and can be solved efficiently online.

When *worst-case error bounds* on the system dynamics, barrier function and its gradient are given, we formulate a robust safe control synthesis problem. Under worst-case disturbances, we show that the input appears both linearly and within a norm term in the CLF and CBF constraints. Like the probabilistic formulation, the original QP problem can be reformulated as a convex SOCP for safe control synthesis.

We demonstrate our safe control synthesis techniques in mobile robot navigation simulations. We consider a robot tasked to follow a desired path in an unknown environment, relying on online noisy obstacle sensing and offline dynamic model estimation to ensure safety and stability. We show that both the probabilistic and the robust CLF-CBF-SOCP formulation allows the robot to safely track the deisred path.

In summary, we make the following contributions. First, we formulate novel probabilistic safety and stability constraints by considering stochastic uncertainty in the barrier functions and system dynamics. Second, we formulate novel robust safety and stability constraints by considering worst-case error bounds in the barrier functions and system dynamics. Finally, we show that either the probabilistic or the worst-case formulations lead to a

Manuscript received February 19, 2022; accepted June 2, 2022. Date of publication June 20, 2022; date of current version June 21, 2022. This letter was recommended for publication by Associate Editor X. Li and Editor C. Gosselin upon evaluation of the reviewers' comments. This work was supported by NSF RI under Grant IIS-2007141. *(Corresponding author: Kehan Long.)*

Kehan Long, Melvin Leok, Jorge Cortés, and Nikolay Atanasov are with the Contextual Robotics Institute, University of California, San Diego, CA 92093 USA (e-mail: k3long@ucsd.edu; mleok@ucsd.edu; cortes@ucsd.edu; natanasov@ucsd.edu).

Vikas Dhiman is with the Department of Electrical and Computer Engineering, University of Maine, Bangor, ME 04469 USA (e-mail: vikas.dhiman@maine.edu).

Digital Object Identifier 10.1109/LRA.2022.3182544

(convex) SOCP, enabling efficient synthesis of safe and stable control.

II. Related Work

This section reviews recent works on safe control synthesis that address uncertainty due to unmodeled dynamics, input disturbances, and barrier function estimation.

Jankovic [14] considers worst-case disturbance bounds on the system dynamics and proposes robust CBF formulations. Eman *et al.* [15] utilize convex hulls to model disturbances in a CBF-based safety framework. Clark [16] considers stochastic control systems with incomplete information and derives sufficient conditions for ensuring safety on average. Nguyen and Sreenath [17] formulate a robust CLF-CBF QP by introducing robust constraints to guarantee stability and safety under model uncertainty. Hewing *et al.* [18] present a model predictive control (MPC) approach that integrates a nominal system with a residual part modeled as a GP. Compared to our formulation, this approach enables optimizing the control performance over a longer future horizon but requires time discretization and convexification of the safety constraints. In contrast, our formulations operate in continuous time and handle general safe set descriptions. Ahmadi *et al.* [19] introduce a conditional value-at-risk (CVaR) barrier function to ensure safety for systems with stochastic uncertainty. The approach guarantees safety with high probability even for worst-case scenarios but the computation cost is high and the formulation is restricted to linear systems. Our approach enables efficient control synthesis for general control-affine systems. Another line of research formulates safe control synthesis as trajectory optimization. Alcan and Kyrki [20] employ differential dynamics programming (DDP) to enforce safety under additive uncertainty. In [21], the DDP idea is combined with CBF to introduce a barrier state formulation for safety of discrete-time systems.

Input-to-state safety (ISSf) was introduced in [22] to handle input disturbances and was used in [23] to enlarge a safe set by modifying a CBF. Alan *et al.* [24] introduce a tunable ISSf-CBF for safe control synthesis while reducing conservatism. Cosner *et al.* in [25] introduce measurement-robust CBFs to account for uncertainty in state estimation and conduct experiments on a Segway.

Srinivasan *et al.* [26] estimate barrier functions online using a Support Vector Machine and solve a CLF-CBF QP to generate safe control inputs. Zhang *et al.* [27] construct robust output CBFs from safe expert demonstrations while considering worst-case error bounds in the measurement map and system dynamics.

This paper unifies and extends our prior work [28], [29] by considering safe control synthesis with uncertainty in the system dynamics and the barrier function simultaneously and studying two separate cases of probabilistic and worst-case uncertainty. In contrast, [28] only considered probabilistic uncertainty in the dynamics using Gaussian process regression, while [29] only considered worst-case error bounds in the barrier function. We show that in either case the safe control synthesis problem is a convex SOCP, which enables efficient safe and stable control synthesis online.

III. Problem Formulation

Consider a robot with dynamics model:

$$\dot{\mathbf{x}} = f(\mathbf{x}) + g(\mathbf{x})\mathbf{u} = [f(\mathbf{x})\ g(\mathbf{x})] \cdot \begin{bmatrix} 1 \\ \mathbf{u} \end{bmatrix} \triangleq F(\mathbf{x})\underline{\mathbf{u}}, \tag{1}$$

where $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^n$ is the robot state and $\underline{\mathbf{u}} \in \underline{\mathcal{U}} = \{1\} \times \mathbb{R}^m$ is the control input.[1] We assume $f : \mathbb{R}^n \mapsto \mathbb{R}^n$ and $g : \mathbb{R}^n \mapsto \mathbb{R}^{n\times m}$ are continuously differentiable.

Definition III.1: A continuously differentiable function $V : \mathbb{R}^n \mapsto \mathbb{R}_{\geq 0}$ is a *control Lyapunov function (CLF)* for the system (1) if there exists a class $\mathcal{K}$ function α_V such that:

$$\inf_{\underline{\mathbf{u}}\in\underline{\mathcal{U}}} CLC(\mathbf{x}, \underline{\mathbf{u}}) \leq 0, \quad \forall \mathbf{x} \in \mathcal{X}, \tag{2}$$

where the *control Lyapunov condition (CLC)* is:

$$\begin{aligned} CLC(\mathbf{x}, \underline{\mathbf{u}}) &\triangleq \mathcal{L}_f V(\mathbf{x}) + \mathcal{L}_g V(\mathbf{x})\mathbf{u} + \alpha_V(V(\mathbf{x})) \\ &= [\nabla_{\mathbf{x}} V(\mathbf{x})]^\top F(\mathbf{x})\underline{\mathbf{u}} + \alpha_V(V(\mathbf{x})). \end{aligned} \tag{3}$$

A CLF V may be used to encode a variety of control objectives, including path following [29], adaptive cruise control [12], and bipedal robot walking [11].

To define safety requirements for the control objective, consider a continuously differentiable function $h : \mathbb{R}^n \mapsto \mathbb{R}$, which implicitly defines a (closed) safe set of system states $\mathcal{S} \triangleq \{\mathbf{x} \in \mathcal{X} \mid h(\mathbf{x}) \geq 0\}$. The following definition is a useful tool to ensure that $\mathcal{S}$ is forward invariant, i.e., the robot state remains in $\mathcal{S}$ throughout its evolution.

Definition III.2: A continuously differentiable function $h : \mathbb{R}^n \mapsto \mathbb{R}$ is a *control barrier function (CBF)* on $\mathcal{X} \subseteq \mathbb{R}^n$ for (1) if there exists an extended class $\mathcal{K}_\infty$ function α_h with:

$$\sup_{\underline{\mathbf{u}}\in\underline{\mathcal{U}}} CBC(\mathbf{x}, \underline{\mathbf{u}}) \geq 0, \quad \forall \mathbf{x} \in \mathcal{X}, \tag{4}$$

where the *control barrier condition (CBC)* is:

$$\begin{aligned} CBC(\mathbf{x}, \underline{\mathbf{u}}) &\triangleq \mathcal{L}_f h(\mathbf{x}) + \mathcal{L}_g h(\mathbf{x})\mathbf{u} + \alpha_h(h(\mathbf{x})) \\ &= [\nabla_{\mathbf{x}} h(\mathbf{x})]^\top F(\mathbf{x})\underline{\mathbf{u}} + \alpha_h(h(\mathbf{x})). \end{aligned} \tag{5}$$

According to [7], [9], any Lipschitz-continuous controller $\underline{\mathbf{k}} : \mathcal{X} \mapsto \underline{\mathcal{U}}$ that satisfies $CBC(\mathbf{x}, \underline{\mathbf{k}}(\mathbf{x})) \geq 0$ for all $\mathbf{x} \in \mathcal{X}$ renders the set $\mathcal{S}$ forward invariant for the system (1).

A. Safety and Stability With Known System Dynamics and Barrier Function

When the system dynamics $F(\mathbf{x})$ and barrier function $h(\mathbf{x})$ are known, a safe controller can be synthesized by combining CLF and CBF constraints in a quadratic program:

$$\begin{aligned} &\min_{\underline{\mathbf{u}}\in\underline{\mathcal{U}}, \delta\in\mathbb{R}} \ \|L(\mathbf{x})^\top(\underline{\mathbf{u}} - \tilde{\underline{\mathbf{k}}}(\mathbf{x}))\|^2 + \lambda\delta^2, \\ &\text{s.t. } CLC(\mathbf{x}, \underline{\mathbf{u}}) \leq \delta,\ CBC(\mathbf{x}, \underline{\mathbf{u}}) \geq 0. \end{aligned} \tag{6}$$

The term $\tilde{\underline{\mathbf{k}}}(\mathbf{x})$ is a baseline controller and may be used to specify additional control requirements, such as desirable velocity or orientation. This term may be set to $\tilde{\underline{\mathbf{k}}}(\mathbf{x}) \equiv \boldsymbol{e}_1$ if minimum control effort is the main objective. The term $L(\mathbf{x})$ is a weighting matrix penalizing deviation from the baseline controller. The term $\delta \geq 0$ is a slack variable that relaxes the CLF constraints to ensure the feasibility of the QP, controlled

[1] *Notation:* We denote by $\mathbf{I}_n \in \mathbb{R}^{n\times n}$ the identity matrix and $\partial\mathcal{A}$ the boundary of a set $\mathcal{A} \subset \mathbb{R}^n$. For a vector $\mathbf{x}$ and a matrix $\mathbf{X}$, we use $\|\mathbf{x}\|$ and $\|\mathbf{X}\|$ to denote the Euclidean norm and the spectral norm. We use $\text{vec}(\mathbf{X}) \in \mathbb{R}^{nm}$ to denote the vectorization of $\mathbf{X} \in \mathbb{R}^{n\times m}$, obtained by stacking its columns. We denote by ∇ the gradient and $\mathcal{L}_f V = \nabla V \cdot f$ the Lie derivative of a differentiable function V along a vector field f. We use $\otimes$ to denote the Kronecker product and $\mathcal{GP}(\mu(\mathbf{x}), K(\mathbf{x}, \mathbf{x}'))$ to denote a Gaussian Process distribution with mean function $\mu(\mathbf{x})$ and covariance function $K(\mathbf{x}, \mathbf{x}')$. A continuous function $\alpha : [0, a) \to [0, \infty)$ is of class $\mathcal{K}$ if it is strictly increasing and $\alpha(0) = 0$, and it is of class $\mathcal{K}$ and $\lim_{r\to\infty} \alpha(r) = \infty$.

by the scaling factor $\lambda > 0$. The QP formulation in (6) modifies the baseline controller $\tilde{\underline{\mathbf{k}}}(\mathbf{x})$ online to ensure safety and stability via the CBF and CLF constraints.

We focus on enforcing safety and stability for the control-affine system in (1) when the system dynamics $F(\mathbf{x})$ and the barrier function $h(\mathbf{x})$ are *unknown* and need to be estimated from data. We present an approach for estimating the system dynamics and barrier functions from data in Section VI-A and Section VI-B, respectively. Our main goal is to develop techniques for safe and stable control synthesis with the estimated $F(\mathbf{x})$ and $h(\mathbf{x})$. We consider two scenarios, depending on whether probabilistic or worst-case error descriptions of the dynamics and barrier functions are available.

B. Safety and Stability With Gaussian Process Distributed System Dynamics and Barrier Function

When the system dynamics and barrier functions can be described as GPs, we consider the following probabilistic control synthesis problem.

Problem 1 (Safety and stability under Gaussian uncertainty): Gion on the unknown system dynamics $\text{vec}(F(\mathbf{x})) \sim \mathcal{GP}(\text{vec}(\tilde{F}(\mathbf{x})), K_F(\mathbf{x}, \mathbf{x}'))$ and an estimated distribution on the barrier function $h(\mathbf{x}) \sim \mathcal{GP}(\tilde{h}(\mathbf{x}), K_h(\mathbf{x}, \mathbf{x}'))$, design a feedback controller $\underline{\mathbf{k}}$ such that, for each $\mathbf{x} \in \mathcal{X}$:

$$\mathbb{P}(CLC(\mathbf{x}, \underline{\mathbf{k}}(\mathbf{x})) \leq \delta) \geq p, \quad \mathbb{P}(CBC(\mathbf{x}, \underline{\mathbf{k}}(\mathbf{x})) \geq 0) \geq p,$$

where $p \in (0, 1)$ is a user-specified risk tolerance.

C. Safety and Stability With Worst-Case Uncertainty in System Dynamics and Barrier Function

Many robotic systems require instead the guarantee that safety and stability hold under all possible error realizations, which motivates us to also consider the following problem.

Problem 2 (Safety and stability under worst-case uncertainty): Given estimated system dynamics $\tilde{F}(\mathbf{x})$ with known error bound $e_F(\mathbf{x})$,

$$\|F(\mathbf{x}) - \tilde{F}(\mathbf{x})\| \leq e_F(\mathbf{x}), \ \forall \mathbf{x} \in \mathcal{X}, \tag{7}$$

and estimated barrier function $\tilde{h}(\mathbf{x})$ and gradient $\nabla\tilde{h}(\mathbf{x})$ with known error bounds $e_h(\mathbf{x})$ and $e_{\nabla h}(\mathbf{x})$, i.e., for all $\mathbf{x} \in \mathcal{X}$,

$$|h(\mathbf{x}) - \tilde{h}(\mathbf{x})| \leq e_h(\mathbf{x}), \ \|\nabla h(\mathbf{x}) - \nabla\tilde{h}(\mathbf{x})\| \leq e_{\nabla h}(\mathbf{x}), \tag{8}$$

design a feedback controller $\underline{\mathbf{k}}$ such that, for each $\mathbf{x} \in \mathcal{X}$:

$$CLC(\mathbf{x}, \underline{\mathbf{k}}(\mathbf{x})) \leq \delta, \quad CBC(\mathbf{x}, \underline{\mathbf{k}}(\mathbf{x})) \geq 0.$$

IV. Probabilistic Safe Control

This section presents our solution to Problem 1. Inspired by the design (6) when the dynamics and the barrier function are known, we formulate the control synthesis problem via the following optimization problem:

$$\min_{\underline{\mathbf{u}} \in \underline{\mathcal{U}}, \delta \in \mathbb{R}} \ \|L(\mathbf{x})^\top(\underline{\mathbf{u}} - \tilde{\underline{\mathbf{k}}}(\mathbf{x}))\|^2 + \lambda\delta^2,$$

$$\text{s.t. } \mathbb{P}(CLC(\mathbf{x}, \underline{\mathbf{u}}) \leq \delta) \geq p, \quad \mathbb{P}(CBC(\mathbf{x}, \underline{\mathbf{u}}) \geq 0) \geq p. \tag{9}$$

The uncertainty in F and h affects the linearity in $\underline{\mathbf{u}}$ of the CLC and CBC conditions in the constraints of (9), making this optimization problem no longer a QP. Here, we justify that nevertheless the optimization can be solved efficiently. To show this, we start by analyzing the distributions of $CBC(\mathbf{x}, \underline{\mathbf{u}})$ and $CLC(\mathbf{x}, \underline{\mathbf{u}})$ in detail.

Proposition IV.1 (Mean and Variance for CBC): Assume h is a CBF with a linear function α_h, i.e., $\alpha_h(z) = a \cdot z$ for $a \in \mathbb{R}_{\geq 0}$. Given independent distributions $h(\mathbf{x}) \sim \mathcal{GP}(\tilde{h}(\mathbf{x}), K_h(\mathbf{x}, \mathbf{x}'))$ and $\text{vec}(F(\mathbf{x})) \sim \mathcal{GP}(\text{vec}(\tilde{F}(\mathbf{x})), K_F(\mathbf{x}, \mathbf{x}'))$, the mean and variance of $CBC(\mathbf{x}, \underline{\mathbf{u}})$ satisfy

$$\mathbb{E}[CBC(\mathbf{x}, \underline{\mathbf{u}})] = \mathbb{E}[\mathbf{p}(\mathbf{x})]^\top \underline{\mathbf{u}} \tag{10a}$$

$$Var[CBC(\mathbf{x}, \underline{\mathbf{u}})] = \underline{\mathbf{u}}^\top Var[\mathbf{p}(\mathbf{x})]\underline{\mathbf{u}}, \tag{10b}$$

where $\mathbf{p}(\mathbf{x}) := F^\top(\mathbf{x})[\nabla_{\mathbf{x}} h(\mathbf{x})] + [ah(\mathbf{x}) \ \ \mathbf{0}_m^\top]^\top \in \mathbb{R}^{m+1}$ and $\mathbb{E}[\mathbf{p}(\mathbf{x})]$, $Var[\mathbf{p}(\mathbf{x})]$ are computed in (16).

Proof: The control barrier condition can be written as:

$$\begin{aligned} CBC(\mathbf{x}, \underline{\mathbf{u}}) &= [\nabla_{\mathbf{x}} h(\mathbf{x})]^\top f(\mathbf{x}) + [\nabla_{\mathbf{x}} h(\mathbf{x})]^\top g(\mathbf{x})\mathbf{u} + ah(\mathbf{x}) \\ &= \left[[\nabla_{\mathbf{x}} h(\mathbf{x})]^\top F(\mathbf{x}) + \begin{bmatrix} ah(\mathbf{x}) & \mathbf{0}_m^\top \end{bmatrix}\right] \underline{\mathbf{u}} = \mathbf{p}(\mathbf{x})^\top \underline{\mathbf{u}}. \end{aligned}$$

Note that $\nabla_{\mathbf{x}} h(\mathbf{x})$ is a GP because the gradient of a GP with differentiable mean function and twice-differentiable covariance function is also a GP, cf. [28, Lemma 6],

$$\nabla_{\mathbf{x}} h(\mathbf{x}) \sim \mathcal{GP}(\nabla_{\mathbf{x}}\tilde{h}(\mathbf{x}), \mathcal{H}_{\mathbf{x},\mathbf{x}'} K_h(\mathbf{x}, \mathbf{x}')),$$

where $\mathcal{H}_{\mathbf{x},\mathbf{x}'} K_h(\mathbf{x}, \mathbf{x}') = \left[\frac{\partial^2 K_h(\mathbf{x}, \mathbf{x}')}{\partial \mathbf{x}_i, \partial \mathbf{x}'_j}\right]_{i=1,j=1}^{n,n}$ is finite for all $(\mathbf{x}, \mathbf{x}') \in \mathbb{R}^{2n}$. Since $\text{vec}(\mathbf{ABC}) = (\mathbf{C}^\top \otimes \mathbf{A})\text{vec}(\mathbf{B})$ for appropriately sized matrices $\mathbf{A}, \mathbf{B}, \mathbf{C}$, we can write

$$\begin{aligned} Var(F(\mathbf{x})\underline{\mathbf{u}}) &= Var((\underline{\mathbf{u}}^\top \otimes \mathbf{I}_n)\text{vec}(F(\mathbf{x}))) \\ &= (\underline{\mathbf{u}}^\top \otimes \mathbf{I}_n) K_F(\mathbf{x}, \mathbf{x})(\underline{\mathbf{u}} \otimes \mathbf{I}_n). \end{aligned} \tag{11}$$

For brevity, we let $K_F := K_F(\mathbf{x}, \mathbf{x}')$ and $K_h := K_h(\mathbf{x}, \mathbf{x}')$ and $\mathbf{p}_1 = F^\top(\mathbf{x})[\nabla_{\mathbf{x}} h(\mathbf{x})]$. The term $[\nabla_{\mathbf{x}} h(\mathbf{x})]^\top F(\mathbf{x})\underline{\mathbf{u}}$ is an inner product of two independent GPs, $\nabla_{\mathbf{x}} h(\mathbf{x})$ and $F(\mathbf{x})\underline{\mathbf{u}}$. Thus, using [28, Lemma 5], (11), and that $Cov(\nabla_{\mathbf{x}} h(\mathbf{x}), F(\mathbf{x})\underline{\mathbf{u}}) = 0$, $\mathbf{p}_1^\top \underline{\mathbf{u}}$ corresponds to a distribution with mean and variance:

$$\begin{aligned} &\mathbb{E}[\mathbf{p}_1^\top \underline{\mathbf{u}}] = [\nabla_{\mathbf{x}}\tilde{h}(\mathbf{x})]^\top \tilde{F}(\mathbf{x})\underline{\mathbf{u}}, \\ &Var[\mathbf{p}_1^\top \underline{\mathbf{u}}] = [\nabla_{\mathbf{x}}\tilde{h}(\mathbf{x})]^\top (\underline{\mathbf{u}}^\top \otimes \mathbf{I}_n) K_F \\ &(\underline{\mathbf{u}} \otimes \mathbf{I}_n)\nabla_{\mathbf{x}}\tilde{h}(\mathbf{x}) + \underline{\mathbf{u}}^\top \tilde{F}^\top(\mathbf{x}) \mathcal{H}_{\mathbf{x},\mathbf{x}'} K_h \tilde{F}(\mathbf{x})\underline{\mathbf{u}}. \end{aligned} \tag{12}$$

To factorize $\underline{\mathbf{u}}$ from the variance expression, we apply the property $(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = \mathbf{AC} \otimes \mathbf{BD}$ two times,

$$\begin{aligned} &(\underline{\mathbf{u}} \otimes \mathbf{I}_n)[\nabla_{\mathbf{x}}\tilde{h}(\mathbf{x})] = (\underline{\mathbf{u}} \otimes \mathbf{I}_n)(1 \otimes [\nabla_{\mathbf{x}}\tilde{h}(\mathbf{x})]) \\ &= \underline{\mathbf{u}} \otimes \nabla_{\mathbf{x}}\tilde{h}(\mathbf{x}) = (\mathbf{I}_{m+1} \otimes \nabla_{\mathbf{x}}\tilde{h}(\mathbf{x}))\underline{\mathbf{u}}. \end{aligned} \tag{13}$$

By substituting (13) in (12), we can factorize out $\underline{\mathbf{u}}$ to get,

$$\begin{aligned} Var[\mathbf{p}_1] &= (\mathbf{I}_{m+1} \otimes [\nabla_{\mathbf{x}}\tilde{h}(\mathbf{x})]^\top) K_F (\mathbf{I}_{m+1} \otimes \nabla_{\mathbf{x}}\tilde{h}(\mathbf{x})) \\ &\quad + \tilde{F}^\top(\mathbf{x}) \mathcal{H}_{\mathbf{x},\mathbf{x}'} K_h \tilde{F}(\mathbf{x}). \end{aligned} \tag{14}$$

Next, we write $Cov(h(\mathbf{x}), \mathbf{p}_1^\top \underline{\mathbf{u}})$ using [28, Lemma 5] and $Cov(h(\mathbf{x}), F(\mathbf{x})\underline{\mathbf{u}}) = 0$,

$$\begin{aligned} Cov(h(\mathbf{x}), \mathbf{p}_1^\top \underline{\mathbf{u}}) &= Cov(h(\mathbf{x}), \nabla_{\mathbf{x}} h(\mathbf{x}))\tilde{F}(\mathbf{x})\underline{\mathbf{u}} \\ &= \left[[\nabla_{\mathbf{x}} K_h]^\top \tilde{f}(\mathbf{x}) \ \ [\nabla_{\mathbf{x}} K_h]^\top \tilde{g}(\mathbf{x})\right] \underline{\mathbf{u}}. \end{aligned} \tag{15}$$

Using (12), (14) and (15), we write the mean and variance,

$$\begin{aligned} &\mathbb{E}[\mathbf{p}(\mathbf{x})] = [\nabla_{\mathbf{x}}\tilde{h}(\mathbf{x})]^\top \tilde{F}(\mathbf{x}) + a[\tilde{h}(\mathbf{x}) \ \ \mathbf{0}_m^\top]^\top \\ &Var[\mathbf{p}(\mathbf{x})] = \tilde{F}^\top(\mathbf{x}) \mathcal{H}_{\mathbf{x},\mathbf{x}'} K_h \tilde{F}(\mathbf{x}) \\ &\quad + (\mathbf{I}_{m+1} \otimes \nabla_{\mathbf{x}}\tilde{h}(\mathbf{x})^\top) K_F (\mathbf{I}_{m+1} \otimes \nabla_{\mathbf{x}}\tilde{h}(\mathbf{x})) \end{aligned}$$

$$+\begin{bmatrix} a^2K_h + 2a[\nabla_{\mathbf{x}}K_h]^\top f(\mathbf{x}) & a[\nabla_{\mathbf{x}}K_h]^\top g(\mathbf{x}) \\ ag(\mathbf{x})^\top[\nabla_{\mathbf{x}}K_h] & \mathbf{0}_{m\times m} \end{bmatrix}, \quad (16)$$

from which the statement follows. ■

Next, we describe the distribution of $CLC(\mathbf{x}, \underline{\mathbf{u}})$.

Proposition IV.2 (Gaussian distribution for CLC): Given the distribution $\text{vec}(F(\mathbf{x})) \sim \mathcal{GP}(\text{vec}(\tilde{F}(\mathbf{x})), K_F(\mathbf{x}, \mathbf{x}'))$, the $CLC(\mathbf{x}, \underline{\mathbf{u}})$ is Gaussian with mean and variance:

$$\mathbb{E}[CLC(\mathbf{x}, \underline{\mathbf{u}})] = \mathbb{E}[\mathbf{q}(\mathbf{x})]^\top \underline{\mathbf{u}} \quad (17a)$$

$$Var[CLC(\mathbf{x}, \underline{\mathbf{u}})] = \underline{\mathbf{u}}^\top Var[\mathbf{q}(\mathbf{x})]\underline{\mathbf{u}}, \quad (17b)$$

where $\mathbf{q}(\mathbf{x}) := F^\top(\mathbf{x})[\nabla_{\mathbf{x}}V(\mathbf{x})] + [\alpha_V(V(\mathbf{x}))\ \ \mathbf{0}_m^\top]^\top \in \mathbb{R}^{m+1}$ and $\mathbb{E}[\mathbf{q}(\mathbf{x})]$, $Var[\mathbf{q}(\mathbf{x})]$ are computed in (18).

Proof: We can write the control Lyapunov condition as $CLC(\mathbf{x}, \underline{\mathbf{u}}) = [\nabla_{\mathbf{x}}V(\mathbf{x})]^\top F(\mathbf{x})\underline{\mathbf{u}} + \alpha_V(V(\mathbf{x})) = \mathbf{q}^\top(\mathbf{x})\underline{\mathbf{u}}$. We use the Kronecker product property $\text{vec}(\mathbf{ABC}) = (\mathbf{C}^\top \otimes \mathbf{A})\text{vec}(\mathbf{B})$ to rewrite first term in $\mathbf{q}(\mathbf{x})$ as:

$$[\nabla_{\mathbf{x}}V(\mathbf{x})]^\top F(\mathbf{x}) = (\mathbf{I}_{m+1} \otimes [\nabla_{\mathbf{x}}V(\mathbf{x})]^\top)\text{vec}(F(\mathbf{x})).$$

Since $[\nabla_{\mathbf{x}}V(\mathbf{x})], \alpha_V(V(\mathbf{x}))$ are known and deterministic and $\text{vec}(F(\mathbf{x})) \sim \mathcal{GP}(\text{vec}(\tilde{F}(\mathbf{x})), K_F(\mathbf{x}, \mathbf{x}'))$, we can express the distribution of $\mathbf{q}(\mathbf{x})$ as follows:

$$\begin{aligned} \mathbb{E}[\mathbf{q}(\mathbf{x})] &= \tilde{F}^\top(\mathbf{x})[\nabla_{\mathbf{x}}V(\mathbf{x})] + [\alpha_V(V(\mathbf{x}))\ \ \mathbf{0}_m^\top]^\top \\ Var[\mathbf{q}(\mathbf{x})] &= (\mathbf{I}_{m+1} \otimes [\nabla_{\mathbf{x}}V(\mathbf{x})]^\top)K_F(\mathbf{I}_{m+1} \otimes [\nabla_{\mathbf{x}}V(\mathbf{x})]). \end{aligned} \quad (18)$$

The result follows from plugging (18) into $CLC(\mathbf{x}, \underline{\mathbf{u}})$. ■

We use the mean and variance of $CBC(\mathbf{x}, \underline{\mathbf{u}})$ and $CLC(\mathbf{x}, \underline{\mathbf{u}})$ obtained above to approximate the probabilistic safety and stability constraints in (9).

Proposition IV.3 (Probabilistic CLF-CBF SOCP): Given a user-specified risk tolerance $p \in [0, 1)$, let $c(p) = \sqrt{\frac{p}{1-p}}$. The optimization problem (9) can be formulated as the following second-order cone program:

$$\begin{aligned} &\min_{\underline{\mathbf{u}}\in\underline{\mathcal{U}}, \delta\in\mathbb{R}, l\in\mathbb{R}} \quad l \\ &\text{s.t. } \delta - \mathbb{E}[\mathbf{q}(\mathbf{x})]^\top \underline{\mathbf{u}} \geq c(p)\sqrt{\underline{\mathbf{u}}^\top Var[\mathbf{q}(\mathbf{x})]\underline{\mathbf{u}}}, \\ &\quad \mathbb{E}[\mathbf{p}(\mathbf{x})]^\top \underline{\mathbf{u}} \geq c(p)\sqrt{\underline{\mathbf{u}}^\top Var[\mathbf{p}(\mathbf{x})]\underline{\mathbf{u}}}, \\ &\quad l + 1 \geq \sqrt{\|2L(\mathbf{x})^\top(\underline{\mathbf{u}} - \tilde{\underline{\mathbf{k}}}(\mathbf{x}))\|^2 + (2\sqrt{\lambda}\delta)^2 + (l-1)^2} \end{aligned} \quad (19)$$

where $\mathbf{p}$, $\mathbf{q}$ are defined in Propositions IV.1 and IV.2, resp.

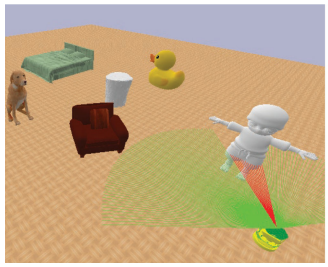
Proof: To deal with the probabilistic constraints in (9), we employ Cantelli's inequality [13]. For any scalar $\gamma \geq 0$,

$$\begin{aligned} \mathbb{P}(CBC(\mathbf{x}, \underline{\mathbf{u}}) \geq \mathbb{E}[CBC(\mathbf{x}, \underline{\mathbf{u}}))] - \gamma | \mathbf{x}, \underline{\mathbf{u}}) \\ \geq 1 - \frac{Var[CBC(\mathbf{x}, \underline{\mathbf{u}})]}{Var[CBC(\mathbf{x}, \underline{\mathbf{u}})] + \gamma^2}. \end{aligned}$$

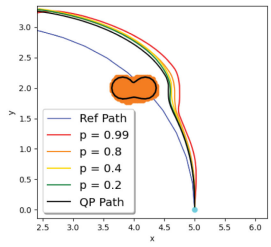
Given this inequality, and since we want $\mathbb{P}(CBC(\mathbf{x}, \underline{\mathbf{u}}) \geq 0) \geq p$, we choose $\gamma = \mathbb{E}[CBC(\mathbf{x}, \underline{\mathbf{u}})]$ and require the lower bound to be greater than or equal to p, i.e., $1 - \frac{Var[CBC(\mathbf{x}, \underline{\mathbf{u}})]}{Var[CBC(\mathbf{x}, \underline{\mathbf{u}})] + \gamma^2} \geq p$. The equation can be rearranged into

$$\mathbb{E}[CBC(\mathbf{x}, \underline{\mathbf{u}})] = \gamma \geq \sqrt{\frac{p}{1-p} Var[CBC(\mathbf{x}, \underline{\mathbf{u}})]},$$

which corresponds to the safety constraint in (19).



(a) Pybullet Simulator



(b) Probabilistic Trajectory

Fig. 1. (a) is the Pybullet simulation environment where we conduct our experiments. (b) shows the results in a region of an environment, where the probabilistic ($p = 0.2, 0.4, 0.8, 0.99$) controller and QP controller both succeed. The ground-truth obstacle surface is shown in black while the estimated obstacles is shown in orange.

Next, we show that this is a second-order cone (SOC) constraint. By (10), given that $\tilde{h}$, $\nabla\tilde{h}$ and $\tilde{F}$ are known and deterministic, the expectation $\mathbb{E}[CBC(\mathbf{x}, \underline{\mathbf{u}})] = \mathbb{E}[\mathbf{p}(\mathbf{x})]^\top \underline{\mathbf{u}}$ is affine in $\underline{\mathbf{u}}$. Since $Var[\mathbf{p}(\mathbf{x})]$ is positive semi-definite,

$$\sqrt{Var[CBC(\mathbf{x}, \underline{\mathbf{u}})]} = \sqrt{\underline{\mathbf{u}}^\top Var[\mathbf{p}(\mathbf{x})]\underline{\mathbf{u}}} = \|\mathbf{D}(\mathbf{x})\underline{\mathbf{u}}\| \quad (20)$$

where $\mathbf{D}(\mathbf{x})^\top\mathbf{D}(\mathbf{x}) = Var[\mathbf{p}(\mathbf{x})]$. Acccording to [30], the safety constraint in (19) is a valid SOC constraint.

For stability, the CLC condition can be constructed using a similar approach with Cantelli's inequality, resulting in (19). By (17), we know that the expectation is affine in $\underline{\mathbf{u}}$ and the variance is quadratic in terms of $\underline{\mathbf{u}}$, similar to (20). This shows that the CLC condition is also a valid SOC constraint.

Our last step is to reformulate the minimization of the objective function as a linear objective with an SOC constraint, resulting in the standard SOCP in (19). We introduce a new variable l so that the problem in (9) is equivalent to

$$\begin{aligned} &\min_{\underline{\mathbf{u}}\in\underline{\mathcal{U}}, \delta\in\mathbb{R}, l\in\mathbb{R}} \quad l \\ &\text{s.t.} \quad \mathbb{P}(CLC(\mathbf{x}, \underline{\mathbf{u}}) \leq \delta) \geq p, \quad \mathbb{P}(CBC(\mathbf{x}, \underline{\mathbf{u}}) \geq 0) \geq p, \\ &\quad \|L(\mathbf{x})^\top(\underline{\mathbf{u}} - \tilde{\underline{\mathbf{k}}}(\mathbf{x}))\|^2 + \lambda\delta^2 \leq l. \end{aligned} \quad (21)$$

The last constraint in (21) corresponds to a rotated second-order cone, $\mathcal{Q}^n_{rot} := \{(\mathbf{x}_r, y_r, z_r) \in \mathbb{R}^{n+2} \,|\, \|\mathbf{x}_r\|^2 \leq y_r z_r, y_r \geq 0, z_r \geq 0\}$, which can be converted into a standard SOC constraint [30], $\| \begin{bmatrix} 2\mathbf{x}_r & y_r - z_r \end{bmatrix}^\top \| \leq y_r + z_r$. Let $y_r = l$, $z_r = 1$ and consider the constraint $\|L(\mathbf{x})^\top(\underline{\mathbf{u}} - \tilde{\underline{\mathbf{k}}}(\mathbf{x}))\|^2 + \lambda\delta^2 \leq l$. Multiplying both sides by 4 and adding $(l-1)^2$, makes the constraint equivalent to

$$4\|L(\mathbf{x})^\top(\underline{\mathbf{u}} - \tilde{\underline{\mathbf{k}}}(\mathbf{x}))\|^2 + 4\lambda\delta^2 + (l-1)^2 \leq (l+1)^2.$$

Taking a square root on both sides, we end up with $\sqrt{\|2L(\mathbf{x})^\top(\underline{\mathbf{u}} - \tilde{\underline{\mathbf{k}}}(\mathbf{x}))\|^2 + (2\sqrt{\lambda}\delta)^2 + (l-1)^2} \leq l + 1$, which is equivalent to the third constraint in (19). ■

Remark IV.4 (Effects of risk-tolerance p and variance): When $p = 0$, the probabilistic CLF-CBF-SOCP (19) reduces to the original CLF-CBF-QP (6). As p and/or $Var[\mathbf{p}(\mathbf{x})]$, $Var[\mathbf{q}(\mathbf{x})]$ increase, the feasible region of (19) gets smaller, and the optimal value worsens, cf. Fig. 1(b) for an illustration.

V. Robust Safe Control

In this section, we develop a solution to Problem 2. Let $\tilde{F}$ denote the estimated system dynamics, $\tilde{h}$, $\nabla\tilde{h}$ the estimated barrier function and its gradient, and let $e_F : \mathbb{R}^{n\times(m+1)} \mapsto \mathbb{R}_{\geq 0}$, $e_h : \mathbb{R} \mapsto \mathbb{R}_{\geq 0}$, and $e_{\nabla h} : \mathbb{R}^n \mapsto \mathbb{R}_{\geq 0}$ be associated error bounds. For convenience, for each $\mathbf{x} \in \mathcal{X}$, we denote $D_F(\mathbf{x}) := F(\mathbf{x}) - \tilde{F}(\mathbf{x})$, $d_h(\mathbf{x}) := h(\mathbf{x}) - \tilde{h}(\mathbf{x})$ and $\mathbf{d}_{\nabla h}(\mathbf{x}) := \nabla h(\mathbf{x}) - \nabla\tilde{h}(\mathbf{x})$. By (7) and (8), we have

$$\|D_F(\mathbf{x})\| \leq e_F(\mathbf{x}),\ |d_h(\mathbf{x})| \leq e_h(\mathbf{x}),\ \|\mathbf{d}_{\nabla h}(\mathbf{x})\| \leq e_{\nabla h}(\mathbf{x}). \tag{22}$$

Using this notation, we can rewrite $CBC(\mathbf{x}, \underline{\mathbf{u}})$ as

$$\begin{aligned} CBC(\mathbf{x}, \underline{\mathbf{u}}) &= [\nabla h(\mathbf{x})]^\top F(\mathbf{x})\underline{\mathbf{u}} + \alpha_h(h(\mathbf{x})) \\ &= [\nabla\tilde{h}(\mathbf{x})]^\top \tilde{F}(\mathbf{x})\underline{\mathbf{u}} + \mathbf{d}_{\nabla h}^\top(\mathbf{x})\tilde{F}(\mathbf{x})\underline{\mathbf{u}} \\ &\quad + [\nabla\tilde{h}(\mathbf{x})]^\top D_F(\mathbf{x})\underline{\mathbf{u}} + \mathbf{d}_{\nabla h}^\top(\mathbf{x})D_F(\mathbf{x})\underline{\mathbf{u}} \\ &\quad + \alpha_h(\tilde{h}(\mathbf{x}) + d_h(\mathbf{x})). \end{aligned}$$

Let $\tilde{\mathbf{p}}(\mathbf{x}) := \tilde{F}^\top(\mathbf{x})\nabla\tilde{h}(\mathbf{x})$. We group the error term in the expression for $CBC(\mathbf{x}, \underline{\mathbf{u}})$ in the variable $d_{CBC}(\mathbf{x}, \underline{\mathbf{u}}) := CBC(\mathbf{x}, \underline{\mathbf{u}}) - \tilde{\mathbf{p}}(\mathbf{x})^\top\underline{\mathbf{u}}$. Thus, $CBC(\mathbf{x}, \underline{\mathbf{u}}) \geq 0$ is satisfied if

$$\min_{D_F, \mathbf{d}_{\nabla h}, d_h} CBC(\mathbf{x}, \underline{\mathbf{u}}) = \tilde{\mathbf{p}}(\mathbf{x})^\top\underline{\mathbf{u}} + \min_{D_F, \mathbf{d}_{\nabla h}, d_h} d_{CBC}(\mathbf{x}, \underline{\mathbf{u}}) \geq 0.$$

Similarly, let $\tilde{\mathbf{q}}(\mathbf{x}) := \tilde{F}^\top(\mathbf{x})\nabla V(\mathbf{x}) + [\alpha_V(V(\mathbf{x}))\ \ \mathbf{0}_m^\top]^\top$ and $d_{CLC}(\mathbf{x}, \underline{\mathbf{u}}) := [\nabla V(\mathbf{x})]^\top D_F(\mathbf{x})\underline{\mathbf{u}}$, a robust version of the stability constraint $CLC(\mathbf{x}, \underline{\mathbf{u}}) \leq \delta$ can be written as:

$$\max_{D_F} CLC(\mathbf{x}, \underline{\mathbf{u}}) = \tilde{\mathbf{q}}(\mathbf{x})^\top\underline{\mathbf{u}} + \max_{D_F} d_{CLC}(\mathbf{x}, \underline{\mathbf{u}}) \leq \delta. \tag{23}$$

This leads us to the following robust reformulation of the original control synthesis problem in (6),

$$\begin{aligned} \min_{\underline{\mathbf{u}}\in\underline{\mathcal{U}}, \delta\in\mathbb{R}, l\in\mathbb{R}}\ & l \\ \text{s.t. } & \tilde{\mathbf{q}}(\mathbf{x})^\top\underline{\mathbf{u}} + \max_{D_F}\ d_{CLC}(\mathbf{x}, \underline{\mathbf{u}}) \leq \delta \\ & \tilde{\mathbf{p}}(\mathbf{x})^\top\underline{\mathbf{u}} + \min_{D_F, d_h, \mathbf{d}_{\nabla h}} d_{CBC}(\mathbf{x}, \underline{\mathbf{u}}) \geq 0 \\ & l + 1 \geq \sqrt{\|2L(\mathbf{x})^\top(\underline{\mathbf{u}} - \underline{\tilde{\mathbf{k}}}(\mathbf{x}))\|^2 + (2\sqrt{\lambda}\delta)^2 + (l-1)^2}. \end{aligned} \tag{24}$$

Note that we used the same approach as in the proof of Proposition IV.3 to reformulate the original quadratic objective with a linear objective plus a SOC constraint. The second constraint in (24) requires solving $\min_{D_F, d_h, \mathbf{d}_{\nabla h}} d_{CBC}(\mathbf{x}, \underline{\mathbf{u}})$ subject to (22). In general, this is a non-convex constrained quadratic program which does not have a closed-form expression of the minimizer as a function of $\underline{\mathbf{u}}$. Instead, we make the second constraint in (24) more conservative using the Cauchy-Schwarz inequality, which leads to a convex SOCP, whose optimal solution is guaranteed to be feasible for (24).

Proposition V.1 (Robust CLF-CBF SOCP): Let $\tilde{F}$, $\tilde{h}$, $\nabla\tilde{h}$ denote estimates of the system dynamics and barrier function, with error bounds in (22). Then, the feasible set of the following SOCP is included in the feasible set of (24):

$$\begin{aligned} \min_{\underline{\mathbf{u}}\in\underline{\mathcal{U}}, \delta\in\mathbb{R}, p\in\mathbb{R}, q\in\mathbb{R}, l\in\mathbb{R}}\ & l \\ \text{s.t. } & \delta - \tilde{\mathbf{q}}(\mathbf{x})^\top\underline{\mathbf{u}} \geq e_F(\mathbf{x})\|\nabla V(\mathbf{x})\|\|\underline{\mathbf{u}}\|, \\ & p \geq e_{\nabla h}(\mathbf{x})\|\tilde{F}(\mathbf{x})\underline{\mathbf{u}}\|, \\ & q \geq \Big(e_F(\mathbf{x})\|\nabla\tilde{h}(\mathbf{x})\| + e_{\nabla h}(\mathbf{x})e_F(\mathbf{x})\Big)\|\underline{\mathbf{u}}\|, \\ & [\nabla\tilde{h}(\mathbf{x})]^\top\tilde{F}(\mathbf{x})\underline{\mathbf{u}} + \alpha_h(\tilde{h}(\mathbf{x}) - e_h(\mathbf{x})) \geq p + q, \\ & l + 1 \geq \sqrt{\|2L(\mathbf{x})^\top(\underline{\mathbf{u}} - \underline{\tilde{\mathbf{k}}}(\mathbf{x}))\|^2 + (2\sqrt{\lambda}\delta)^2 + (l-1)^2} \end{aligned} \tag{25}$$

Proof: The stability constraint in (24) is reformulated using:

$$\max_{\|D_F(\mathbf{x})\| \leq e_F(\mathbf{x})} d_{CLC}(\mathbf{x}, \underline{\mathbf{u}}) = e_F(\mathbf{x})\|\nabla V(\mathbf{x})\|\|\underline{\mathbf{u}}\|.$$

For the safety constraint in (24), note that

$$\begin{aligned} &\min_{D_F, d_h, \mathbf{d}_{\nabla h}} d_{CBC}(\mathbf{x}, \underline{\mathbf{u}}) \\ &= \min_{D_F, \mathbf{d}_{\nabla h}} \Big(\mathbf{d}_{\nabla h}^\top(\mathbf{x})\tilde{F}(\mathbf{x})\underline{\mathbf{u}} + [\nabla\tilde{h}(\mathbf{x})]^\top D_F(\mathbf{x})\underline{\mathbf{u}} \\ &\qquad + \mathbf{d}_{\nabla h}^\top(\mathbf{x})D_F(\mathbf{x})\underline{\mathbf{u}}\Big) + \min_{d_h} \alpha_h(\tilde{h}(\mathbf{x}) + d_h(\mathbf{x})). \end{aligned} \tag{26}$$

Since $e_h(\mathbf{x}) \geq 0$ and α_h is an extended class $\mathcal{K}_\infty$ function,

$$\min_{|d_h(\mathbf{x})| \leq e_h(\mathbf{x})} \alpha_h(\tilde{h}(\mathbf{x}) + d_h(\mathbf{x})) = \alpha_h(\tilde{h}(\mathbf{x}) - e_h(\mathbf{x})). \tag{27}$$

Applying the Cauchy-Schwarz inequality on each term,

$$\begin{aligned} &\min_{D_F, d_h, \mathbf{d}_{\nabla h}} d_{CBC}(\mathbf{x}, \underline{\mathbf{u}}) \geq -\|\mathbf{d}_{\nabla h}\|\|\tilde{F}(\mathbf{x})\underline{\mathbf{u}}\| \\ &\quad - \|\nabla\tilde{h}(\mathbf{x})\|\|D_F(\mathbf{x})\underline{\mathbf{u}}\| - \|\mathbf{d}_{\nabla h}(\mathbf{x})\|\|D_F(\mathbf{x})\underline{\mathbf{u}}\| \\ &\quad + \alpha_h(\tilde{h}(\mathbf{x}) - e_h(\mathbf{x})) \\ &\geq -e_{\nabla h}(\mathbf{x})\|\tilde{F}(\mathbf{x})\underline{\mathbf{u}}\| - e_F(\mathbf{x})\|\nabla\tilde{h}(\mathbf{x})\|\|\underline{\mathbf{u}}\| \\ &\quad - e_{\nabla h}(\mathbf{x})e_F(\mathbf{x})\|\underline{\mathbf{u}}\| + \alpha_h(\tilde{h}(\mathbf{x}) - e_h(\mathbf{x})). \end{aligned}$$

In the last step, we minimized each term independently, so the lower bound is not tight. We write the safety constraint as

$$\begin{aligned} &e_{\nabla h}(\mathbf{x})\|\tilde{F}(\mathbf{x})\underline{\mathbf{u}}\| + (e_F(\mathbf{x})\|\nabla\tilde{h}(\mathbf{x})\| + e_{\nabla h}(\mathbf{x})e_F(\mathbf{x}))\|\underline{\mathbf{u}}\| \\ &\quad \leq [\nabla\tilde{h}(\mathbf{x})]^\top\tilde{F}(\mathbf{x})\underline{\mathbf{u}} + \alpha_h(\tilde{h}(\mathbf{x}) - e_h(\mathbf{x})). \end{aligned} \tag{28}$$

Constraints of the form $\|\mathbf{A}\mathbf{z} - \boldsymbol{a}\| + \|\mathbf{B}\mathbf{z} - \boldsymbol{b}\| \leq \boldsymbol{c}^\top\mathbf{z}$ can be replaced by the set of constraints $\|\mathbf{A}\mathbf{z} - \boldsymbol{a}\| \leq p$, $\|\mathbf{B}\mathbf{z} - \boldsymbol{b}\| \leq q$, $p + q \leq \boldsymbol{c}^\top\mathbf{z}$ combined. Thus, (28) is equivalent to the second, third, and fourth constraints in (25) together. ■

Remark V.2 (Effects of error bounds): If there are no errors in either the dynamics or the barrier function ($e_F \equiv e_h \equiv e_{\nabla h} \equiv 0$), then the robust CLF-CBF SOCP (25) reduces to a CLF-CBF QP (6). If $e_F \equiv 0$ while $e_h(\mathbf{x}), e_{\nabla h}(\mathbf{x}) > 0$, the result in Proposition V.1 recovers [29, Proposition 2]. As the error bounds $e_F, e_h, e_{\nabla h}$ increase, the feasible region of (25) gets smaller and the optimal solution worsens. Also, note that the choice of kernel function, $K_F(\mathbf{x}, \mathbf{x}) = \frac{e_F^2(\mathbf{x})}{c^2(p)}\mathbf{I}_{(m+1)n}$, reduces the inequality for stability in (19) to that in (25).

VI. Evaluation

In this section, we present an approach to estimate the unknown dynamics of a mobile robot, and construct CBF constraints online. Then, we evaluate our safe control synthesis using the estimated robot dynamics and CBFs in autonomous navigation tasks in 10 simulated environments, containing obstacles a priori unknown to the robot.

A. System Dynamics Estimation

We consider a Turtlebot robot simulated in the PyBullet simulator [31] (see Fig. 1(a)). We first present a learning approach to model the unknown dynamics of the TurtleBot using training data collected from the PyBullet simulator. The robot state and input are $\mathbf{x} := [x, y, \mu]^\top \in \mathbb{R}^2 \times [-\pi, \pi)$ and $\underline{\mathbf{u}} := [1, v, \omega]^\top \in \{1\} \times \mathbb{R}^2$, respectively. We collect a dataset $\mathcal{D} = \{t^{(i)}_{0:N}, \mathbf{x}^{(i)}_{0:N}, \underline{\mathbf{u}}^{(i)}_{0:N}\}_{i=1}^{D}$ of $D = 40000$ state sequences $\mathbf{x}^{(i)}_{0:N}$ obtained by applying random control inputs $\underline{\mathbf{u}}^{(i)}_{0:N}$ to the robot with initial condition $\mathbf{x}^{(i)}_0$ at time intervals of $\tau = 0.02$ seconds. For each trajectory i, a constant control input is applied for $N = 5$ time steps.

We employ a neural ODE network [32] to approximate the unknown robot dynamics F with a neural network $F_{\boldsymbol{\theta}}$ based on the dataset $\mathcal{D}$. A forward pass through the ODE network is obtained using an ODE solver:

$$\{\tilde{\mathbf{x}}^i_1, \tilde{\mathbf{x}}^i_2, \ldots, \tilde{\mathbf{x}}^i_N\} = \text{ODESolve}(\mathbf{x}^i_0, F_{\boldsymbol{\theta}}(\cdot)\underline{\mathbf{u}}^i, t^i_1, \ldots, t^i_N).$$

We use a loss function,

$$\min_{\boldsymbol{\theta}} \sum_{i=1}^{D}\sum_{j=1}^{N} \ell(\mathbf{x}^{(i)}_j, \tilde{\mathbf{x}}^{(i)}_j),$$

$$\text{s.t. } \dot{\tilde{\mathbf{x}}}^{(i)}(t) = F_{\boldsymbol{\theta}}(\tilde{\mathbf{x}}^{(i)}(t))\underline{\mathbf{u}}^{(i)}(t), \quad \tilde{\mathbf{x}}^{(i)}(j\tau) = \tilde{\mathbf{x}}^{(i)}_j,$$

$$\underline{\mathbf{u}}^{(i)}(t) \equiv \underline{\mathbf{u}}^{(i)}_j \quad \text{for } t \in [j\tau, (j+1)\tau), \tag{29}$$

where $\ell(\mathbf{x}, \tilde{\mathbf{x}}) = \|[x, y, \cos\mu, \sin\mu]^\top - [\tilde{x}, \tilde{y}, \cos\tilde{\mu}, \sin\tilde{\mu}]^\top\|^2$. To update the weights $\boldsymbol{\theta}$, the gradient of the loss function is back-propagated by solving another ODE with adjoint states backwards in time. Please refer to [32] for details.

Gal and Ghahramani [33] showed that introducing dropout layers in a neural network is approximately equivalent to performing deep Gaussian Process regression. We use a 6-layer fully-connected neural network with $\tanh$ activations and 800 neurons in each layer to model $F_{\boldsymbol{\theta}}$, and apply dropout to each hidden layer with rate 0.05. Given a query state $\mathbf{x} \in \mathcal{X}$, Monte-Carlo estimates of the predictive mean $\tilde{F}_{\boldsymbol{\theta}}(\mathbf{x})$ and element-wise standard deviation $\tilde{\Sigma}(\mathbf{x})$ of the dynamics are obtained with $T = 100$ stochastic forward passes through the dropout neural network model. We use $\tilde{F}_{\boldsymbol{\theta}}(\mathbf{x})$ for the mean of system dynamics and $K_F(\mathbf{x}, \mathbf{x}) = \text{diag}(\text{vec}(\tilde{\Sigma}(\mathbf{x}))^2)$ for the variance of the dynamics. To obtain worst-case error bounds $e_F(\mathbf{x})$, we set $e_F(\mathbf{x}) = \|3.89\tilde{\Sigma}(\mathbf{x})\|$ (99.99% *confidence*).

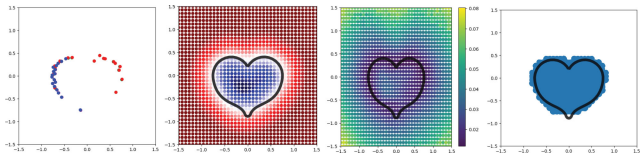
In our experiment, no external disturbances are added to the system dynamics model. Given $M = 5000$ random-sampled different state control sequences $\{\mathbf{x}_i, \underline{\mathbf{u}}_i\}_{i=1}^{M}$ as test data, we consider the following test-time loss function, $L = \frac{1}{M}\sum_{i=1}^{M} \ell(F(\mathbf{x}_i)\underline{\mathbf{u}}_i, \tilde{F}(\mathbf{x}_i)\underline{\mathbf{u}}_i)$. Our learned dynamics model is quite accurate, and the average test loss is $L = 0.0037$.

B. Online CBF Estimation

The robot is equipped with a LiDAR scanner with a 270° field of view, 200 rays per scan, 3 m range, and zero-mean Gaussian measurement noise with standard deviation $\sigma \in \{0.01, 0.02, 0.05\}$. The LiDAR scans are used to estimate the unsafe regions $\mathcal{O}_i$ in the environment and construct a CBF constraint for each. We rely on the concept of signed distance function (SDF) (e.g. Fig. 2(b)) to describe each $\mathcal{O}_i$. The SDF

TABLE I
EMPIRICAL SDF ESTIMATION ERROR $\mathcal{E}$ AND DROPOUT-NETWORK SDF ESTIMATION ERROR AVERAGED ACROSS 8 OBJECT INSTANCES UNDER DIFFERENT LIDAR MEASUREMENT NOISE STANDARD DEVIATION σ

LiDAR Noise σ	SDF Empirical Error	SDF Dropout Error
0.01	0.0173	0.0132
0.02	0.0288	0.0184
0.05	0.0463	0.0242



(a) Training data (b) Mean SDF (c) Variance (d) $\mathbb{P}(\tilde{\varphi} \leq 0) = 0.95$

Fig. 2. Shape estimation with dropout neural network. (a) shows the training data. (b) shows the estimated mean SDF results. The black heart curve shows the ground-truth obstacle boundary, while colored regions are level-sets of the SDF estimate. The white region denotes the estimated obstacle boundary. The blue (resp. red) region denotes negative (resp. positive) signed distance. In (c), the variance of the SDF estimate is shown. In (d), we plot the estimated unsafe region with high probability, where $\mathbb{P}(\tilde{\varphi} \leq 0) = 0.95$.

TABLE II
SUCCESS RATE OF THE NAVIGATION TASKS IN 100 REALIZATIONS (10 REALIZATIONS FOR EACH OF THE 10 DIFFERENT ENVIRONMENTS) USING THE PROBABILISTIC CLF-CBF-SOCP, ROBUST CLF-CBF-SOCP, AND THE ORIGINAL CLF-CBF-QP FRAMEWORKS FOR DIFFERENT LIDAR MEASUREMENT NOISE LEVELS σ

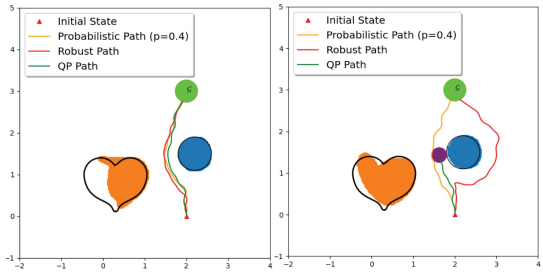
LiDAR Noise σ	QP Success Rate	Probabilistic Success Rate			Robust Success Rate
		$p = 0.2$	$p = 0.4$	$p = 0.8$	
0.01	0.82	0.98	1.0	1.0	1.0
0.02	0.65	0.92	0.97	1.0	1.0
0.05	0.37	0.72	0.89	0.96	1.0

function $\varphi_i : \mathbb{R}^2 \mapsto \mathbb{R}$ of set $\mathcal{O}_i \subseteq \mathbb{R}^2$ is

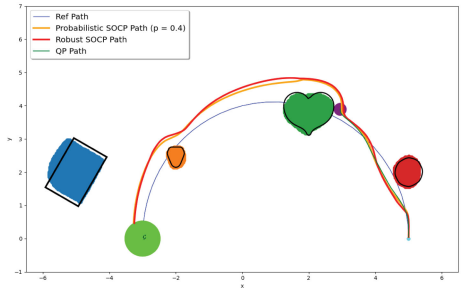
$$\varphi_i(\mathbf{y}) := \begin{cases} -d(\mathbf{y}, \partial\mathcal{O}_i), & \mathbf{y} \in \mathcal{O}_i, \\ d(\mathbf{y}, \partial\mathcal{O}_i), & \mathbf{y} \notin \mathcal{O}_i, \end{cases} \tag{30}$$

where d denotes the Euclidean distance from a point $\mathbf{y} \in \mathbb{R}^2$ and the set boundary $\partial\mathcal{O}_i$. We employ incremental training with replay memory (ITRM) [29, Section IV] to estimate an SDF φ_i for each $\mathcal{O}_i$ from the LiDAR measurements. We use a 4-layer fully-connected neural network with parameters $\boldsymbol{\theta}$ and dropout layers to yield $\tilde{\varphi}_i(\mathbf{y}; \boldsymbol{\theta})$ with dropout rate 0.05 applied to each 512-neuron hidden layer. Given $\mathbf{y} \in \mathbb{R}^2$, we obtain the predictive SDF mean $\hat{\varphi}_i(\mathbf{y})$ and standard deviation $\hat{\sigma}_i(\mathbf{y})$ by Monte-Carlo estimation with $T = 20$ stochastic forward passes through the dropout neural network model. When the TurtleBot moves along a circle of radius 2 while the object is placed at the center, we measure the accuracy of the online SDF method using the empirical SDF error, $\mathcal{E}_i = \frac{1}{m}\sum_{j=1}^{m} |\hat{\varphi}_i(\mathbf{y}_j)|$, where $\{\mathbf{y}_j\}_{j=1}^{m}$ are $m = 500$ points uniformly sampled on the surface of the object. The results are shown in Table 1. In Fig. 2, we show the SDF estimation with measurement noise $\sigma = 0.01$.

Since we deal with system dynamics with relative degree one, one can verify [34] that the SDF is a valid CBF. Let $\mathbf{z} = [x, y] \in \mathcal{Z} \subset \mathbb{R}^2$ be the position part of $\mathbf{x}$. To account for the fact that the robot body is not a point mass, we subtract the robot radius $\rho = 0.177$ from each SDF estimate when defining each mean CBF: $\tilde{h}_i(\mathbf{x}) = \tilde{\varphi}_i(\mathbf{z}; \boldsymbol{\theta}) - \rho$. For variance



(a) Noise: $\sigma = 0.01$ (b) Noise: $\sigma = 0.02$



(c) Safe Trajectory Tracking

Fig. 3. Performance comparison among the three controllers. Ground-truth obstacle surfaces are shown as black curves. The mean of the estimated obstacles, obtained after the whole path is traversed by the probabilistic CLF-CBF-SOCP controller are shown in different colors (red, green, orange, blue). The trajectories generated by the probabilistic and robust CLF-CBF-SOCP controllers are in red and blue, respectively, while the CLF-CBF-QP trajectory is in green. The starting point is cyan and the goal region is a light-green disk. The robot (purple disk in (b) and (c)), controlled by the CLF-CBF-QP controller, collides with obstacles and does not reach the goal. In (a) and (b), we compare the controller performance under different LiDAR noise level for a same environment. In (a), the results are collected under LiDAR measurement noise $\sigma = 0.01$. In (b), the results are collected under LiDAR noise $\sigma = 0.02$. In (c), the trajectory tracking results of environment 8 is shown and the reference path is shown in blue.

$K_h(\mathbf{x}, \mathbf{x})$ in Section IV, we set $K_h^i(\mathbf{x}, \mathbf{x}) = \hat{\sigma}_i^2(\mathbf{z})$. We also take $\nabla \tilde{h}_i(\mathbf{x}) = \nabla \tilde{\varphi}_i(\mathbf{z}; \boldsymbol{\theta})$ and compute $\mathcal{H}_{\mathbf{x},\mathbf{x}'} K_h^i(\mathbf{x}, \mathbf{x}')$ by Monte-Carlo estimation using double back-propagation. We set the worst case error bounds $e_h(\mathbf{x})$, $e_{\nabla h}(\mathbf{x})$ in Section V as the 99.99% confidence bounds of a Gaussian random variable with standard deviation $\hat{\sigma}_i(\mathbf{z})$. If the robot observes multiple obstacles in the environment, we compute multiple CBFs $\tilde{h}_i(\mathbf{x})$ and their corresponding uncertainty, and add multiple CBCs to (6), (19), (25) for safe control synthesis.

C. Safe Navigation

Our main experiments demonstrate safe navigation and safe trajectory tracking using the proposed probabilistic (19) and robust (25) CLF-CBF-SOCP formulations, utilizing the dynamics estimates from Section VI.A and the online CBF estimates from Section VI.B. To emphasize the importance of accounting for estimation errors, we also implement the original CLF-CBF-QP controller (6), which assumes the estimated barrier functions and system dynamics are accurate (i.e., uses the mean values from the dropout-network estimation as the true values). In all three controllers, we set $L(\mathbf{x}) = \text{diag}([0, 10, 3])$ and $\underline{\tilde{\mathbf{k}}}(\mathbf{x}) = [1, v_{\max}, 0]^\top$ where $v_{\max} = 0.65$ is the maximum linear velocity for the TurtleBot. The remaining parameters were $\lambda = 1000$, $\alpha_V(V(\mathbf{x})) = 2V(\mathbf{x})$, and $\alpha_h(h_i(\mathbf{x})) = h_i(\mathbf{x})$.

TABLE III
FRÉCHET DISTANCE BETWEEN THE REFERENCE PATH AND THE ROBOT TRAJECTORIES GENERATED BY THE PROBABILISTIC CLF-CBF-SOCP, ROBUST CLF-CBF-SOCP, AND THE CLF-CBF-QP CONTROLLERS (SMALLER VALUES INDICATE LARGER TRAJECTORY SIMILARITY, THE VALUE IN THE PARENTHESES INDICATES THE SUCCESS RATES WHILE VALUES WITHOUT PARENTHESES INDICATE THE SUCCESS RATE IS 1, AND N/A INDICATES THE ROBOT COLLIDES WITH OBSTACLES IN ALL 10 REALIZATIONS)

Env	QP	Probabilistic			Robust
		$p = 0.2$	$p = 0.4$	$p = 0.8$	
1	0.337	0.338	0.343	0.363	0.357
2	0.378	0.408	0.404	0.432	0.485
3	0.372	0.398	0.412	0.457	0.538
4	0.416	0.438	0.427	0.473	0.515
5	0.395	0.418	0.412	0.483	0.572
6	0.385 (0.8)	0.371	0.378	0.392	0.424
7	0.462 (0.5)	0.502	0.546	0.593	0.737
8	0.535 (0.2)	0.588	0.612	0.673	0.814
9	N/A	0.756 (0.8)	0.887 (0.9)	0.926	1.016
10	N/A	0.905 (0.4)	0.937 (0.8)	1.046	1.224

In the first set of experiments (Fig. 3(a) and Fig. 3(b)), we demonstrate safe navigation to a goal point with a CLF candidate $V(\mathbf{x}) = (x-2)^2 + (y-3)^2$. In Fig. 3(a), when the LiDAR noise level is low, the robot controlled by all three controllers succeeds to reach the goal region and the SOCP formulations are slightly more conservative than the QP formulation. In Fig. 3(b), the LiDAR noise level increases to $\sigma = 0.02$ and we can observe major differences among the paths generated by the three controllers. This is because the estimated variance and error bounds of the barrier function and its gradient increase with the increase of the LiDAR noise. The robot controlled by the CLF-CBF-QP controller collides with an obstacle, while the robot controlled by probabilistic or robust SOCP controller succeeds in avoiding obstacles. Importantly, the robot controlled by the robust SOCP controller switches to bypass the round obstacle from the right because controller cannot find a feasible path on the left with larger error bounds on estimated barrier functions.

In the following set of experiments, we consider the problem of safe trajectory tracking using the approach in [29, Section VI] to construct a valid CLF $V(\mathbf{x})$ for path following.

In Table II, we report the success rate of the trajectory tracking task using the proposed formulations and the original QP framework under different measurement noises. As the noise increases, the success rate of the CLF-CBF-QP controller decreases rapidly, while the success rate of the probabilistic framework with high p and the robust framework stays high.

In Fig. 3(c), we show one realization in environment 8 (with $\sigma = 0.01$), where the CLF-CBF-QP controller fails to avoid obstacles because it does not consider the errors in $CBC(\mathbf{x}, \underline{\mathbf{u}})$, while the proposed frameworks guarantee safety. When there is an obstacle near or on the reference path, the robot controlled by the robust SOCP controller stays furthest away, while the probabilistic SOCP controller also guarantees the robot stays further away from the obstacles than the robot controlled by the CLF-CBF-QP controller.

In Table III, we show quantitative results using the Fréchet distance [29, Section VI] as the metric to measure trajectory similarities. The distance value is computed by averaging the successful realizations in each environment, and the LiDAR noise is set to be $\sigma = 0.02$ in this set of experiments. We see that the robust CLF-CBF-SOCP controller is most conservative as it has the largest Fréchet distance values while the probabilistic CLF-CBF-SOCP controller is less conservative if we set

the user-specified risk tolerance $p = 0.8$. By lowering the risk tolerance value ($p = 0.2/0.4$), the robot with the probabilistic controller follows the reference path better while facing a higher risk of collision. A qualitative result is shown in Fig. 2(b), where larger p values indicates higher probability of being safe for the robot. The trajectory generated by the CLF-CBF-QP controller has the smallest Fréchet distance values, but fails in several environments.

Finally, to demonstrate the efficiency of the proposed formulations, we compare the average time needed for solving the QP, probabilistic SOCP, and robust SOCP formulations per control synthesis along the trajectory tracking task. All optimization problems are solved using the Embedded Conic Solver in CVXPY [35] with an Intel i7 9700 K CPU. The time needed for solving one QP instance is 0.00863 s while the times needed for solving the proposed probabilistic and robust SOCPs are 0.0109 s and 0.0122 s. As expected, our SOCP formulations require slightly more time than the original QP but are still suitable for online robot navigation.

VII. Conclusion

We considered the problem of enforcing safety and stability of unknown robot systems operating in unknown environments. We showed that accounting for either Gaussian or worst-case error bounds in the system dynamics and safety constraints leads to a novel CLF-CBF-SOCP formulation for control synthesis. We validated our formulations in autonomous navigation tasks, simulating a ground robot in several unknown environments. Some drawbacks of our formulations include that large model error bounds may lead to infeasibility of the robust SOCP, and that the assumption that system dynamics and barrier functions are GPs may not be true in practice. Future work will implement the proposed formulations on a real robot, consider object category pre-training of the SDF neural network, and explore adaptive techniques for safe control synthesis given varying uncertainty levels and robot objectives.

References

[1] L. Lamport, "Proving the correctness of multiprocess programs," *IEEE Trans. Softw. Eng.*, vol. SE-3, no. 2, pp. 125–143, Mar. 1977.

[2] Z. Artstein, "Stabilization with relaxed controls," *Nonlinear Anal.-Theory Methods Appl.*, vol. 7, pp. 1163–1173, 1983.

[3] E. D. Sontag, "A 'universal' construction of Artstein's theorem on nonlinear stabilization," *Syst. Control Lett.*, vol. 13, no. 2, pp. 117–123, 1989.

[4] S. Prajna, "Barrier certificates for nonlinear model validation," in *Proc. Conf. Decis. Control*, 2003, pp. 2884–2889.

[5] S. Prajna and A. Jadbabaie, "Safety Verification of hybrid systems using barrier certificates," in *Proc. Hybrid Systems: Computation and Control*, Berlin Heidelberg, Germany, Springer, 2004, pp. 477–492.

[6] P. Wieland and F. Allgöwer, "Constructive safety using control barrier functions," in *IFAC Proc. Volumes*, vol. 40, no. 12, pp. 462–467, 2007.

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

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

[9] A. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *Proc. Eur. Control Conf.*, 2019, pp. 3420–3431.

[10] L. Wang, A. D. Ames, and M. Egerstedt, "Safe certificate-based maneuvers for teams of quadrotors using differential flatness," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2017, pp. 3293–3298.

[11] Q. Nguyen, A. Hereid, J. W. Grizzle, A. D. Ames, and K. Sreenath, "3D dynamic walking on stepping stones with control barrier functions," in *Proc. IEEE Conf. Decis. Control*, 2016, pp. 827–834.

[12] X. Xu *et al.*, "Realizing simultaneous lane keeping and adaptive speed regulation on accessible mobile robot testbeds," in *Proc. IEEE Conf. Control Technol. Appl.*, 2017, pp. 1769–1775.

[13] F. P. Cantelli, "Sui confini della probabilità," *Atti del Congresso Internazionale dei Matematici*, vol. 6, pp. 47–60, 1929.

[14] M. Jankovic, "Robust control barrier functions for constrained stabilization of nonlinear systems," *Automatica*, vol. 96, pp. 359–367, 2018.

[15] Y. Emam, P. Glotfelter, and M. Egerstedt, "Robust barrier functions for a fully autonomous, remotely accessible swarm-robotics testbed," in *Proc. IEEE Conf. Decis. Control*, 2019, pp. 3984–3990.

[16] A. Clark, "Control barrier functions for complete and incomplete information stochastic systems," in *Proc. Amer. Control Conf.*, 2019, pp. 2928–2935.

[17] Q. Nguyen and K. Sreenath, "Robust safety-critical control for dynamic robotics," *IEEE Trans. Autom. Control*, vol. 67, no. 3, pp. 1073–1088, Mar. 2022.

[18] L. Hewing, J. Kabzan, and M. N. Zeilinger, "Cautious model predictive control using Gaussian process regression," *IEEE Trans. Control Syst. Technol.*, vol. 28, no. 6, pp. 2736–2743, Nov. 2020.

[19] M. Ahmadi, X. Xiong, and A. D. Ames, "Risk-averse control via CVaR barrier functions: Application to bipedal robot locomotion," *IEEE Control Syst. Lett.*, vol. 6, pp. 878–883, 2022.

[20] G. Alcan and V. Kyrki, "Differential dynamic programming with nonlinear safety constraints under system uncertainties," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 1760–1767, Apr. 2022.

[21] H. Almubarak, K. Stachowicz, N. Sadegh, and E. A. Theodorou, "Safety embedded differential dynamic programming using discrete barrier states," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 2755–2762, Apr. 2022.

[22] M. Z. Romdlony and B. Jayawardhana, "On the new notion of input-to-state safety," in *Proc. IEEE 55th Conf. Decis. Control*, 2016, pp. 6403–6409.

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

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

[25] 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 *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 6286–6291.

[26] M. Srinivasan, A. Dabholkar, S. Coogan, and P. Vela, "Synthesis of control barrier functions using a supervised machine learning approach," *IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 7139–7145.

[27] T. T. Zhang, S. Tu, N. M. Boffi, J.-J. E. Slotine, and N. Matni, "Adversarially robust stability certificates can be sample-efficient," in *L4DC*, 2022.

[28] V. Dhiman*, M. J. Khojasteh*, M. Franceschetti, and N. Atanasov, "Control barriers in Bayesian learning of system dynamics," *IEEE Trans. Autom. Control*, to be published, doi: 10.1109/TAC.2021.3137059.

[29] K. Long, C. Qian, J. Cortés, and N. Atanasov, "Learning barrier functions with memory for robust safe navigation," *IEEE Robot. Automat. Lett.*, vol. 6, no. 3, pp. 4931–4938, Jul. 2021.

[30] F. Alizadeh and D. Goldfarb, "Second-order cone programming," *Math. Program.*, vol. 95, no. 1, pp. 3–51, 2003.

[31] E. Coumans and Y. Bai, "PyBullet, a python module for physics simulation for games, robotics and machine learning," 2016. [Online]. Available: http://pybullet.org

[32] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud, "Neural ordinary differential equations," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 6572–6583.

[33] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," in *Proc. Int. Conf. Mach. Learn.*, 2016, vol. 48, pp. 1050–1059.

[34] Y. Chen, M. Jankovic, M. A. Santillo, and A. Ames, "Backup control barrier functions: Formulation and comparative study," in *Proc. 60th IEEE Conf. Decis. Control*, 2021, pp. 6835–6841.

[35] S. Diamond and S. Boyd, "CVXPY: A python-embedded modeling language for convex optimization," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 2909–2913, 2016.