Safe Control Synthesis with Uncertain Dynamics and Constraints

Kehan Long1 Vikas Dhiman2 Melvin Leok1 Jorge Cortés1 Nikolay Atanasov1

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.

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 formally proposed to 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 safety and 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 desired 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 (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 safe control Lyapunov condition (CLC) for safety of discrete-time systems. Another line of research formulates safe control synthesis using a Support Vector Machine and solve a CLF-CBF QP for safety of discrete-time systems. Another line of research formulates safe control synthesis using a Support Vector Machine and solve a CLF-CBF QP for safety of discrete-time systems.

Input-to-state safety (ISSf) was introduced in [21] to handle input disturbances and was used in [22] to enlarge a safe set by modifying a CBF. Alan et al. [23] introduce a tunable ISSf-CBF for safe control synthesis while reducing conservatism. Cosner et al. in [24] 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:

$$\dot{x} = f(x) + g(x)u = \begin{bmatrix} f(x) \\ g(x) \end{bmatrix} \cdot \begin{bmatrix} 1 \\ u \end{bmatrix} \triangleq F(x)u, \quad (1)$$

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

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

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

where the control Lyapunov condition (CLC) is:

$$CLC(x, u) \triangleq L_f V(x) + L_g V(x)u + \alpha_V(V(x)) \quad (3)$$

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 $h : \mathbb{R}^n \to \mathbb{R}$, which implicitly defines a (closed) safe set of system states $\mathcal{S} \triangleq \{x \in \mathcal{X} \mid h(x) \geq 0\}$. 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 $h : \mathbb{R}^n \to \mathbb{R}$ 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 $\alpha_h$ with:

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

where the control barrier condition (CBC) is:

$$CBC(x, u) \triangleq L_f h(x) + L_g h(x)u + \alpha_h(h(x)) \quad (5)$$

According to [7], [9], any Lipschitz-continuous controller $\tilde{k} : \mathcal{X} \to \mathcal{U}$ that satisfies $CBC(x, \tilde{k}(x)) \geq 0$ for all $x \in \mathcal{X}$ 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 $F(x)$ and barrier function $h(x)$ are known, a safe controller can be synthesized by combining CLF and CBF constraints in a quadratic program:

$$\min_{u \in \mathcal{U}, \delta \in \mathbb{R}} \|L(x) (u - \tilde{k}(x))\|^2 + \lambda \delta^2, \quad \text{s.t. } CLC(x, u) \leq \delta, \ CBC(x, u) \geq 0. \quad (6)$$

The term $\tilde{k}(x)$ 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 $\tilde{k}(x) \equiv e_1$ if minimum control effort is the main objective. The term $L(x)$ 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

**Notation:** We denote by $I_n \in \mathbb{R}^{n \times n}$ the identity matrix and $\partial A$ the boundary of a set $A \subseteq \mathbb{R}^n$. For a vector $x$ and a matrix $X$, we use $||x||$ and $||X||$ to denote the Euclidean norm and the spectral norm. We use $\text{vec}(X) \in \mathbb{R}^{n \times m}$ to denote the vectorization of $X \in \mathbb{R}^{n \times m}$, obtained by stacking its columns. We denote by $\nabla$ the gradient and $L_f V = \nabla V \cdot f$ the Lie derivative of a differentiable function $V$ along a vector field $f$. We use $\otimes$ to denote the Kronecker product and $\mathcal{G}(\mu(x), \mathcal{K}(x,x'))$ to denote a Gaussian Process distribution with mean function $\mu(x)$ and covariance function $K(x,x')$. A continuous function $\alpha : [0, \infty) \to [0, \infty]$ is of class $\mathcal{K}$ if it is strictly increasing and $\alpha(0) = 0$, and it is of class $\mathcal{K}_\infty$
by the scaling factor $\lambda > 0$. The QP formulation in (6) modifies the baseline controller $\bar{k}(x)$ 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 $F(x)$ and the barrier function $h(x)$ are unknown and need to be estimated from data. We present an approach for estimating the system dynamics and barrier functions from data in Sec.VI-A and Sec.VI-B, respectively. Our main goal is to develop techniques for safe and stable control synthesis with the estimated $F(x)$ and $h(x)$. 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). Given an estimated distribution on the unknown system dynamics $\text{vec}(F(x)) \sim \mathcal{GP}(\text{vec}(\hat{F}(x)), K_F(x, x'))$ and an estimated distribution on the barrier function $h(x) \sim \mathcal{GP}(\hat{h}(x), K_h(x, x'))$, design a feedback controller $\hat{k}$ such that, for each $x \in \mathcal{X}$:

$$P(\text{CLC}(x, \hat{k}(x)) \leq \delta) \geq p, \quad P(\text{CBC}(x, \hat{k}(x)) \geq 0) \geq p,$$

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 $\hat{F}(x)$ with known error bound $e_F(x)$,

$$\|F(x) - \hat{F}(x)\| \leq e_F(x), \quad \forall x \in \mathcal{X},$$

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

$$|h(x) - \hat{h}(x)| \leq e_h(x), \quad \|\nabla h(x) - \nabla \hat{h}(x)\| \leq e_{\nabla h}(x),$$

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

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

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:

$$\min_{u \in U} \|L(x)^{\top}(u - \hat{k}(x))\|^2 + \lambda \delta^2,$$

s.t. $P(\text{CLC}(x, u) \leq \delta) \geq p, \quad P(\text{CBC}(x, u) \geq 0) \geq p.$
where \( \gamma \geq 0 \).

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

\[
\mathbb{E}[p(x)] = [\nabla_x \hat{h}(x)]^T \hat{F}(x) + a[\nabla_x \hat{h}(x)] 0_m^T
\]

\[
\text{Var}[p(x)] = \hat{F}^T(x) \mathcal{H}_{x,x} K_h \hat{F}(x) + (I_{m+1} \otimes [\nabla_x \hat{h}(x)]^T) K_F (I_{m+1} \otimes [\nabla_x \hat{h}(x)]) + \left[ a^2 K_h + 2a[\nabla_x \hat{h}(x)]^T f(x) \right] a[\nabla_x K_h] g(x) 0_{m \times m}
\]

from which the statement follows.

Next, we describe the distribution of \( CLC(x, u) \).

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

\[
\mathbb{E}[CLC(x, u)] = E[q(x)]^T u \tag{17a}
\]

\[
\text{Var}[CLC(x, u)] = u^T \text{Var}[q(x)] u \tag{17b}
\]

where \( q(x) := \hat{F}^T(x) [\nabla_x V(x)] + [\alpha \nabla V(X)] 0_m^T \in \mathbb{R}^{m+1} \) and \( \text{Var}[q(x)] \) are computed in (18).

Proof. We can write the control Lyapunov condition as

\[
\mathbb{E}[\text{vec}(F(x)) + \text{vec}(V(X))] = \text{vec}(ABC) = (C^T \otimes A) \text{vec}(B)
\]

to rewrite first term in \( q(x) \) as:

\[
[\nabla_x V(x)]^T F(x) = (I_{m+1} \otimes [\nabla_x V(x)]^T) \text{vec}(F(x)).
\]

Since \( [\nabla_x V(x)]^T F(x) \) is known and deterministic and \( \text{vec}(F(x)) \sim \mathcal{GP}(\text{vec}(\hat{F}(x)), K_F(x,x')) \), we can express the distribution of \( q(x) \) as follows:

\[
\mathbb{E}[q(x)] = \hat{F}^T(x) [\nabla_x V(x)] + [\alpha V(X)] 0_m^T \tag{18}
\]

\[
\text{Var}[q(x)] = (I_{m+1} \otimes [\nabla_x V(x)]^T) K_F (I_{m+1} \otimes [\nabla_x V(x)])
\]

The result follows from plugging (18) into \( CLC(x, u) \). □

We use the mean and variance of \( CBC(x, u) \) and \( CLC(x, u) \) 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:

\[
\min_{u \in U, \delta \in \mathcal{R}, \lambda \in \mathcal{R}} l
\]

\[
\text{s.t. } \delta - \mathbb{E}[q(x)]^T u \geq c(p) \sqrt{u^T \text{Var}[q(x)] u},
\]

\[
\mathbb{E}[p(x)]^T u \geq c(p) \sqrt{u^T \text{Var}[p(x)] u},
\]

\[
l + 1 \geq \sqrt{(2l x)^T(u - \mathbf{k})(u - \mathbf{k})}^2 + (2\sqrt{\lambda} \delta)^2 + (l - 1)^2 \tag{19}
\]

where \( p, q \) 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 \( \gamma \geq 0 \),

\[
\mathbb{P}(CBC(x, u) \geq \mathbb{E}[CBC(x, u)]) - \gamma |x, u) \geq 1 - \frac{\text{Var}[CBC(x, u)]}{\text{Var}[CBC(x, u)] + \gamma^2}.
\]

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

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

which corresponds to the safety constraint in (19).

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

\[
\sqrt{\text{Var}[CBC(x, u)]} = \sqrt{u^T \text{Var}[p(x)] u} = ||D(x)|| \tag{20}
\]

where \( D(x)^T D(x) = ||p(x)|| \). According 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 \( u \) and the variance is quadratic in terms of \( u \), 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

\[
\min_{u \in U, \delta \in \mathcal{R}, \lambda \in \mathcal{R}} \sqrt{u^T \text{Var}[p(x)] u}
\]

\[
l \geq 2D(x)^T(u - \mathbf{k})^2 + \lambda \delta^2 \leq l.
\]

(21)

The last constraint in (21) corresponds to a rotated second-order cone, \( Q^{n_2}_\delta := \{ (x, y, z) \in \mathbb{R}^{n+2} | ||x||_r \leq \gamma r, z_r \geq 0, z_r \geq 0 \} \), which can be converted into a standard SOC constraint [30], \( \|2x^T + y - z\|_r^2 \leq y^2 + z_r \). Let \( y_r = l, z_r = 1 \) and consider the constraint \( \|L(x)^T(u - \mathbf{k})(u - \mathbf{k})\|^2 + \lambda \delta^2 \leq l \). Multiplying both sides by 4 and adding \( (l - 1)^2 \), makes the constraint equivalent to

\[
4\|L(x)^T \mathbf{k} - \mathbf{k}(u - \mathbf{k})\|^2 + 4\lambda \delta^2 + (l - 1)^2 \leq (l + 1)^2.
\]

Taking a square root on both sides, we end up with

\[
\sqrt{4\|L(x)^T(u - \mathbf{k})(u - \mathbf{k})\|^2 + (2\sqrt{\lambda} \delta)^2} \leq l + 1,
\]

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 \( \text{Var}[p(x)] \) increase, the feasible region of (19) gets smaller, and the optimal value worsens, cf. Fig. 1b for an illustration.

V. ROBUST SAFE CONTROL

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

(22)  

Using this notation, we can rewrite CBC(x, u) as

\[ CBC(x, u) = [\nabla h(x)]^T F(x) u + \alpha_h(h(x)) \]
\[ = [\nabla h(x)]^T \tilde{F}(x) u + d_{\nabla h}(x) F(x) u + \alpha_h(h(x) + d_h(x)). \]

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

\[ \min_{D_F, d_{\nabla h}, d_h} CBC(x, u) = \bar{p}(x)^T u + \min_{D_F, d_{\nabla h}, d_h} d_{\text{CBC}}(x, u) \geq 0. \]

Similarly, let \( \bar{q}(x) := F(x) \nabla V(x) + [\alpha_V(V(x))]^T \). Then, \( d_{\text{CBC}}(x, u) := [\nabla V(x)]^T D_F(x) u \), a robust version of the stability constraint \( CLC(x, u) \leq \delta \) can be written as:

\[ \max_{D_F} CLC(x, u) = \bar{q}(x)^T u + \max_{D_F} d_{\text{CLC}}(x, u) \leq \delta. \]  

(23)  

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

\[ \min_{u \in U, \delta \in U, l \in \mathbb{R}} l \]
\[ \text{s.t. } \bar{q}(x)^T u + \max_{D_F} d_{\text{CLC}}(x, u) \leq \delta \]
\[ \bar{p}(x)^T u + \min_{D_F, d_{\nabla h}, d_h} d_{\text{CBC}}(x, u) \geq 0 \]
\[ l + 1 \geq \sqrt{\|2L(x)^T (u - k(x))\|^2 + (2\sqrt{\lambda} \delta)^2 + (l - 1)^2}. \]  

(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 \( \min_{D_F, d_h, d_{\nabla h}} d_{\text{CBC}}(x, u) \) subject to (22). In general, this is not a convex constrained quadratic program which does not have a closed-form expression of the minimizer as a function of \( u \). Instead, we make the second constraint in (24) more conservative using the Cauchy-Schwarz inequality, which leads to a convex SOC, whose optimal solution is guaranteed to be feasible for (24).

**Proposition V.1 (Robust CLF-CBF SOC).** 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 SOC is included in the feasible set of (24):

\[ \min_{u \in U, \delta \in U, l \in \mathbb{R}} l \]
\[ \text{s.t. } \delta - \bar{q}(x)^T u \geq e_F(x) \|\nabla V(x)\| \|u\|, \]
\[ p \geq e_{\nabla h}(x) \|\tilde{F}(x) u\|, \]
\[ q \geq \left( e_F(x) \|\nabla \tilde{h}(x)\| + e_{\nabla h}(x) e_F(x) \right) \|u\|, \]
\[ (\nabla \tilde{h}(x))^T \tilde{F}(x) u + \alpha_h(\tilde{h}(x) - e_h(x)) \geq p + q, \]
\[ l + 1 \geq \sqrt{\|2L(x)^T (u - k(x))\|^2 + (2\sqrt{\lambda} \delta)^2 + (l - 1)^2}. \]  

(25)

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

\[ \max_{D_F, d_h} d_{\text{CLC}}(x, u) = e_F(x) \|\nabla V(x)\| \|u\|. \]

For the safety constraint in (24), note that

\[ \min_{D_F, d_h} d_{\text{CBC}}(x, u) = \min_{D_F, d_h} \left( \|d_{\nabla h}(x) D_F(x) u\| + \|d_{\nabla h}(x) D_F(x) u\| + \min_{d_{\nabla h}} \alpha_h(\tilde{h}(x) + d_h(x)). \right. \]  

(26)

Since \( e_h(x) \geq 0 \) and \( \alpha_h \) is an extended class \( \mathcal{K}\alpha \) function,

\[ \min_{d_h} \alpha_h(\tilde{h}(x) + d_h(x)) = \alpha_h(\tilde{h}(x) - e_h(x)). \]  

(27)

Applying the Cauchy-Schwarz inequality on each term,

\[ \min_{D_F, d_h, d_{\nabla h}} d_{\text{CBC}}(x, u) \geq -\|d_{\nabla h}\| \|\tilde{F}(x) u\| - \|\nabla \tilde{h}(x)\| \|D_F(x) u\| - \|d_{\nabla h}\| \|D_F(x) u\| + \alpha_h(\tilde{h}(x) - e_h(x)) \geq -e_{\nabla h}(x) \|\tilde{F}(x) u\| - e_F(x) \|\nabla \tilde{h}(x)\| \|u\| + \alpha_h(\tilde{h}(x) - e_h(x)). \]

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

\[ e_{\nabla h}(x) \|\tilde{F}(x) u\| + (e_F(x) \|\nabla \tilde{h}(x)\| + e_{\nabla h}(x) e_F(x)) \|u\| \leq \|\nabla \tilde{h}(x)\| \tilde{F}(x) u + \alpha_h(\tilde{h}(x) - e_h(x)). \]

(28)

Constraints of the form \( \|Ax - a\| + \|Bz - b\| \leq c^T z \) can be replaced by the set of constraints \( \|Ax - a\| \leq p, \|Bz - b\| \leq q, p + q \leq c^T z \) combined. Thus, (28) is equivalent to the second, third, and fourth constraints in (25) together.

\[ \square \]

**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 SOC (25) reduces to a CLF-CBF QP (6). If \( e_F \equiv 0 \) while \( e_h(x), e_{\nabla h}(x) > 0 \), 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, \( K_F(x, x) = \frac{e_f(x)}{e_{\nabla h}(x)} I_{(m+1)n} \), 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. 1a). 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 \( x := [x, y, \mu]^T \in \mathbb{R}^2 \times [-\pi, \pi] \).
and \( \mathbf{u} := [1, v, \omega]^\top \in \{1\} \times \mathbb{R}^2 \), respectively. We collect a dataset \( \mathcal{D} = \{t_{0:N}, x_{0:N}, \mathbf{u}_{0:N}\}_{i=1}^D \) of \( D = 40000 \) state sequences \( x_{0:N}^{(i)} \) obtained by applying random control inputs \( \mathbf{u}_{0:N}^{(i)} \) to the robot with initial condition \( x_0^{(i)} \) 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 \cite{abrusi2022} to approximate the unknown robot dynamics \( F \) with a neural network \( F_{\theta} \) based on the dataset \( \mathcal{D} \). A forward pass through the ODE network is obtained using an ODE solver:

\[
\{\dot{x}_1^{(i)}, \dot{x}_2^{(i)}, \ldots, \dot{x}_N^{(i)}\} = \text{ODESolve}(x_0^{(i)}, F_{\theta}(\cdot)\mathbf{u}^{(i)}, t_1^{(i)}, \ldots, t_N^{(i)}).
\]

We use a loss function,

\[
\begin{align*}
\min_{\theta} & \sum_{i=1}^D \sum_{j=1}^N \ell(x_j^{(i)}, \dot{x}_j^{(i)}), \\
\text{s.t.} & \quad \dot{x}_j^{(i)}(t) = F_{\theta}(x_j^{(i)}(t))\mathbf{u}_j^{(i)}(t), \quad \dot{x}_j^{(i)}(j\tau) = x_j^{(i)}, \\
& \quad \mathbf{u}_j^{(i)}(t) \equiv \mathbf{u}_j^{(i)} \quad \text{for } t \in \{j\tau, (j+1)\tau\},
\end{align*}
\]

where \( \ell(x, \dot{x}) = \|x, y, \cos \mu, \sin \mu, \dot{x}, \dot{y}, \cos \mu, \sin \mu\|^2 \). To update the weights \( \theta \), the gradient of the loss function is back-propagated by solving another ODE with different states backwards in time. Please refer to \cite{abrusi2022} for details.

Gal and Ghahramani \cite{gal2016} 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 \( F_{\theta} \), and apply dropout to each hidden layer with rate 0.05. Given a query state \( x \in \mathcal{X} \), Monte-Carlo estimates of the predictive mean \( F_{\theta}(x) \) and element-wise standard deviation \( \bar{\Sigma}(x) \) of the dynamics are obtained with \( T = 100 \) stochastic forward passes through the dropout neural network model. We use \( F_{\theta}(x) \) for the mean of system dynamics and \( K_{F}(x, x) = \text{diag}(\text{vec}(\Sigma(x)))^2 \) for the variance of the dynamics. To obtain worst-case error bounds \( e_F(x) \), we set \( e_F(x) = \|3.89\Sigma(x)\| \) (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 \( \{x_i, \mathbf{u}_j\}_{j=1}^M \) as test data, we consider the following test-time loss function,

\[
L = \frac{1}{M} \sum_{i=1}^M \ell(F(x), \mathbf{u}, \hat{F}(x, \mathbf{u})).
\]

Our learned dynamics model is quite accurate, and the average test loss is \( L = 0.0037 \).

\begin{table}[h]
\centering
\caption{Empirical SDF estimation error \( \mathcal{E} \) and dropout-network SDF estimation error averaged across 8 object instances under different LiDAR measurement noise standard deviation \( \sigma \).}
\begin{tabular}{|c|c|c|}
\hline
LiDAR Noise \( \sigma \) & SDF Empirical Error \( \mathcal{E} \) & SDF Dropout Error \( \mathcal{E} \) \\
\hline
0.01 & 0.0173 & 0.0132 \\
0.02 & 0.0288 & 0.0184 \\
0.05 & 0.0463 & 0.0242 \\
\hline
\end{tabular}
\end{table}

\section{B. Online CBF Estimation}

The robot is equipped with a LiDAR scanner with a 270° field of view, 200 rays per scan, 3 meter range, and zero-mean Gaussian measurement noise with standard deviation \( \sigma \in \{0.01, 0.02, 0.03\} \). 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. 2b) to describe each \( \mathcal{O}_i \). The SDF function \( \varphi_i : \mathbb{R}^2 \to \mathbb{R} \) is

\[
\varphi_i(y) = \begin{cases} 
-d(y, \partial \mathcal{O}_i), & y \in \mathcal{O}_i, \\
d(y, \partial \mathcal{O}_i), & y \notin \mathcal{O}_i,
\end{cases}
\]

where \( d \) denotes the Euclidean distance from a point \( y \in \mathbb{R}^2 \) and the set boundary \( \partial \mathcal{O}_i \). We employ incremental training with replay memory (ITRM) \cite[Sec. IV]{gupta2020} to estimate an SDF \( \varphi_i \) for each \( \mathcal{O}_i \) from the LiDAR measurements. We use a 4-layer fully-connected neural network with parameters \( \theta \) and dropout layers to yield \( \hat{\varphi}_i(y, \theta) \) with dropout rate 0.05 applied to each 512-neuron hidden layer. Given \( y \in \mathbb{R}^2 \), we obtain the predictive SDF mean \( \hat{\varphi}_i(y) \) and standard deviation \( \hat{\sigma}_i(y) \) 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,

\[
\mathcal{E}_i = \frac{1}{m} \sum_{j=1}^{m} \left| \varphi_i(y_j) \right|, \quad \{y_j\}_{j=1}^{m} = m = 500 \text{ uniformly sampled on the surface of the object}.
\]

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 \cite{amato2020} that the SDF is a valid CBF. Let \( z = [x, y] \in \mathcal{Z} \subset \mathbb{R}^2 \) be the position part of \( x \). 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:

\[
\mathcal{h}_i(x) = \varphi_i(z, \theta) - \rho.
\]

For variance \( \mathcal{K}_i(x, x) \) in Sec. IV, we set \( \mathcal{K}_i(x, x) = \hat{\sigma}_i^2(x) \). We also take \( \nabla \mathcal{h}_i(x) = \nabla \varphi_i(z, \theta) \) and compute \( \mathcal{H}_{x,x} \mathcal{K}_i(x, x') \) by Monte-Carlo estimation using double back-propagation. We
set the worst case error bounds $e_h(x)$, $e_{\nabla h}(x)$ in Sec. V as the 99.99% confidence bounds of a Gaussian random variable with standard deviation $\tilde{\sigma}_i(z)$. If the robot observes multiple obstacles in the environment, we compute multiple CBFs $\tilde{h}_i(x)$ 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 Sec. VI.A and the online CBF estimates from Sec. 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 $L(x) = \text{diag}([0, 10, 3])$ and $\tilde{k}(x) = [1, v_{\text{max}}, 0]^\top$ where $v_{\text{max}} = 0.65$ is the maximum linear velocity for the TurtleBot. The remaining parameters were $\lambda = 1000$, $\alpha_V(V(x)) = 2V(x)$, and $\alpha_h(h_i(x)) = h_i(x)$.

In the first set of experiments (Fig. 3a and Fig. 3b), we demonstrate safe navigation to a goal point with a CLF candidate $V(x) = (x - 2)^2 + (y - 3)^2$. In Fig. 3a, 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. 3b, 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, Sec. VI] to construct a valid CLF $V(x)$ 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. 3c, 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 $CBC(x, u)$, 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 farther 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, Sec. VI] as the metric to measure trajectory similarities. The distance value is computed by averaging the realizations ($\sigma_1, \sigma_2, \sigma_3$) = $(0, 0, 0)$, where the CLF-CBF-QP controller fails to construct a valid CLF $V(x)$ of safe trajectory tracking using the approach in [29, Sec. VI] as the metric to measure trajectory similarities.
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. 2b, 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 9700K CPU. The time needed for solving one QP instance is 0.00863s while the times needed for solving the proposed probabilistic and robust SOCPs are 0.0109s and 0.0122s. 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 pretraining of the SDF neural network, and explore adaptive techniques for safe control synthesis given varying uncertainty levels and robot objectives.

REFERENCES


